CipherLab

User Guide

Android Programming

for RS30 Mobile Computer

Version 1.07

CIPHERLAB

Copyright © 2015 ~ 2017 CIPHERLAB CO., LTD.
All rights reserved

The software contains proprietary information of CIPHERLAB CO., LTD.; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

Due to continued product development this information may change without notice. The information and intellectual property contained herein is confidential between CIPHERLAB and the client and remains the exclusive property of CIPHERLAB CO., LTD. If you find any problems in the documentation, please report them to us in writing. CIPHERLAB does not warrant that this document is error-free.

No part of this publication 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 CIPHERLAB CO., LTD.

For product consultancy and technical support, please contact your local sales representative. Also, you may visit our web site for more information.

The CipherLab logo is a registered trademark of CIPHERLAB CO., LTD.

Other product name mentioned in this manual may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

The editorial use of these names is for identification as well as to the benefit of the owners, with no intention of infringement.

CIPHERLAB CO., LTD.
Website: http://www.cipherlab.com

RELEASE NOTES

Version	Date	Notes
1.07	Apr. 12, 2017	▸ Modified: **1.5.1** – change “InverseType.Autodetect” to “InverseType.AutoDetect” (Get_UserPreferences) ▸ Modified: **1.5.1** – Continuous and Presentation modes for timeoutBetweenSameSymbology ▸ Modified: **1.5.1** – Replace “0x06” with “InterCharacterGapSize.Normal” of Get_UserPreferences ▸ Modified: **1.5.1** – Replace “0x0A” with “InterCharacterGapSize.Large” of Get_UserPreferences ▸ Modified: **1.5.1** – IlluminationPowerLevel added to Get_UserPreferences & Set_UserPreferences ▸ Modified: **1.5.3** – parameter names of notisEditingType changed ▸ Modified: **1.5.14** – description for transmitCheckDigit ▸ Modified: **1.5.36** – descriptions for MatrixMirrorImage ▸ New: **1.8 Callback**
1.06	Nov. 04, 2016	▸ New: **1.7 Intent** ▸ New: **Appendix V Open Source License**
1.05	Oct. 14, 2016	▸ Modified: **Development Tool** – Xamarin appended ▸ Modified: **Chapter 1** – Barcodeapi_vx_x_xx.dll added ▸ New: **1.1.3 Xamarin for Visual Studio 2015** ▸ New: **1.3.3 Reader API Version** ▸ Modified: **1.5.1** – TriggerType.PresentationMode added to GetUserPreferences ▸ Modified: **1.5.1** – “public int timeoutPresentationMode” added to GetUserPreferences and SetUserPreferences ▸ Modified: **1.5.10 GS1128 Class** – application ID added ▸ Modified: **1.5.17 GS1DataBar14 Class** – “public int securityLevel” added ▸ Modified: **1.5.19 GS1DataBarExpanded Class** – “public int securityLevel” added
1.04	May 24, 2016	▸ Modified: **1.5.5 Code39 Class** – convertToCode32 changed to “False” by default ▸ Modified: **1.5.6 TriopticCode39 Class** – changed to “False” by default ▸ Modified: **1.5.9 GS1DataBarExpanded Class** – “fieldSeparator” corrected ▸ Modified: Data type of all “fieldSeparator” class members changed from ‘char’ to ‘String’ ▸ Modified: **Appendix II**: remove “MacroPDF” from Supported Symbologies

1.03	Feb. 02, 2016	• Modified: **1.3.1 Get_ReaderOutputConfiguration** – add szCharsetName and clearPreviousData parameters • Modified: **1.3.1 Set_ReaderOutputConfiguration** - add szCharsetName and clearPreviousData parameters • Modified: **1.5.3 Codabar Class** – "Start/Stop characters" description for notisEditingType added • Modified: **1.5.3 Codabar Class** – notisEditing marked as "Reserved" • New: **Appendix III** – Code Type & Symbology
1.02	Oct. 26, 2015	• **1.5.3 Codabar Class** – notes for Modulo_7DR of verifyCheckDigit added
1.01	Oct. 12, 2015	• **1.3.1 Get_ReaderOutputConfiguration** - replace Enable_State with KeyboardEmulationType • **1.3.1 Set_ReaderOutputConfiguration** - replace Enable_State with KeyboardEmulationType • **1.5.1 Get_UserPreferences** - TriggerType.PulseMode, BlinkingMode, HostMode, PresentationMode removed • **1.5.2 Get_Symbology** - Codabar, GS1DataBarLimited updated • **1.5.3** - Codabar Class updated • **1.5.18** - GS1DatabarLimited Class updated • **Appendix III** - Sample Code updated
1.00	Jul. 14, 2015	First Release

CONTENTS

RELEASE NOTES ..... - 3 -

INTRODUCTION 1

Development Tool 2

BARCODE READER API 3

1.1. Import Library 4

1.1.1 Android Studio 4

1.1.2 Eclipse 7

1.1.3 Xamarin for Visual Studio 2015 11

1.2. Initialize/Identify Reader 14

1.2.1. Initialization 14

1.2.2. Active Device 15

1.2.3. Reader Type 16

1.3. Obtain Data 17

1.3.1. Data Output Settings 17

1.3.2. Reader Service Version 23

1.3.3. Reader API Version 23

1.4. Manipulate Status Indication 24

1.4.1. Notification Settings 24

1.5. Configure Scan Engine 26

1.5.1. Preferences 26

1.5.2. Symbology Settings 34

1.5.3. Codabar Class 41

1.5.4. Code11 Class 43

1.5.5. Code39 Class 44

1.5.6. TriopticCode39 Class 45

1.5.7. Korean3Of5 Class 45

1.5.8. Code93 Class 45

1.5.9. Code128 Class 46

1.5.10. GS1128 Class 46

1.5.11. ISBT128 Class 47

1.5.12. Chinese2Of5 Class 47

1.5.13. Industrial2Of5 Class 48

1.5.14. Interleaved2Of5 Class 48

1.5.15. Matrix2Of5 Class 49

1.5.16. UccCoupon Class 50

1.5.17. GS1DataBar14 Class 50

1.5.18. GS1DataBarLimited Class 51

1.5.19. GS1DataBarExpanded Class 51

1.5.20. Msi Class 52

1.5.21. Ean8 Class 53

1.5.22. Ean13 Class 54

1.5.23. UpcA Class 55

1.5.24. UpcE Class 56
1.5.25. UpcE1 Class 57
1.5.26. Composite Class 58
1.5.27. USPostal Class 59
1.5.28. UKPostal Class 59
1.5.29. JapanPostal Class 60
1.5.30. AustralianPostal Class 60
1.5.31. DutchPostal Class 60
1.5.32. USPSPostal Class 61
1.5.33. UPUFICSPostal Class 61
1.5.34. PDF417 Class 61
1.5.35. MicroPDF417 Class 62
1.5.36. DataMatrix Class 62
1.5.37. MaxiCode Class 63
1.5.38. QRCode Class 63
1.5.39. MicroQR Class 63
1.5.40. Aztec Class Class 64
1.6. Reset Reader 64
1.7. Intent 65
1.7.1. Reader Service Connection 65
1.7.2. Software Trigger 65
1.7.3. Hardware Scan Key 65
1.7.4. Decoding Error 66
1.7.5. Data Sending 66
1.8. Callback 67
1.8.1. SetReaderCallback 67
1.8.2. GetReaderCallback 67
1.8.3. Sample Code 67
SAM API 69
2.1 Bind SAM Service 70
2.2 Service Information 71
RESPONSE CODE INSTRUCTIONS 73
SCAN ENGINE SETTINGS 75
Symbologies Supported 76
CODE TYPE & SYMBOLOGY 79
SAMPLE CODE 83
OPEN SOURCE LICENSE 85

INTRODUCTION

This Programming Guide contains necessary information for building Android applications that can tune reader module(s), capture data, or control built-in hardware on RS30 Series Mobile Computers, which are powered by Android.

Android Framework makes it easy to create such applications. Simply import the prospective Android component (Android Class Library) to your Android application to make your way to build it.

We recommend that you read the documents thoroughly before use and keep them at hand for quick reference.

Thank you for choosing CipherLab products!

DEVELOPMENT TOOL

Before developing Android applications, programmers are supposed to make their machine ready with the requirements as follows:

- Java SE Development Kit (JDK, Java SE 7 or greater is recommended)
- Android SDK
- Android Studio, Eclipse IDE, or Xamarin for Visual Studio
- Visual Studio 2015 (a must while using Xamarin)

The software tools listed above are free and can be downloaded from their official websites respectively. Programmers are assumed to possess Android programming knowledge.

Chapter 1

BARCODE READER API

Before developing your self-made application, the offered "***barcodeapi_vx_x_xx.jar***" or "***barcodeapi_vx_x_xx.dll***" library file has to be imported into your project.

Library required:
Barcodeapi_vx_x_xx.jar (for Android Stusio or Eclipse)
Barcodeapi_vx_x_xx.dll (for Xamarin)

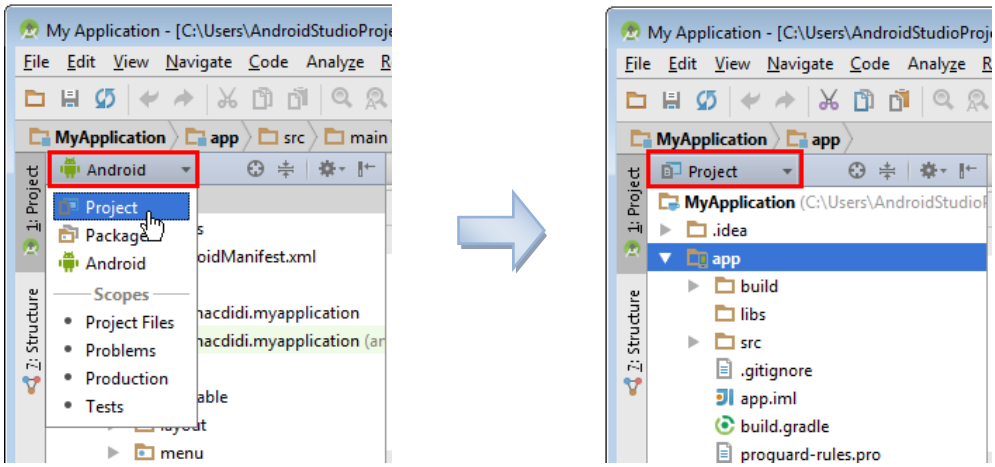
IN THIS CHAPTER

1.1 Import Library 4
1.2 Initialize/Identify Reader 14
1.3 Obtain Data 17
1.4 Manipulate Status Indication 24
1.5 Configure Scan Engine 26
1.6 Reset Reader 64
1.7 Intent 65
1.8 Callback 67

1.1. IMPORT LIBRARY

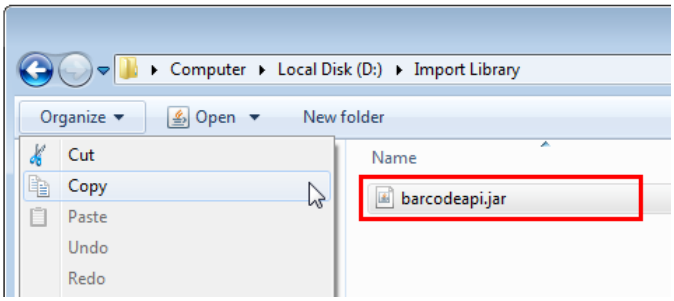
1.1.1 ANDROID STUDIO

1) After creating an Android Studio project, click the ***Android*** project view icon to switch to the Traditional project view.

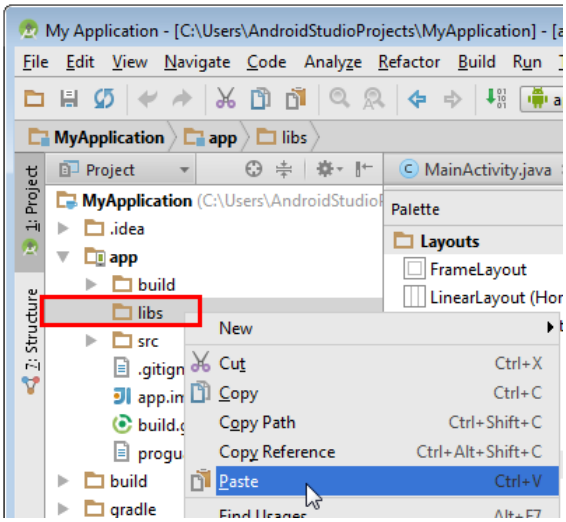


Android project view *Traditional project view*

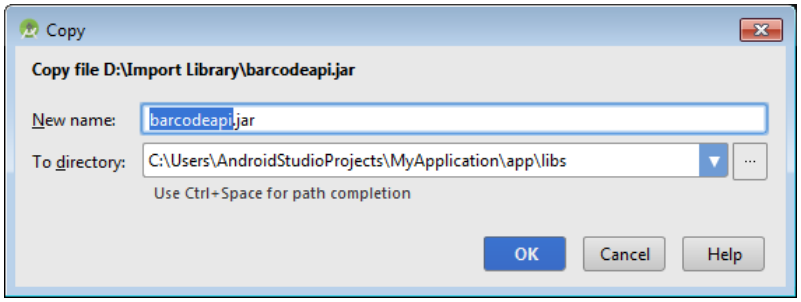
2) Locate the offered "***barcodeapi.jar***" library file in your file system and copy it.



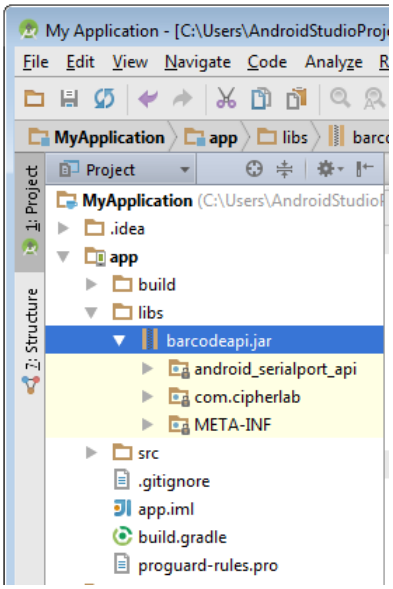
3) Right-click on the ***libs*** folder in the project view, and then select **Paste**.



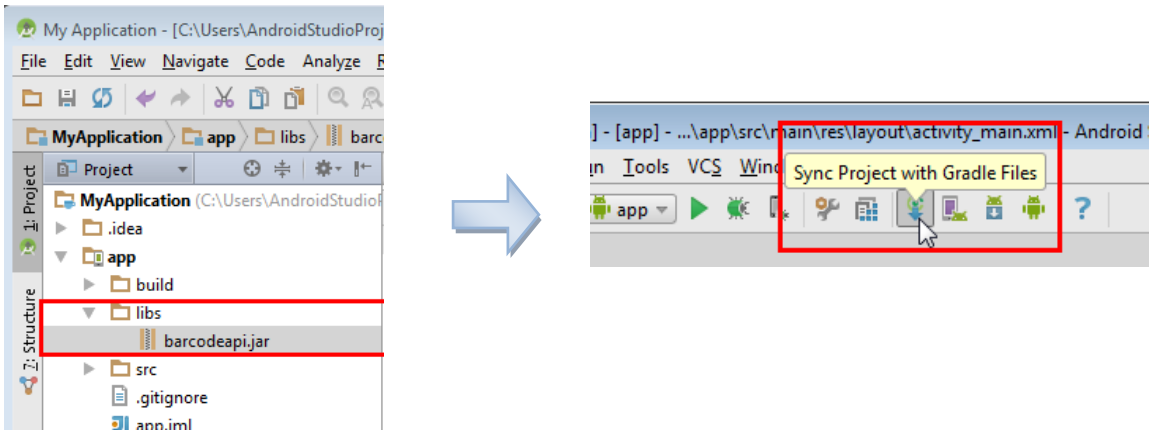
4) A dialog shows up indicating the file name and the destination directory to be copied. Click the **OK** button to start importing the library file.



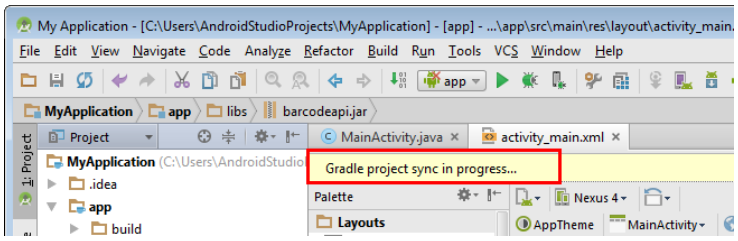
5) In the project view, you can see the library is imported.



If you don't see any files listed under the *barcodeapi.jar* item, please click the **Sync Project with Gradle Files** button from the toolbar.



After the project sync completes, the files relating to the library will be displayed.



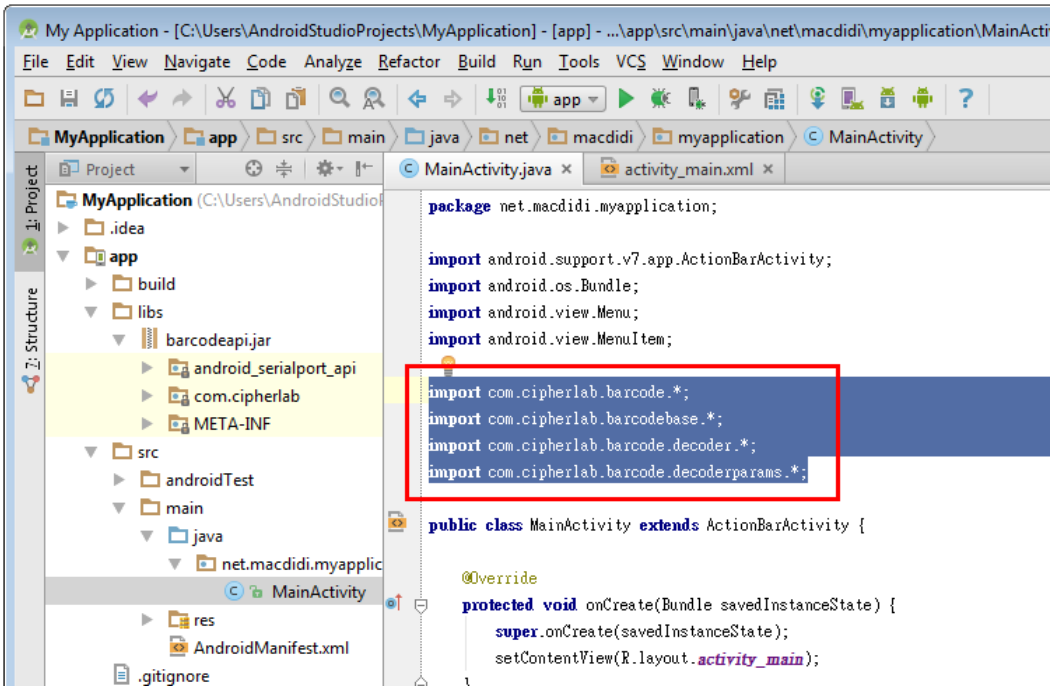
6) Finally, import the packages by manually typing statements as follows to finish the library import process:

import com.cipherlab.barcode.;*

import com.cipherlab.barcodebase.;*

import com.cipherlab.barcode.decoder.;*

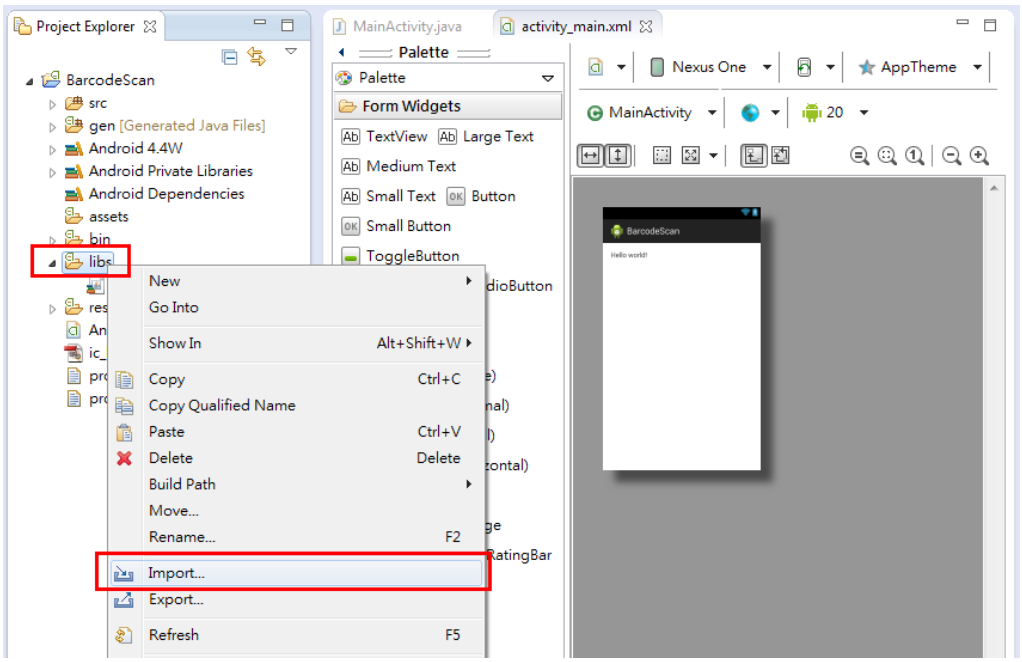
import com.cipherlab.barcode.decoderparams.;*



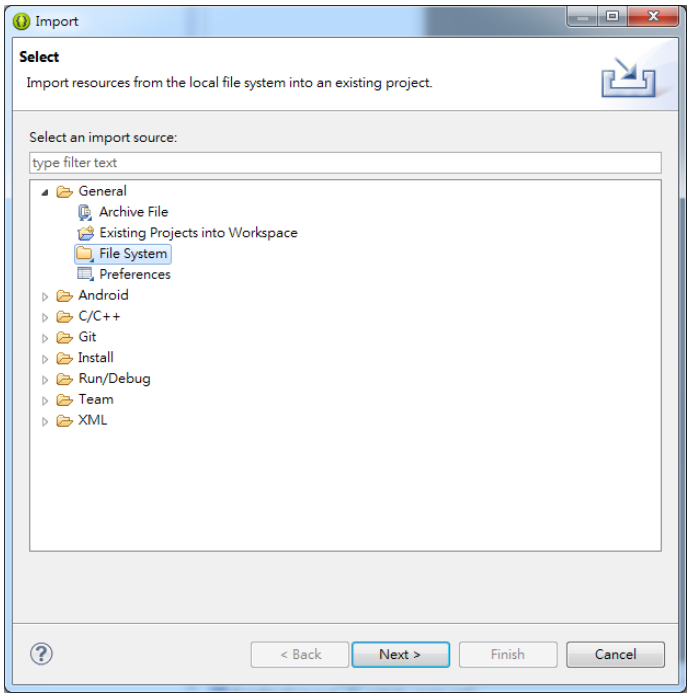
1.1.2 ECLIPSE

Have the library file (barcodeapi.jar) be ready on the file system. And then follow the instructions below:

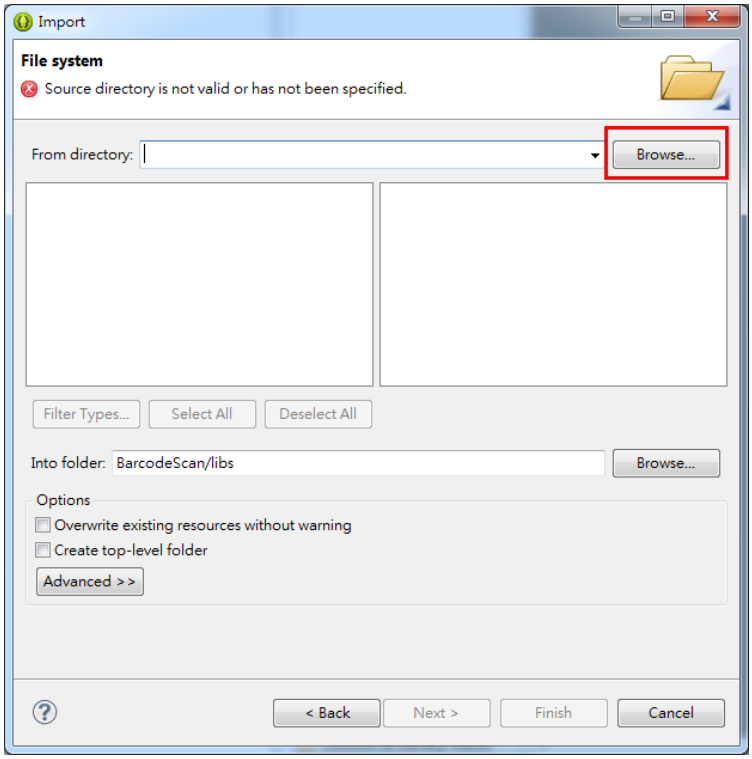
1) In the **Project Explorer** view, right-click the ***libs*** folder in your Android project and then select **Import**.



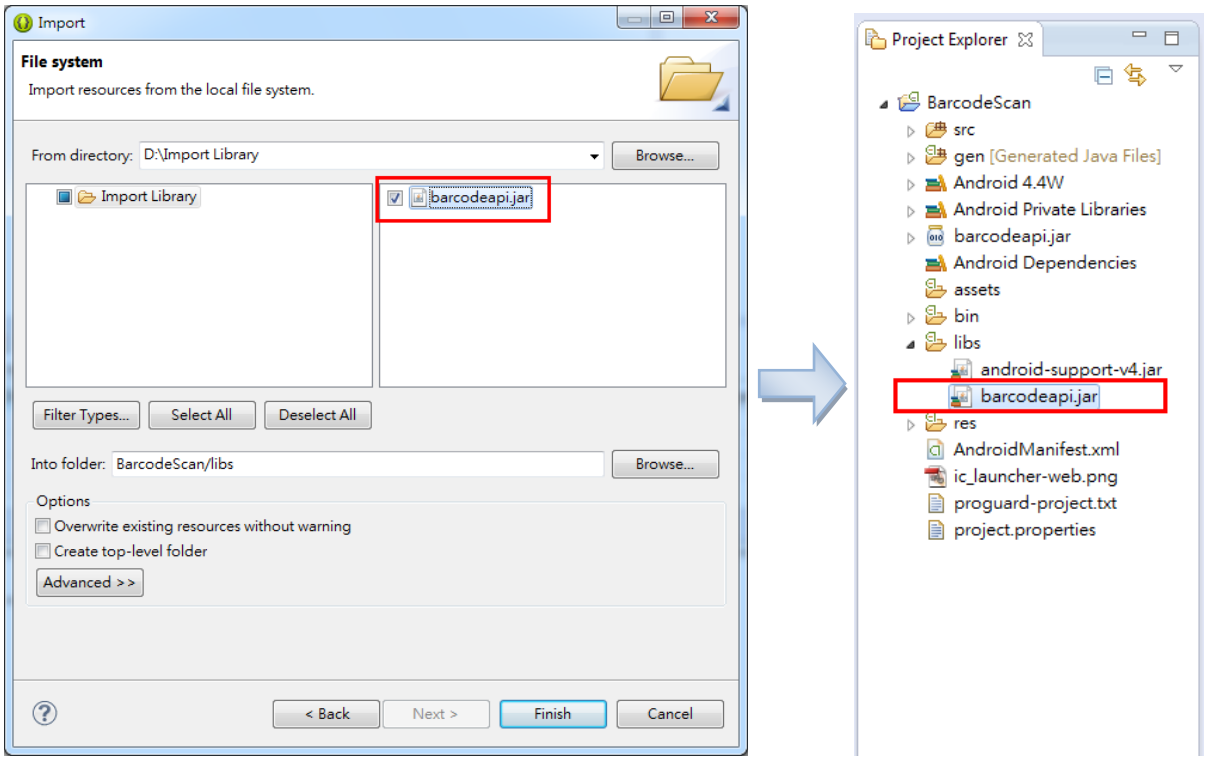
2) With the **Import** dialog showing up, select **General -> File System** and click the **Next** button.



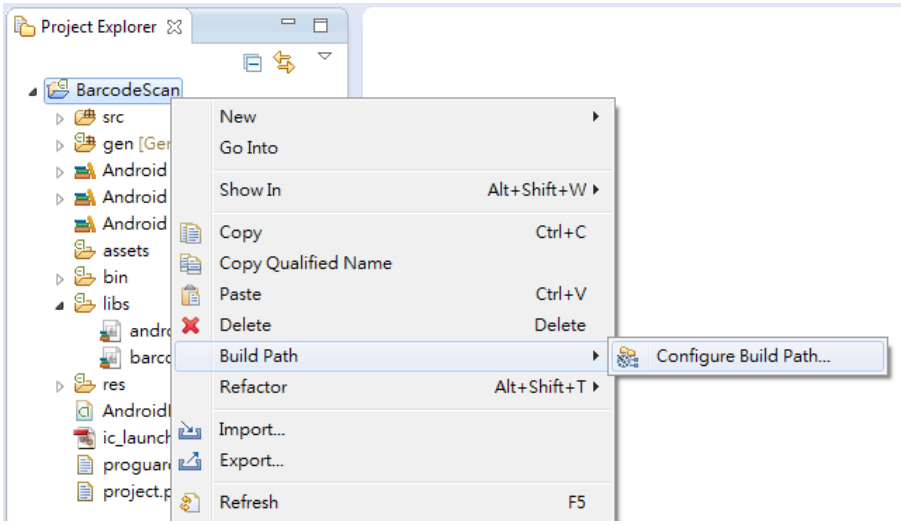
3) Click the **Browse** button to locate the library file.



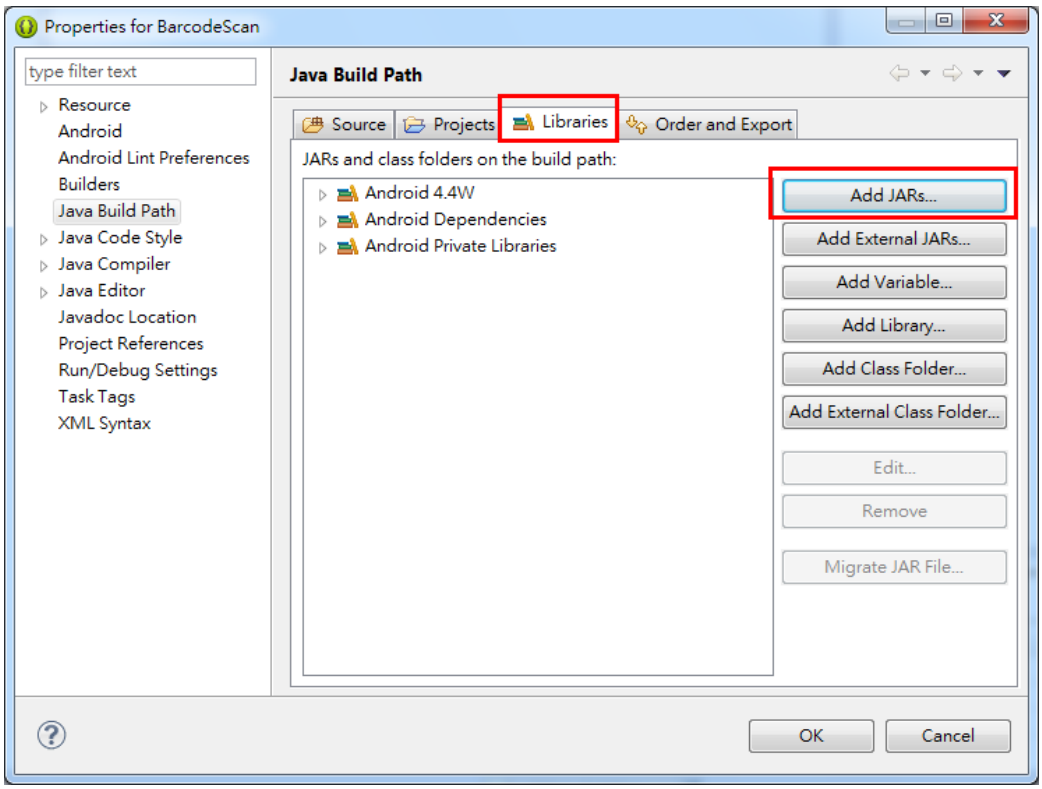
4) For example, "D:\Import Library" is the destination directory where the library file is located. Check the .JAR file in the right pane and click the **Finish** button. You will see the library has been imported into your project.



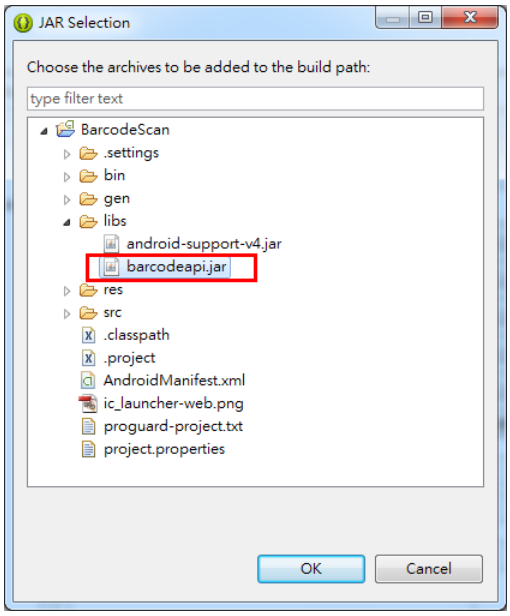
5) If the library is not on your build path, please right-click on the project name and then select **Build Path -> Configure Build Path** on the pop-up menu.



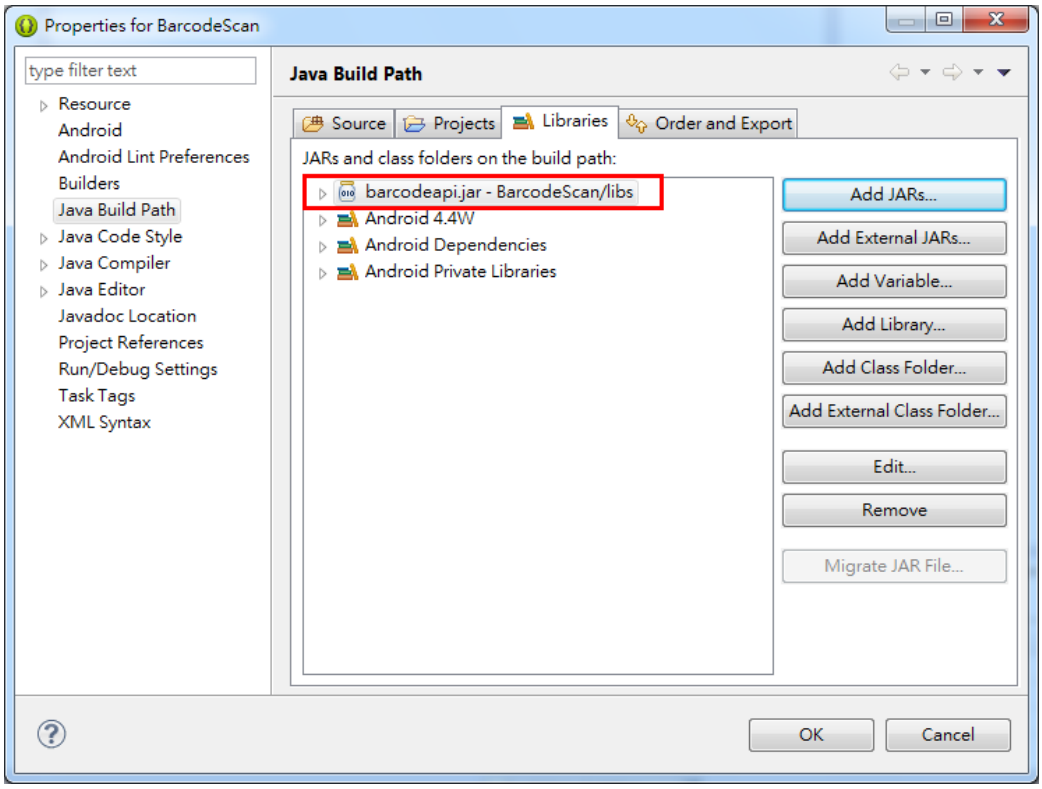
6) After the project properties window shows up, click the Libraries tab and then click the **Add JARs** button.



7) Select the JAR file you've just imported. Click the **OK** button.

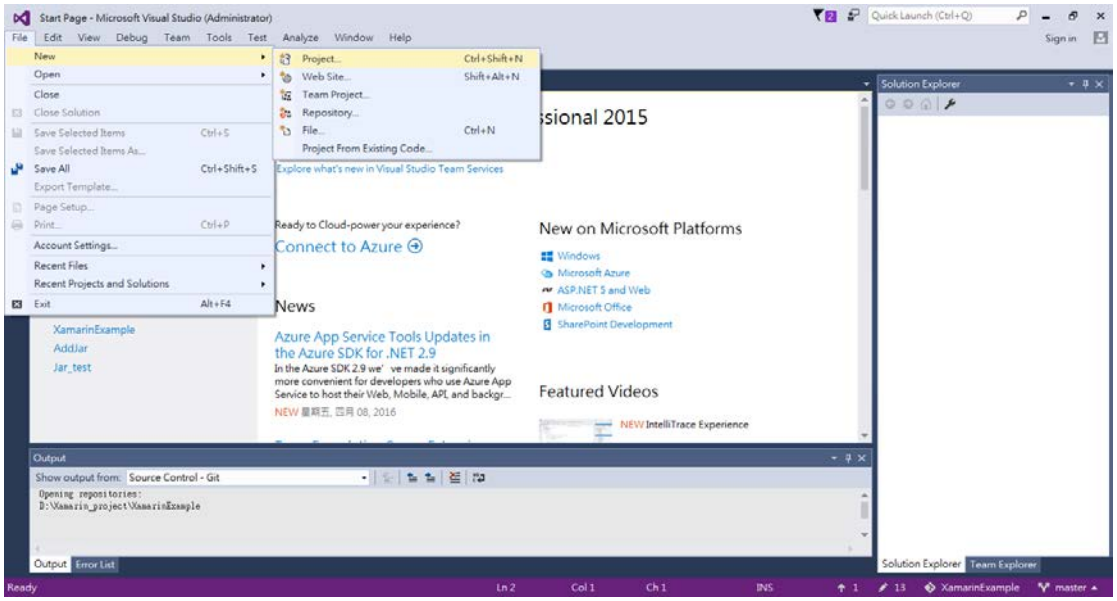


8) Now the library is on the build path.

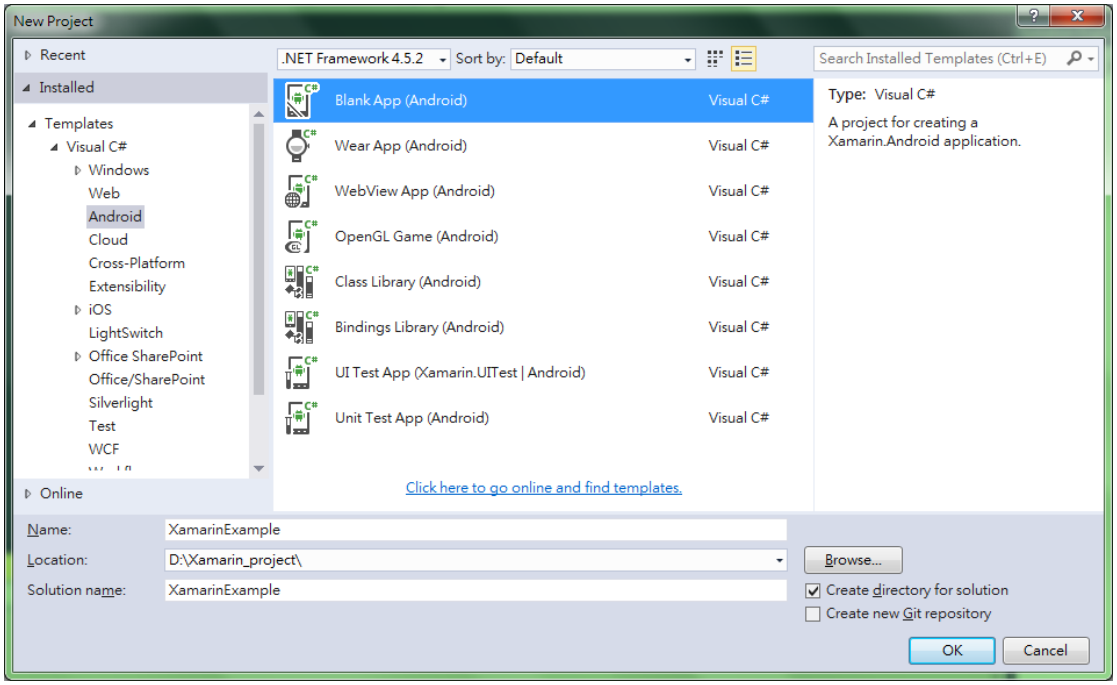


1.1.3 XAMARIN FOR VISUAL STUDIO 2015

1) Download Xamarin from http://store.xamarin.com and install it.
2) Open Visual Studio on your PC to create a new project.

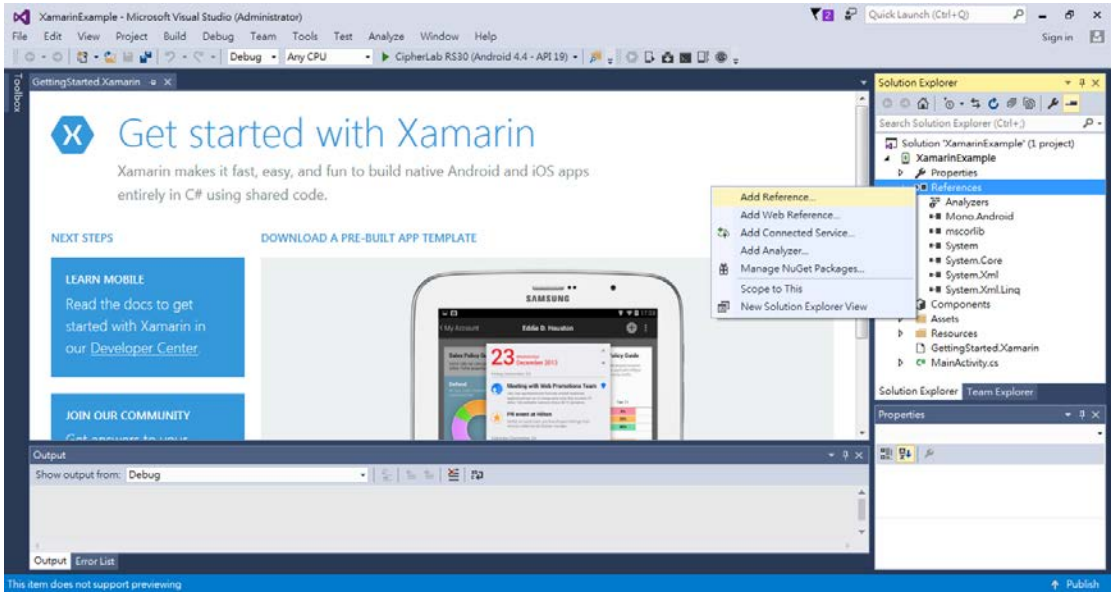


3) Select **File → New → Project** and then choose the **C# → Windows → Android → Blank APP (Android)** project template.

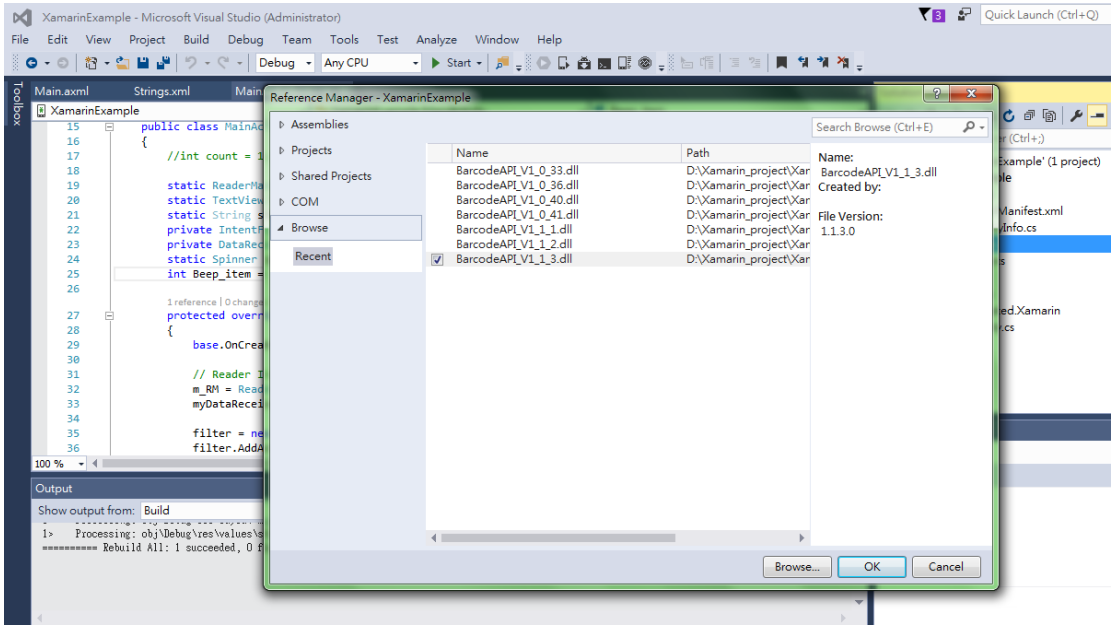


4) Copy the offered "***barcodeapi_vx_x_xx.dll***" library file to the new project folder.

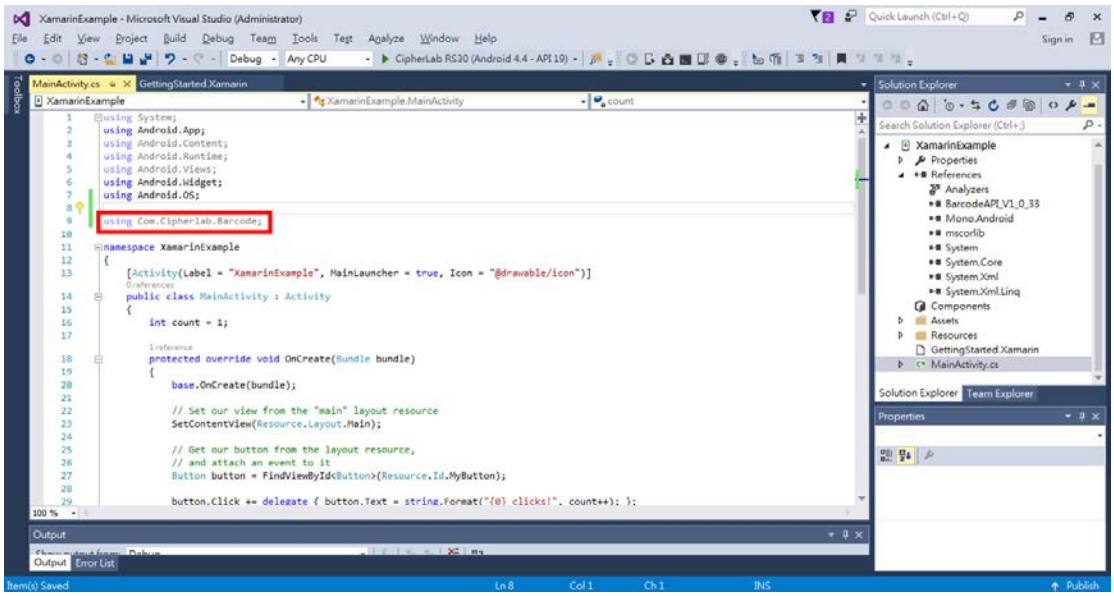
5) In the right pane of Visual Studio, right-click on **References** and then select **Add Reference** under the name of the new project.



6) In the **Reference Manager** dialog, click **Browse** in the left pane and then click the **Browse** button to locate the library file. When the "***barcodeapi_vx_x_xx.dll***" file is selected in the file browser dialog, click the **Add** button. Now check the added file and click the **OK** button.



7) Include the “using Com.Cipherlab.Barcode;” statement in MainActivity.cs.



1.2. INITIALIZE/IDENTIFY READER

1.2.1. INITIALIZATION

InitInstance	
Purpose	Creates a ReaderManager instance before employing any APIs.
Syntax	**ReaderManager InitInstance (Context context);**
Example	`private ReaderManager mReaderManager;` `mReaderManager = ReaderManager.InitInstance(this);`
Return Value	Gets a ReaderManager instance if successful, else null.
Remarks	As this is a function that gets reader module(s) ready, it must be called before any other functions.
See Also	GetActive, SetActive, GetReaderType, ResetReaderToDefault

Release	
Purpose	Releases resources.
Syntax	**void Release ();**
Example	`mReaderManager.Release();`
See Also	InitInstance, ResetReaderToDefault

1.2.2. ACTIVE DEVICE

GetActive

Purpose: Gets the reader active state.

Syntax: **boolean GetActive ();**

Example:
```
boolean bRet = mReaderManager.GetActive();
```

Return Value: If successful, it returns the reader active state accordingly:

false	Disable
true	Enable

See Also: InitInstance, SetActive, GetReaderType

SetActive

Purpose: Sets the reader module to use.

Syntax: **ClResult SetActive (boolean** *bActive***);**

Parameters: *bActive*
[in] A value that specifies the active device.

false	Disable
true	Enable

Example:
```
boolean bRet = mReaderManager.GetActive();
if (bRet==false){
      ClResult clRet = mReaderManager.SetActive(true);
}
```

Return Value: If successful, it returns ClResult.S_OK.
Otherwise, it returns ClResult.S_ERR.

Remarks: Trigger key's triggering behavior applies only to the active device. No event occurs with the inactive device.
There is no error signal if the reader to set is not found.

See Also: GetActive, InitInstance, GetReaderType

1.2.3. READER TYPE

GetReaderType	
Purpose	Gets the available reader type(s)
Syntax	**BcReaderType GetReaderType ();**
Example	`BcReaderType myReaderType = mReaderManager.GetReaderType();`
Return Value	If successful, it returns the reader type(s) accordingly:

Moto_1D_SE955
Moto_2D_4500
CL_1D_SM1
Moto_1D_SE965

See Also	InitInstance, GetActive, SetActive

1.3. OBTAIN DATA

1.3.1. DATA OUTPUT SETTINGS

Processed Data

Set_ReaderOutputConfiguration() sets which info to attach to a decoded barcode data:

Info	Description
Code Type	Barcode Type. See **Set_ReaderOutputConfiguration()** paramenter *showCodeType*.
Prefix Code	No prefix code if value is 0. See **Set_ ReaderOutputConfiguration()** parameter sz*PrefixCode*.
Decode Data	Decoded barcode data
Suffix Code	No suffix code if value is 0. See **Set_ ReaderOutputConfiguration()** parameter sz*SuffixCode*.
Code Length	Decoded barcode length (excluding prefix and/or suffix codes). See **Set_ReaderOutputConfiguration()** paramenter *showCodeLen*.

Note: Data fields by sequence may include –
[Code Type], [Prefix Code], [Decode Data], [Suffix Code], [Code Length]

Get_ReaderOutputConfiguration

Purpose	Gets the current ouptut data format.
Syntax	**ClResult Get_ReaderOutputConfiguration (ReaderOutputConfiguration** *settings***)**
Parameters	The default value (if there is) for each setting is indicated by an asterisk "*".

KeyboardEmulationType *enableKeyboardEmulation*

[in][out] A value that specifies whether to emulate data as typed text to subsequently pass it to the focused text field of active application.

KeyboardEmulationType. None	Disables keyboard emulation. The decoded data will be sent by the broadcast intent message.
KeyboardEmulationType. InputMethod	*Enables keyboard emulation via input method.
KeyboardEmulationType. KeyEvent	Enables keyboard emulation via key event.

OutputEnterWay *autoEnterWay*

[in][out] A value that specifies whether to auto-affix a character after decoding.

OutputEnterWay.Disable	Disables
OutputEnterWay.SuffixData	*Suffixes the decoded data (= decoded data + Enter-character)
OutputEnterWay.PreffixData	Prefixes the decoded data (= Enter-character + decoded data)

OutputEnterChar *autoEnterChar*

[in][out] A value that specifies the character to auto-affix.

OutputEnter.None	None
OutputEnter.Return	*Carriage Return (= 0x0d)
OutputEnter.Tab	Tab
OutputEnter.Comma	Comma (= 0x2c)
OutputEnter.Semicolon	Semicolon (= 0x3b)

Enable_State *showCodeType*

[in][out] A value that specifies whether to transmit barcode type in data records.

Enable_State.FALSE	*Does not transmit
Enable_State.TRUE	Transmits

Enable_State *showCodeLen*

[in][out] A value that specifies whether to transmit code length for a barcode in data records.

Enable_State.FALSE	*Does not transmit
Enable_State.TRUE	Transmits

String *szPrefixCode*

[in][out] A string variable that stores prefix code.

String *szSuffixCode*

[in][out] A string variable that stores suffix code.

int *useDelim*

[in][out] An ASCII value that specifies the delimiter in use.

0	*No delimiter
1~127	Adds a delimiter between UID and data when both are decoded

String *szCharsetName*

[in][out] A string variable that specifies the current encoding for barcode data.

windows-1252	*Standard ASCII
big5	Traditional Chinese
shift_JIS	Japanese

Enable_State *clearPreviousData*

[in][out] A value that specifies whether to clear barcode data.

Enable_State.FALSE	*Does not clear
Enable_State.TRUE	Clear

Return Value

If successful, it returns ClResult.S_OK.

Otherwise, it returns ClResult.S_ERR.

Example

```
ReaderOutputConfiguration settings = new ReaderOutputConfiguration();
mReaderManager.Get_ReaderOutputConfiguration(settings);
```

Remarks

Depending on reader type and associated reader setting, the fields of output record may differ.

When data comes from barcode reader, data fields may include:
`[Code Type][Prefix Code][Decode Data][Suffix Code][Code Length]`

[Code Type]:	This field is output only when *showCodeType* value is TRUE.
[Prefix Code]:	This field is output only when *szPrefixCode* is non-zero.
[Decode Data]:	This field is output only when *enableKeyboardEmulation* value is InputMethod or KeyEvent.
[Suffix Code]:	This field is output only when *szSuffiixCode* value is non-zero.
[Code Length]:	This field is output only when *showCodeLen* value is TRUE. (Prefix/suffix codes are not included.)

See Also

Set_ReaderOutputConfiguration

Set_ReaderOutputConfiguration

Purpose

Sets the current ouptut data format.

Syntax

ClResult Set_ReaderOutputConfiguration (ReaderOutputConfiguration *settings***)**

Parameters

The default value (if there is) for each setting is indicated by an asterisk "*".

KeyboardEmulationType *enableKeyboardEmulation*

[in][out] A value that specifies whether to emulate data as typed text to subsequently pass it to the focused text field of active application.

KeyboardEmulationType.None	Disables keyboard emulation. The decoded data will be sent by the broadcast intent message.
KeyboardEmulationType.InputMethod	*Enables keyboard emulation via input method.
KeyboardEmulationType.KeyEvent	Enables keyboard emulation via key event.

OutputEnterWay *autoEnterWay*

[in][out] A value that specifies whether to auto-affix a character after decoding.

OutputEnterWay.Disable	Disables
OutputEnterWay.SuffixData	*Suffixes the decoded data (= decoded data + Enter-character)
OutputEnterWay.PreffixData	Prefixes the decoded data (= Enter-character + decoded data)

OutputEnterChar *autoEnterChar*

[in][out] A value that specifies the character to auto-affix.

OutputEnter.None	None
OutputEnter.Return	*Carriage Return (= 0x0d)
OutputEnter.Tab	Tab
OutputEnter.Space	Space (= 0x20)

OutputEnter.Comma	Comma (= 0x2c)
OutputEnter.Semicolon	Semicolon (= 0x3b)

Enable_State *showCodeType*

[in][out] A value that specifies whether to transmit barcode type in data records.

Enable_State.FALSE	*Does not transmit
Enable_State.TRUE	Transmits

Enable_State *showCodeLen*

[in][out] A value that specifies whether to transmit code length for a barcode in data records.

Enable_State.FALSE	*Does not transmit
Enable_State.TRUE	Transmits

String *szPrefixCode*

[in][out] A string variable that stores prefix code.

String *szSuffixCode*

[in][out] A string variable that stores suffix code.

int *useDelim*

[in][out] An ASCII value that specifies the delimiter in use.

0	*No delimiter
1~127	Adds a delimiter between UID and data when both are decoded

String *szCharsetName*

[in][out] A string variable that specifies the current encoding for barcode data.

windows-1252	*Standard ASCII
big5	Traditional Chinese
shift_JIS	Japanese

Enable_State *clearPreviousData*

[in][out] A value that specifies whether to clear barcode data.

Enable_State.FALSE	*Does not clear
Enable_State.TRUE	Clear

Return Value

If successful, it returns ClResult.S_OK.

Otherwise, it returns ClResult.S_ERR.

Example

```
ReaderOutputConfiguration settings = new ReaderOutputConfiguration();
settings.enableKeyboardEmulation =
KeyboardEmulationType.InputMethod;
settings.autoEnterWay = OutputEnterWay.SuffixData;
settings.autoEnterChar = OutputEnterChar.Return;
settings.showCodeLen = Enable_State.TRUE;
settings.showCodeType = Enable_State.TRUE;
settings.szPrefixCode = "PreStr";
settings.szSuffixCode = "SufStr";
settings.useDelim = ':';
settings.szCharsetName = 'shift_JIS';
```

```
settings.clearPreviousData = Enable_State.TRUE;
mReaderManager.Set_ReaderOutputConfiguration(settings);
```

Remarks

Depending on reader type and associated reader setting, the fields of output record may differ.

When data comes from barcode reader, data fields may include:
`[Code Type][Prefix Code][Decode Data][Suffix Code][Code Length]`

[Code Type]:	This field is output only when *showCodeType* value is TRUE.
[Prefix Code]:	This field is output only when *szPrefixCode* is non-zero.
[Decode Data]:	This field is output only when *enableKeyboardEmulation* value is InputMethod or KeyEvent.
[Suffix Code]:	This field is output only when *szSuffiixCode* value is non-zero.
[Code Length]:	This field is output only when *showCodeLen* value is TRUE. (Prefix/suffix codes are not included.)

See Also

Get_ReaderOutputConfiguration

SoftScanTrigger

Purpose

Emulates the behaviour of physical trigger key.

The following steps have to be done beforehand:

1. Register for the CipherLab-specific string – *com.cipherlab.barcode.GeneralString.Intent_SOFTTRIGGER_DATA* – by calling the *android.content.ContextWrapper.registerReceiver* function.
2. Receive the registered string by calling the Android *BroadcastReceiver()* function.
3. Fetch the data from the received Intent.

Syntax

void SoftScanTrigger ();

Example

```
public class MainActivity extends Activity {
private IntentFilter filter;
Button b1 = null;
ReaderManager m_RM = null;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
b1 = (Button) findViewById(R.id.button1);
b1.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
if (m_RM != null)
{
   m_RM.SoftScanTrigger();
}
}
});
m_RM = ReaderManager.InitInstance(this);
filter = new IntentFilter();
filter.addAction(GeneralString.Intent_SOFTTRIGGER_DATA);
registerReceiver(myDataReceiver, filter);
}

@Override
protected void onDestroy() {
// TODO Auto-generated method stub
super.onDestroy();
unregisterReceiver(myDataReceiver);
}
private final BroadcastReceiver myDataReceiver = new
       BroadcastReceiver() {

@Override
public void onReceive(Context context, Intent intent) {
if (intent.getAction().equals(GeneralString.Intent_SOFTTRIGGER_
       DATA))
{
// Fetch data from the intent
String sDataStr = intent.getStringExtra(GeneralString.BcReaderData);
Toast.makeText(MainActivity.this, "Decoded data is " + sDataStr,
Toast.LENGTH_SHORT).show();
}
}
};
}
```

1.3.2. READER SERVICE VERSION

Get_BarcodeServiceVer	
Purpose	Obtains the reader service version.
Syntax	**String Get_BarcodeServiceVer ();**
Example	`String ver = mReaderManager.Get_BarcodeServiceVer();`

1.3.3. READER API VERSION

GetBarcodeAPIVersion	
Purpose	Obtains the reader API version.
Syntax	**String GetBarcodeAPIVersion ();**
Example	`String ver = mReaderManager.GetBarcodeAPIVer();`

1.4. MANIPULATE STATUS INDICATION

The device auto-signals the receipt of a successful decoding by a sound or vibration according to **NotificationParams()** settings. Before calling Set_xxx functions, users are supposed to call Get_xxx functions to retrieve the current reader service settings.

1.4.1. NOTIFICATION SETTINGS

Get_NotificationParams

Purpose: Gets notification settings.

Syntax: **ClResult Get_NotificationParams (NotificationParams** *settings***)**

Parameters: A default value comes with an asterisk "*".

BeepType *ReaderBeep*

[in][out] A value that specifies the sound to play.

BeepType.Mute
BeepType.Default
BeepType.Hwandsw
BeepType.MenuPop
BeepType.MsgBox
BeepType.Notify
BeepType.VoiceBeep
BeepType.Alarm2
BeepType.Alarm3
BeepType.LowBatt

Enable_State *enableVibrator*

[in][out] A value that specifies whether to vibrate for a successful reading.

Enable_State.FALSE	Disables Vibrator*
Enable_State.TRUE	Enables Vibrator

int *vibrationCounter*

[in][out] A value that specifies how long to vibrate.

0	No vibration
1 ~ 10	*1 (in increments of 0.5 seconds)

int *ledDuration*

[in][out] A value that specifies whether to light the LED indicator for a successful reading.

0	*Disables
1 ~ 5000	Enables LED and sets lighting duration (millisecond)

Example

```
NotificationParams settings = new NotificationParams();
mReaderManager.Get_NotificationParams(settings);
```

Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.
See Also	Set_NotificationParams

Set_NotificationParams

Purpose: Configures notification settings.

Syntax: **ClResult Set_NotificationParams (NotificationParams** *settings***)**

Parameters: A default value comes with an asterisk "*".

BeepType *ReaderBeep*

[in][out] A value that specifies the sound to play.

BeepType.Mute
BeepType.Default
BeepType.Hwandsw
BeepType.MenuPop
BeepType.MsgBox
BeepType.Notify
BeepType.VoiceBeep
BeepType.Alarm2
BeepType.Alarm3
BeepType.LowBatt

Enable_State *enableVibrator*

[in][out] A value that specifies whether to vibrate for a successful reading.

Enable_State.FALSE	*Disables Vibrator
Enable_State.TRUE	Enables Vibrator

int *vibrationCounter*

[in][out] A value that specifies how long to vibrate.

0	*No vibration
1 ~ 10	in increments of 0.5 seconds

int *ledDuration*

[in][out] A value that specifies whether to light the LED indicator for a successful reading.

0	Disables
1 ~ 5000	*500, Enables LED and sets lighting duration (millisecond)

Example

```
NotificationParams settings = new NotificationParams();
settings.enableReaderBeep = Enable_State.TRUE;
settings.enableVibrator = Enable_State.TRUE;
settings.ledDuration = 500; //ms
settings.vibrationCounter = 1; //500ms * count
mReaderManager.Set_NotificationParams(settings);
```

Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.
See Also	Get_NotificationParams

1.5. CONFIGURE SCAN ENGINE

1.5.1. PREFERENCES

Get_Decoders_Status

Purpose	Gets symbology status (enabled or disabled) of the reader.
Syntax	**ClResult Get_Decoders_Status (Readers** *settings***)**
Parameters	[in][out] Each value specifies whether the reader is able to decode the corresponding symbology. A default value comes with an asterisk "*".

Enable_State *enableAustrailianPostal*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableAztec*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableCompositeCC_AB*

Enable_State.FALSE	*Disabled
Enable_State.TRUE	Enabled

Enable_State *enableCompositeCC_C*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableCompositeTlc39*

Enable_State.FALSE	*Disabled
Enable_State.TRUE	Enabled

Enable_State *enableCode11*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableCode39*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableCode93*

Enable_State.FALSE	Disabled

Enable_State.TRUE	*Enabled

Enable_State *enableCode128*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableCodabar*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableChinese2Of5*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableDataMatrix*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableDutchPostal*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableEanJan8*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableEanJan13*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableGs1128*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableGs1DataBar14*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableGs1DataBarLimited*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableGs1DataBarExpanded*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableGs1DatabarToUpcEan*

Enable_State.FALSE	Disabled

Enable_State.TRUE	*Enabled

Enable_State *enableIsbt128*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableIndustrial2Of5*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableInterleaved2Of5*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableJapanPostal*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableKorean3Of5*

Enable_State.FALSE	*Disabled
Enable_State.TRUE	Enabled

Enable_State *enableMatrix2Of5*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableMaxiCode*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableMicroPDF417*

Enable_State.FALSE	*Disabled
Enable_State.TRUE	Enabled

Enable_State *enableMicroQR*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableMsi*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enablePDF417*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableQRcode*

Enable_State.FALSE	Disabled

Enable_State.TRUE	*Enabled

Enable_State *enableTriopticCode39*

Enable_State.FALSE	*Disabled
Enable_State.TRUE	Enabled

Enable_State *enableUpcA*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableUpcE*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableUpcE1*

Enable_State.FALSE	*Disabled
Enable_State.TRUE	Enabled

Enable_State *enableUccCoupon*

Enable_State.FALSE	*Disabled
Enable_State.TRUE	Enabled

Enable_State *enableUKPostal*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableUPUFICSPostal*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableUSPostnet*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableUSPlanet*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Enable_State *enableUSPSPostal*

Enable_State.FALSE	Disabled
Enable_State.TRUE	*Enabled

Return Value

If successful, it returns ClResult.S_OK.
Otherwise, it returns ClResult.S_ERR.

Example

```
Decoders settings = new Decoders();
mReaderManager.Get_Decoders_Status(settings);
if (Enable_State.NotSupport == settings.enableAustrailianPostal) {
                    // 1D is not supported
}
```

Remarks	Enable_State.FALSE means the reader is disabled to decode the symbology. Enable_State.TRUE means the reader is enabled to decode the symbology.
See Also	Get_Decoders_Status

Set_Decoders_Status

Purpose	Sets symbology status (enabled or disabled) of the reader.
Syntax	**ClResult Set_Decoders_Status (Decoders** *settings***)**
Parameters	42 symbologies are supported. See above for details.
Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.

Example	`Decoders settings = new Decoders();`
	`settings.enableAustrailianPostal = Enable_State.FALSE;`
	`settings.enableAztec = Enable_State.FALSE;`
	`settings.enableChinese2Of5 = Enable_State.FALSE;`
	`settings.enableCodabar = Enable_State.FALSE;`
	`settings.enableCode11 = Enable_State.FALSE;`
	`settings.enableCode128 = Enable_State.FALSE;`
	`settings.enableCode39 = Enable_State.FALSE;`
	`settings.enableCode93 = Enable_State.FALSE;`
	
	`if (ClResult.S_ERR == mReaderManager.Set_Decoders_Status(settings))`
	`Toast.makeText(this, "Set_Decoders_Status was failed", Toast.LENGTH_SHORT).show();`
	`else`
	`Toast.makeText(this, "Set_Decoders_Status was successful", Toast.LENGTH_SHORT).show();`

Remarks	Enable_State.FALSE means the reader is disabled to decode the symbology. Enable_State.TRUE means the reader is enabled to decode the symbology.
See Also	Set_Decoders_Status

Get_UserPreferences

Purpose	Gets user preference settings of the barcode reader.
Syntax	**ClResult Get_UserPreferences (UserPreference** *settings***)**
Parameters	A default value comes with an asterisk "*". int *addonSecurityLevel* [in][out] A value that specifies decode security level for reading UPC/EAN when "Decode with Addons (=Auto-discriminate)" is applied.

2~30	*10 (times of supplementary decoding)

Enable_State *displayMode*

[in][out] A value that specifies whether to enable display mode.

Enable_State.FALSE	*Disables

Enable_State.TRUE	Enables

int *laserOnTime*

[in][out] A value that specifies the maximum time for decoding a printed barcode during a scan act.

500 ~ 9900	*3000 (millisecond)

InverseType *negativeBarcodes*

[in][out] A value that specifies the negative barcode status.

InverseType.RegularOnly	*Regular barcode
InverseType.InverseOnly	Negative barcode
InverseType.AutoDetect	AutoDetection

Enable_State *pickListMode*

[in][out] A value that specifies whether to enable picklist mode for decoding accuracy.

Enable_State.FALSE	*Disables
Enable_State.TRUE	Enables

RedundancyLevel *redundancyLevel*

[in][out] A value that specifies decode redundancy. Higher redundancy levels should be selected for deteriorated barcode quality.

RedundancyLevel.One	*The following barcodes must be successfully read twice before being decoded: Barcode Types / Code Length Codabar / All MSI / 4 characters or less Industrial 25 (Discrete 25) / 8 characters or less Interleaved 25 / 8 characters or less
RedundancyLevel.Two	All barcodes must be successfully read twice before being decoded.
RedundancyLevel.Three	All barcodes must be successfully read twice before being decoded except for the following that must be read three times: Barcode Types "Excluded" / Code Length MSI / 4 characters or less Industrial 25 (Discrete 25) / 8 characters or less Interleaved 25 / 8 characters or less
RedundancyLevel.Four	All barcodes must be successfully read three times before being decoded.

ScanAngleType *scanAngle*

[in][out] A value that specifies scan angle.

ScanAngleType.Narrow	Narrow Angle (35 degrees)
ScanAngleType.Wide	*Wide Angle (47 degrees)

SecurityLevel *securityLevel*

[in][out] A value that specifies decode security level, which is useful to fix some printed quality issues when reading delda barcodes such as Code 128, Code 93, UPC/EAN.

SecurityLevel.Zero	*Security Level 0 – The default. It allows the scan engine to operate aggressively enough to decode most “in-spec” barcodes.
SecurityLevel.One	Security Level 1 – Select this option if misdecodes occur. This level should fix most misdecodes.
SecurityLevel.Two	Security Level 2 – Select this option if Security Level 1 fails to fix misdecodes.
SecurityLevel.Three	Security Level 3 – Select this option if Security Level 2 also fails to fix misdecodes. However, selecting this option impairs the decoding ability of the scan engine. If this level of security is necessary, try to improve the barcode quality.

int *timeoutBetweenSameSymbology*

[in][out] A value that specifies the minimum time interval between reading two identical barcodes. This helps prevent the scanner from accidentally reading the same barcode twice.

This parameter applies to Continuous and Presentation modes.

0 ~ 9900	*1000 (millisecond)

int *timeoutPresentationMode*

[in][out] A value that specifies the time to enable the Prensentation mode.

This parameter applies to Continuous mode.

60000 ~ 1800000	*900000 (millisecond) (default: 15 minutes, ranging from 1 to 30 minutes)

TransmitCodeIDType *transmitCodeIdChar*

[in][out] A value that specifies whether to transmit Code ID characters.

TransmitCodeIDType.None	*Transmits none
TransmitCodeIDType.AimCodeID	Transmits AIM Code ID Character

TriggerType *triggerMode*

[in][out] A value that specifies the mode to scan.

TriggerType.LevelMode	*Level mode
TriggerType.ContinuousMode	Continuous mode
TriggerType.PresentationMode	Presentation mode
TriggerType.AutoAimMode	Auto Aim mode

Enable_State *decodingillumniation*

[in][out] A value that specifies whether to provide flash illumination during every barcode capture to aid decoding.

Enable_State.FALSE	Disables decding illumination
Enable_State.TRUE	*Enables decding illumination

IlluminationPowerLevel *decodingilluminationPowerLevel*

[in][out] A value that specifies the power level of illumination.

IlluminationPowerLevel.Zero ~ IlluminationPowerLevel.Ten	*IlluminationPowerLevel.Ten – The default is set to Ten which is the maximum power level.

Enable_State *decodingAimingPattern*

[in][out] A value that specifies whether to project the aiming pattern on every barcode capture.

Enable_State.FALSE	Disables decode aming pattern
Enable_State.TRUE	*Enables decode aming pattern

InterCharacterGapSize *interCharGapSize*

[in][out] A value that specifies the intercharacter gap size for Code 39 and Codabar, which is typically quite small. Due to various barcode printing technologies, this gap can grow larger than the maximum size allowed and prevent the scan engine from decoding a barcode. If this problem occurs, set it to “Large Intercharacter Gaps” to tolerate these out-of-specification barcodes.

InterCharacterGapSize.Normal	*Normal intercharacter gaps
InterCharacterGapSize.Large	Large intercharacter gaps

Return Value

If successful, it returns ClResult.S_OK.

Otherwise, it returns ClResult.S_ERR.

Example

```
UserPreference settings = new UserPreference();
mReaderManager.Get_UserPreferences(settings);
if (Enable_State.NotSupport == settings.displayMode)
{

}
```

See Also

Set_UserPreferences

Set_UserPreferences

Purpose

Sets user preference settings of the barcode reader.

Syntax

ClResult Set_UserPreferences (UserPreference *settings***)**

Parameters

11 preference settings are listed below. See above for details.

int *addonSecurityLevel*

Enable_State *displayMode*

int *laserOnTime*

InverseType *negativeBarcodes*

Enable_State *pickListMode*

RedundancyLevel *redundancyLevel*

ScanAngleType *scanAngle*

	SecurityLevel *securityLevel* int *timeoutBetweenSameSymbology* int *timeoutPresentationMode* TransmitCodeIDType *transmitCodeIdChar* TriggerType *triggerMode* Enable_State *decodingillumniation* IlluminationPowerLevel *decodingillumniationPowerLevel* Enable_State *decodingAimingPattern* InterCharacterGapSize *interCharGapSize*
Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.

Example	`UserPreference settings = new UserPreference();`
	
	`settings.addonSecurityLevel = 2;`
	`settings.laserOnTime = 3000;`
	`settings.negativeBarcodes = InverseType.AutoDetect;`
	`settings.scanAngle = ScanAngleType.Wide;`
	
	`if (ClResult.S_ERR == mReaderManager.Set_UserPreferences(settings))`
	`Toast.makeText(this, "Set_UserPreferences was failed", Toast.LENGTH_SHORT).show();`
	`else`
	`Toast.makeText(this, "Set_UserPreferences was successful", Toast.LENGTH_SHORT).show();`

See Also	Get_UserPreferences

1.5.2. SYMBOLOGY SETTINGS

Get_Symbology	
Purpose	Gets symbology settings by implementing a symbology interface instance.
Syntax	**ClResult Get_Symbology (SymbologyInterface** *settings***)**
Parameters	This function provides 38 symbology parameters implemented as objects. *Codabar* *{* *Enable_State transmitCheckDigit;* *CodabarDigitAlgorithm verifyCheckDigit;* *NOTISEditingType notisEditingType;* *Enable_State enable;* *int length1;* *int length2;* *Enable_State clsiEditing;* *Enable_State notisEditing;* *}* *Code11*

```
{
Enable_State enable;
int length1;
int length2;
NumberOfCheck numberOfCheckDigits;
Enable_State transmitCheckDigit;
}

Code39
{
Enable_State enable;
int length1;
int length2;
Enable_State checkDigitVerification;
Enable_State transmitCheckDigit;
Enable_State fullASCII;
Enable_State convertToCode32;
Enable_State convertToCode32Prefix;
}

TriopticCode39
{
Enable_State enable
}

Korean3Of5
{
Enable_State enable
}

Code93
{
Enable_State enable;
int length1;
int length2;
}

Code128
{
Enable_State enable
}

GS1128
{
Enable_State enable;
String fieldSeparator;      //ranging from 0 to 127
```

```
}

ISBT128
{
Enable_State enable;
ISBTConcatenationType concatenation;
int concatenationRedundancy;       //ranging from 2 to 20
}

Chinese2Of5
{
Enable_State enable;
}

Industrial2Of5
{
Enable_State enable;
int length1;
int length2;
}

Interleaved2Of5
{
Enable_State enable;
int length1;
int length2;
Enable_State checkDigitVerification;
Enable_State transmitCheckDigit;
Enable_State convertToEan13;
}

Matrix2Of5
{
Enable_State enable;
int length1;
int length2;
int redundancy;
Enable_State checkDigitVerification;
Enable_State transmitCheckDigit;
}

UccCoupon
{
Enable_State enable;
}
```

GS1DataBar14
{
Enable_State enable;
Enable_State convertToUpcEan;
}

GS1DataBarLimited
{
int securityLevel;
Enable_State enable;
Enable_State convertToUpcEan;
}
Note: The convertToUpcEan elements included in GS1DataBar14 and GS1dataBarLimited symbologies correspond to each other, which means changing either of the element value of a particular symbology will also change the other one.

GS1DataBarExpanded
{
Enable_State enable;
String fieldSeparator; //ranging from 0 to 127
}

Msi
{
Enable_State enable;
int length1;
int length2;
MsiDigitOption checkDigitOption;
Enable_State transmitCheckDigit;
DigitAlgorithm checkDigitAlgorithm;
}

Ean8
{
Enable_State enable;
AddonsType addon2;
AddonsType addon5;
Enable_State transmitCheckDigit;
Enable_State convertToEan13;
}

Ean13
{
Enable_State enable;
AddonsType addon2;
AddonsType addon5;

Enable_State convertToISBN;
Enable_State convertToISSN;
ISBNFormat booklandISBNFormat;
Enable_State transmitCheckDigit;
}
Note: The addon2, addon5, and transmitCheckDigit elements included in Ean8 and Ean13 symbologies correspond to each other respectively, which means changing either of the element value of a particular symbology will also change the other one.

UpcA
{
Enable_State enable;
AddonsType addon2;
AddonsType addon5;
Enable_State transmitCheckDigit;
Preamble transmitSystemNumber;
Enable_State convertToEan13
}

UpcE
{
Enable_State enable;
AddonsType addon2;
AddonsType addon5;
Enable_State transmitCheckDigit;
Preamble transmitSystemNumber;
Enable_State convertToUpcA;
}

UpcE1
{
Enable_State enable;
AddonsType addon2;
AddonsType addon5;
Enable_State transmitCheckDigit;
Preamble transmitSystemNumber;
Enable_State convertToUpcA;
}
Note: The addon2 an addon5 elements included in UpcA, UpcE, and UpcE1 symbologies correspond to each other respectively, which means changing either of the element value of a particular symbology will also change the others.

Composite
{
Enable_State enableCc_C;
Enable_State enableCc_AB;

Enable_State enableTlc39;
UpcMode enableUpcMode;
Enable_State enableEmulationMode;
}

USPostal
{
Enable_State enablePlanet;
Enable_State enablePostnet;
Enable_State transmitCheckDigit;
}

UKPostal
{
Enable_State enable;
Enable_State transmitCheckDigit;
}

JapanPostal
{
Enable_State enable;
}

AustralianPostal
{
Enable_State enable;
}

DutchPostal
{
Enable_State enable;
}

USPSPostal
{
Enable_State enable;
}

UPUFICSPostal
{
Enable_State enable;
}

PDF417
{
Enable_State enable;

TransmitMode transmitMode;
char escapeCharacter;
Enable_State transmitControlHeader;
}

MicroPDF417
{
Enable_State enable;
Enable_State code128Emulation;
}

DataMatrix
{
Enable_State enable;
String fieldSeparator; //ranging from 0 to 127
MatrixMirrorImage mirrorImage;
}

MaxiCode
{
Enable_State enable;
}

QRCode
{
Enable_State enable;
}

MicroQR
{
Enable_State enable;
}

Aztec
{
Enable_State enable;
}

Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.
Remarks	*length1* and *length2* are integers ranging from 0 to 55

Example	`Codabar settings = new Codabar();`
	`if (ClResult.Err_NotSupport == mReaderManager.Get_Symbology(settings))`
	`{`
	`  // to verify whether the symbology is supported`
	`}`

```
// if disabled, enable it and then configure it via Set_Symbology
if (Codabar.enable == Enable_State.FALSE)
{
    Codabar.enable = Enable_State.TRUE;
}
```

See Also	Set_Symbology

Set_Symbology	
Purpose	Sets symbology settings by implementing a symbology interface instance.
Syntax	**ClResult Set_Symbology (SymbologyInterface** *settings***)**
Parameters	This function provides 38 symbology parameters implemented as objects. See above for details.
Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.

Example	`Codabar settings = new Codabar();`
	`Codabar.enable = Enable_State.TRUE;`
	`if (ClResult.S_ERR == mReaderManager.Set_Symbology(settings))`
	`Toast.makeText(this, "Set_Symbology was failed", Toast.LENGTH_SHORT).show();`
	`else`
	`Toast.makeText(this, "Set_Symbology was successful", Toast.LENGTH_SHORT).show();`

See Also	Get_Symbology

1.5.3. CODABAR CLASS

```
public class Codabar
{
    public Enable_State transmitCheckDigit;
    public CodabarDigitAlgorithm verifyCheckDigit;
    public NOTISEditingType notisEditingType;
    public Enable_State    enable;
    public int             length1;
    public int             length2;
    public Enable_State    clsiEditing;
    public Enable_State    notisEditing;
    public Enable_State transmitCheckDigit;
}
```

Data Type	Member Name	Description
Enable_State	transmitCheckDigit	A value specifying whether to transmit check digit. *TRUE FALSE

CodabarDigitAlgorithm	verifyCheckDigit	A value specifying whether and how to verify check digit. *None Modulo_16 Modulo_7DR Modulo_Both Note: For Modulo_7DR, the total number of digits in a codabar cannot be greater than '19'; and the first digit in a codabar must be equal to or less than '8'.
NOTISEditingType	notisEditingType	A value specifying whether to transform it to NOTIS editing format (Start/Stop characters) and the way it is transformed. *None ABCD_Upper Abcd_Lower
Enable_State	enable	A value specifying whether to enable Codabar. *TRUE FALSE
int	length1	Length qualification *4 (0 ~ 55)
int	length2	Length qualification *55 (0 ~ 55)
Enable_State	clsiEditing	A value specifying whether to edit CLSI. TRUE *FALSE
Enable_State	notisEditing	(Reserved)
Enable_State	transmitCheckDigit	A value specifying whether to transmit check digit. *TRUE FALSE

1.5.4. CODE11 CLASS

```
public class Code11
{
    public Enable_State enable;
    public int length1;
    public int length2;
    public NumberOfCheck numberOfCheckDigits;
    public Enable_State transmitCheckDigit;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value specifying whether to enable Code 11. *TRUE FALSE
int	length1	Length qualification *4 (0 ~ 55)
int	length2	Length qualification *55 (0 ~ 55)
NumberOfCheck	numberOfCheckDigits	A value specifying whether and how to verify check digit. *None One Two
Enable_State	transmitCheckDigit	A value specifying whether to transmit check digit. TRUE *FALSE

1.5.5. CODE39 CLASS

```
public class Code39
{
    public Enable_State enable;
    public Enable_State checkDigitVerification;
    public Enable_State transmitCheckDigit;
    public Enable_State fullASCII;
    public Enable_State convertToCode32;
    public Enable_State convertToCode32Prefix;
    public int length1;
    public int length2;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Code 39. TRUE *FALSE
Enable_State	checkDigitVerification	A value that specifies whether to verify check digit. *TRUE FALSE
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. TRUE *FALSE
Enable_State	fullASCII	A value that specifies whether to support Code 39 Full ASCII. TRUE *FALSE
Enable_State	convertToCode32	A value that specifies whether to convert Code 39 to Code 32 (= Italian Pharmacode). TRUE *FALSE
Enable_State	convertToCode32Prefix	A value that specifies whether to transmit prefix for Code 32 data. TRUE *FALSE
int	length1	Length qualification *4 (0 ~ 55)
int	length2	Length qualification

		*55 (0 ~ 55)

1.5.6. TRIOPTICCODE39 CLASS

```
public class TriopticCode39
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable TripoticCode39. TRUE *FALSE

1.5.7. KOREAN3OF5 CLASS

```
public class Korean3Of5
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Korean 3 of 5. *TRUE FALSE

1.5.8. CODE93 CLASS

```
public class Code93
{
    public Enable_State enable;
    public int length1;
    public int length2;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Code 93. *TRUE FALSE
int	length1	Length qualification *4 (0 ~ 55)
int	length2	Length qualification

		*55 (0 ~ 55)

1.5.9. CODE128 CLASS

```
public class Code128
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Code 128. *TRUE FALSE

1.5.10. GS1128 CLASS

```
public class GS1128
{
    public Enable_State enable;
    public String fieldSeparator;
    public Enable_State enableApplicationIdentifier; // Default is disable
    public String applicationIdentifierMark1;
    public String applicationIdentifierMark2;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable GS1 128. *TRUE FALSE
String	fieldSeparator	A value that specifies whether to apply a field separator of ASCII ranging from 0 to 127. It's set to zero by default.
Enable_State	enableApplicationIdentifier	A value that specifies whether to enable the application identifier. TRUE *FALSE
String	applicationIdentifierMark1	Application ID mark 1
String	applicationIdentifierMark2	Application ID mark 2

1.5.11. ISBT128 CLASS

```
public class ISBT128
{
    public Enable_State enable;
    public ISBTConcatenationType concatenation;
    public int concatenationRedundancy;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable ISBT 128. *TRUE FALSE
ISBTConcatenationType	concatenation	A value that specifies whether to decode and concatenate pairs of ISBT barcodes. Disable Enable *Auto
int	concatenationRedundancy	A value that specifies concatenation redundancy (2~20 times) when auto-discriminate of ISBT concatenation is enabled. By default, it is set to 10 times.

1.5.12. CHINESE2OF5 CLASS

```
public class Chinese2Of5
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Chinese 25. *TRUE FALSE

1.5.13. INDUSTRIAL2OF5 CLASS

```
public class Industrial2Of5
{
    public Enable_State enable;
    public int length1;
    public int length2;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Industrial 25. *TRUE FALSE
int	length1	Length qualification *4 (0 ~ 55)
int	length2	Length qualification *55 (0 ~ 55)

1.5.14. INTERLEAVED2OF5 CLASS

```
public class Interleaved 25
{
    public Enable_State enable;
    public int length1;
    public int length2;
    public I20f5CheckDigitVerification checkDigitVerification;
    public Enable_State transmitCheckDigit;
    public Enable_State convertToEan13;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Interleaved 25. *TRUE FALSE
int	length1	Length qualification *4 (0 ~ 55)
int	length2	Length qualification *55 (0 ~ 55)
I20f5CheckDigitVerification	checkDigitVerification	A value that specifies whether and how to verify check digit.

		*Disable USS OPCC
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. It is available only when I20f5CheckDigitVerification is not disabled. TRUE *FALSE
Enable_State	convertToEan13	A value that specifies whether to convert Interleaved 25 to EAN-13. TRUE *FALSE

1.5.15. MATRIX2OF5 CLASS

```
public class Matrix 25
{
    public Enable_State enable;
    public int length1;
    public int length2;
    public Enable_State redundancy;
    public Enable_State checkDigitVerification;
    public Enable_State transmitCheckDigit;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Matrix 25. *TRUE FALSE
int	length1	Length qualification *4 (0 ~ 55)
int	length2	Length qualification *55 (0 ~ 55)
Enable_State	redundancy	A value that specifies whether to enable decode redundancy. TRUE *FALSE
Enable_State	checkDigitVerification	A value that specifies whether to verify check digit. TRUE *FALSE

Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. TRUE *FALSE

1.5.16. UCCCOUPON CLASS

```
public class UccCoupon
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable UCC Coupon. TRUE *FALSE

1.5.17. GS1DATABAR14 CLASS

```
public class GS1DataBar14
{
    public int securityLevel;
    public Enable_State enable;
    public Enable_State convertToUpcEan;
}
```

Data Type	Member Name	Description
int	securityLevel	A value that specifies the decode security level while reading GS1 DataBar 14. *1 (0 ~ 3)
Enable_State	enable	A value that specifies whether to enable GS1 DataBar-14. *TRUE FALSE
Enable_State	convertToUpcEan	A value that specifies whether to convert RSS to UPC/EAN barcodes. TRUE *FALSE

1.5.18. GS1DATABARLIMITED CLASS

```
public class GS1DataBarLimited
{
    public int securityLevel;
    public Enable_State enable;
    public Enable_State convertToUpcEan;
}
```

Data Type	Member Name	Description
int	securityLevel	A value that specifies the decode security level while reading GS1 DataBar Limited. *3 (1 ~ 4)
Enable_State	enable	A value that specifies whether to enable GS1 DataBar Limited. *TRUE FALSE
Enable_State	convertToUpcEan	A value that specifies whether to convert RSS to UPC/EAN barcodes. TRUE *FALSE

1.5.19. GS1DATABAREXPANDED CLASS

```
public class GS1DataBarExpanded
{
    public int securityLevel;
    public Enable_State enable;
    public String fieldSeparator;
}
```

Data Type	Member Name	Description
int	securityLevel	A value that specifies the decode security level while reading GS1 DataBar Expanded. *1 (0 ~ 3)
Enable_State	enable	A value that specifies whether to enable GS1 DataBar Expanded. *TRUE FALSE
String	fieldSeparator	A value that specifies whether to apply a field separator of ASCII ranging from 0 to 127. It's set to zero by default.

1.5.20. MSI CLASS

```
public class Msi
{
    public Enable_State enable;
    public int length1;
    public int length2;
    public MsiDigitOption checkDigitOption;
    public Enable_State transmitCheckDigit;
    public DigitAlgorithm checkDigitAlgorithm;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable MSI. *TRUE FALSE
int	length1	Length qualification *4 (0 ~ 55)
int	length2	Le ngth qualification *55 (0 ~ 55)
MsiDigitOption	checkDigitOption	A value that specifies how to verify check digit. *Onedigit TwoDigits
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. *TRUE FALSE
DigitAlgorithm	checkDigitAlgorithm	A value that specifies which algorithm to apply. *Modulo_10_11 DoubleModulo_10

1.5.21. EAN8 CLASS

```
public class Ean8
{
    public Enable_State enable;
    public AddonsType addon2;
    public AddonsType addon5;
    public Enable_State transmitCheckDigit;
    public Enable_State convertToEan13;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable EAN-8. *TRUE FALSE
AddonsType	addon2	A value that specifies the way processing addon2. *IgnoresAddon AutoDiscriminate
AddonsType	addon5	A value that specifies the way processing addon5. *IgnoresAddon AutoDiscriminate
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. *TRUE FALSE
Enable_State	convertToEan13	A value that specifies whether to convert EAN-8 to EAN-13. TRUE *FALSE

1.5.22. EAN13 CLASS

```
public class Ean13
{
    public Enable_State enable;
    public AddonsType addon2;
    public AddonsType addon5;
    public Enable_State convertToISBN;
    public Enable_State convertToISSN;
    public ISBNFormat booklandISBNFormat;
    public Enable_State transmitCheckDigit;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable EAN-13. *TRUE FALSE
AddonsType	addon2	A value that specifies the way processing addon2. *IgnoresAddon AutoDiscriminate
AddonsType	addon5	A value that specifies the way processing addon5. *IgnoresAddon AutoDiscriminate
Enable_State	convertToISBN	A value that specifies whether to convert EAN-13 to ISBN. TRUE *FALSE
Enable_State	convertToISSN	A value that specifies whether to convert EAN-13 to ISSN. TRUE *FALSE
ISBNFormat	booklandISBNFormat	If you enabled Bookland EAN, select one of the following formats for Bookland data. ISBN_10 ISBN_13
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. *TRUE FALSE

1.5.23. UPCA CLASS

```
public class Upca
{
    public Enable_State enable;
    public AddonsType addon2;
    public AddonsType addon5;
    public Enable_State transmitCheckDigit;
    public Preamble transmitSystemNumber;
    public Enable_State convertToEan13;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable UPC-A. *TRUE FALSE
AddonsType	addon2	A value that specifies the way processing addon2. *IgnoresAddon AutoDiscriminate
AddonsType	addon5	A value that specifies the way processing addon5. *IgnoresAddon AutoDiscriminate
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. *TRUE FALSE
Preamble	transmitSystemNumber	A value that specifies whether to verify (transmit) UPC-A preamble. None *SysNumOnly SysNumAndCtyCode
Enable_State	convertToEan13	A value that specifies whether to convert to EAN-13. TRUE *FALSE

1.5.24. UPCE CLASS

```
public class Upce
{
    public Enable_State enable;
    public AddonsType addon2;
    public AddonsType addon5;
    public Enable_State transmitCheckDigit;
    public Preamble transmitSystemNumber;
    public Enable_State convertToUpcA;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable UPC-E. *TRUE FALSE
AddonsType	addon2	A value that specifies the way processing addon2. *IgnoresAddon AutoDiscriminate
AddonsType	addon5	A value that specifies the way processing addon5. *IgnoresAddon AutoDiscriminate
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. *TRUE FALSE
Preamble	transmitSystemNumber	A value that specifies whether to verify (transmit) UPC-E preamble. None *SysNumOnly SysNumAndCtyCode
Enable_State	convertToUpcA	A value that specifies whether to convert to UPC-A. TRUE *FALSE

1.5.25. UPCE1 CLASS

```
public class UpcE1
{
    public Enable_State enable;
    public AddonsType addon2;
    public AddonsType addon5;
    public Enable_State transmitCheckDigit;
    public Preamble transmitSystemNumber;
    public Enable_State convertToUpcA;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable UPC-E1. *TRUE FALSE
AddonsType	addon2	A value that specifies the way processing addon2. *IgnoresAddon AutoDiscriminate
AddonsType	addon5	A value that specifies the way processing addon5. *IgnoresAddon AutoDiscriminate
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. *TRUE FALSE
Preamble	transmitSystemNumber	A value that specifies whether to verify (transmit) UPC-E1 preamble. None *SysNumOnly SysNumAndCtyCode
Enable_State	convertToUpcA	A value that specifies whether to convert to UPC-A. TRUE *FALSE

1.5.26. COMPOSITE CLASS

```
public class Composite
{
    public Enable_State enableCc_C;
    public Enable_State enableCc_AB;
    public Enable_State enableTlc39;
    public UpcMode enableUpcMode;
    public Enable_State enableEmulationMode;
}
```

Data Type	Member Name	Description
Enable_State	enableCc_C	A value that specifies whether to enable Composite CC-C. *TRUE FALSE
Enable_State	enableCc_AB	A value that specifies whether to enable Composite CC-A/B. TRUE *FALSE
Enable_State	enableTlc39	A value that specifies whether to enable Composite TLC-39 (=TCIF Linked). TRUE *FALSE
UpcMode	enableUpcMode	A value that specifies whether to enable UPC Composite Mode. NeverLinksUPC *AlwaysLinksUPC Auto
Enable_State	enableEmulationMode	A value that specifies whether to enable GS1-128 Emulation Mode for UCC/EAN Composite Codes. TRUE *FALSE

1.5.27. USPOSTAL CLASS

```
public class USPostal
{
    public Enable_State enablePlanet;
    public Enable_State enablePostnet;
    public Enable_State transmitCheckDigit;
}
```

Data Type	Member Name	Description
Enable_State	enablePlanet	A value that specifies whether to enable US Planet. *TRUE FALSE
Enable_State	enablePostnet	A value that specifies whether to enable US Postnet. *TRUE FALSE
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. *TRUE FALSE

1.5.28. UKPOSTAL CLASS

```
public class UKPostal
{
    public Enable_State enable;
    public Enable_State transmitCheckDigit;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable UK Postal. *TRUE FALSE
Enable_State	transmitCheckDigit	A value that specifies whether to transmit check digit. *TRUE FALSE

1.5.29. JAPANPOSTAL CLASS

```
public class JapanPostal
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Japan Postal. *TRUE FALSE

1.5.30. AUSTRALIANPOSTAL CLASS

```
public class AustralianPostal
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Australian Postal. *TRUE FALSE

1.5.31. DUTCHPOSTAL CLASS

```
public class DutchPostal
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Dutch Postal. *TRUE FALSE

1.5.32. USPSPOSTAL CLASS

```
public class USPSPostal
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable USPS Postal. *TRUE FALSE

1.5.33. UPUFICSPOSTAL CLASS

```
public class UPUFICSPostal
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable UPUFICS Postal. *TRUE FALSE

1.5.34. PDF417 CLASS

```
public class PDF417
{
    public Enable_State enable;
    public TransmitMode transmitMode;
    public Enable_State escapeCharacter;
    public Enable_State transmitControlHeader;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable PDF417. *TRUE FALSE
TransmitMode	transmitMode	A value that specifies how to handle decoding. BufferAllSymbols TransmitAnySymbolInSet

		*PassthroughAllSymbols
Enable_State	escapeCharacter	A value that specifies whether to use the escape character. TRUE *FALSE
Enable_State	transmitControlHeader	A value that specifies whether to transmit the control header. TRUE *FALSE

1.5.35. MICROPDF417 CLASS

```
public class MicroPDF417
{
    public Enable_State enable;
    public Enable_State code128Emulation;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable MicroPDF417. TRUE *FALSE
Enable_State	code128Emulation	A value that specifies whether to enable Code 128 Emulation for certain MicroPDF417 barcods. TRUE *FALSE

1.5.36. DATAMATRIX CLASS

```
public class DataMatrix
{
    public Enable_State enable;
    public String fieldSeparator;
    public MatrixMirrorImage mirrorImage;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Data Matrix. *TRUE FALSE
String	fieldSeparator	A value that specifies whether to apply a field separator of ASCII ranging from 0 to 127. It's set to zero by default.

MatrixMirrorImage	mirrorImage	A value that specifies whether to decode mirror image Data Matrix barcodes. *Never Always Auto

1.5.37. MAXICODE CLASS

```
public class MaxiCode
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Maxicode. *TRUE FALSE

1.5.38. QRCODE CLASS

```
public class QRCode
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable QR Code. *TRUE FALSE

1.5.39. MICROQR CLASS

```
public class MicroQR
{
    public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable MicroQR. *TRUE FALSE

1.5.40. AZTEC CLASS CLASS

```
public class Aztec
{
   public Enable_State enable;
}
```

Data Type	Member Name	Description
Enable_State	enable	A value that specifies whether to enable Aztec. *TRUE FALSE

1.6. RESET READER

ResetReaderToDefault

Purpose	Resets reader module(s).
Syntax	**ClResult ResetReaderToDefault ()**

Example	`if (ClResult.S_ERR == mReaderManager.ResetReaderToDefault())`
	`{`
	`Toast.makeText(this, "ResetReaderToDefault was failed",`
	`Toast.LENGTH_SHORT).show();`
	`}`
	`else {`
	`Toast.makeText(this, "ResetReaderToDefault was done!",`
	`Toast.LENGTH_SHORT).show();`
	`}`

Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.
Remarks	It takes approximately 2 seconds to reset reader(s) to default.
See Also	InitReader

1.7. INTENT

1.7.1. READER SERVICE CONNECTION

GeneralString.Intent_READERSERVICE_CONNECTED

Purpose: After running InitInstance, the system makes connection between the application and the reader service. With success in making connection, this intent is sent.

1.7.2. SOFTWARE TRIGGER

GeneralString.Intent_SOFTTRIGGER_DATA

Purpose: By calling SoftScanTrigger() to scan the barcode with success, this intent is used to inform the application. Parameters supported are as follows.

Parameters:

GeneralString.BcReaderData	Decode data (Post-Process by Reder Service)
GeneralString.BcReaderData Array	Raw decode data
GeneralString.BcReaderCode Type	Decode code type (refer to Appendix III Code Type & Symbology)
GeneralString.BcReaderCode TypeStr	Decode code type string

Syntax:

getStringExtra(GeneralString.BcReaderData)

getByteArrayExtra(GeneralString.BcReaderDataArray)

getIntExtra(GeneralString.BcReaderCodeType)

getStringExtra(GeneralString.BcReaderCodeTypeStr)

1.7.3. HARDWARE SCAN KEY

GeneralString.Intent_PASS_TO_APP

Purpose: When using the hardware scan key to read the barcode with success, this intent is used to inform the application (Keyboard Emulator is not enabled). Parameters supported are as follows.

Parameters:

GeneralString.BcReaderData	Decode data (Post-Process by Reder Service)
GeneralString.BcReaderData Array	Raw decode data
GeneralString.BcReaderCode Type	Decode code type (refer to Appendix III Code Type & Symbology)
GeneralString.BcReaderCode TypeStr	Decode code type string

Syntax:

getStringExtra(GeneralString.BcReaderData)

getByteArrayExtra(GeneralString.BcReaderDataArray)

getIntExtra(GeneralString.BcReaderCodeType)

getStringExtra(GeneralString.BcReaderCodeTypeStr)

1.7.4. DECODING ERROR

GeneralString.Intent_DECODE_ERROR		
Purpose	If an error occurs in decoding the barcode, this intent is used to inform the application.	
Parameters	**GeneralString.BcReaderDecodeError**	Only for Codabar check digit verification. (DecodeErrorType.ERROR_Codabar_Verify_Check_Digit)
Syntax	**getIntExtra(GeneralString.BcReaderDecodeError)**	

1.7.5. DATA SENDING

The table below depicts the data sent by intents or Keyboard Emulator depending on the conditions.

Keyboard Emulator ON/OFF	Software Trigger	HW Scan Key
On	Data is sent by Intent (Intent_SOFTTRIGGER_DATA) instead of Keyboard Emulator.	Data is sent by Keyboard Emulator.
Off	Data is sent by Intent (Intent_SOFTTRIGGER_DATA).	Data is sent by Intent (Intent_PASS_TO_APP).

1.8. CALLBACK

When the callback function is enabled, the intent is then disabled.

1.8.1. SETREADERCALLBACK

SetReaderCallback	
Purpose	Set the reader callback function.
Syntax	**void SetReaderCallback(ReaderCallback** *readerCallback***);**

Example	`interface ReaderCallback`
	`{`
	`  void onDecodeComplete(in String strDecodeData);`
	`}`

See Also	GetReaderCallbackStatus

1.8.2. GETREADERCALLBACK

GetReaderCallbackStatus	
Purpose	Verify whether the callback function is enabled.
Syntax	**boolean GetReaderCallbackStatus();**
Return Value	If the callback function is enabled, it returns true. Otherwise, it returns false.
See Also	SetReaderCallback

1.8.3. SAMPLE CODE

```
public class MainActivity extends Activity implements ReaderCallback {

private ReaderCallback mReaderCallback = null;

@Override
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);

 e1 = (EditText)findViewById(R.id.editText1);

 // ************************************************//
 // Need to get ReaderManager instance first，or you can't call any APIs of ReaderManager
 // ************************************************//
 mReaderManager = ReaderManager.InitInstance(this);
 mReaderCallback = this;

 // ************************************************//
 // Register an IntentFilter
 // Add GeneralString.Intent_READERSERVICE_CONNECTED for knowing apk is connected with
 // Barcode Reader Service
 // ************************************************//
 filter = new IntentFilter();
```

```
filter.addAction(GeneralString.Intent_READERSERVICE_CONNECTED);
registerReceiver(myDataReceiver, filter);

/// create a BroadcastReceiver for receiving intents from barcode reader service
private final BroadcastReceiver myDataReceiver = new BroadcastReceiver() {
   @Override
   public void onReceive(Context context, Intent intent) {

      if(intent.getAction().equals(GeneralString.Intent_READERSERVICE_CONNECTED)){
          // Make sure this app bind to barcode reader service , then user can use APIs
          // to get/set settings from barcode reader service

          BcReaderType myReaderType = mReaderManager.GetReaderType();
          e1.setText(myReaderType.toString());

          if(mReaderCallback != null)
          {
              // Enable Callback function
              mReaderManager.SetReaderCallback(mReaderCallback);
          }
      }
   }
};

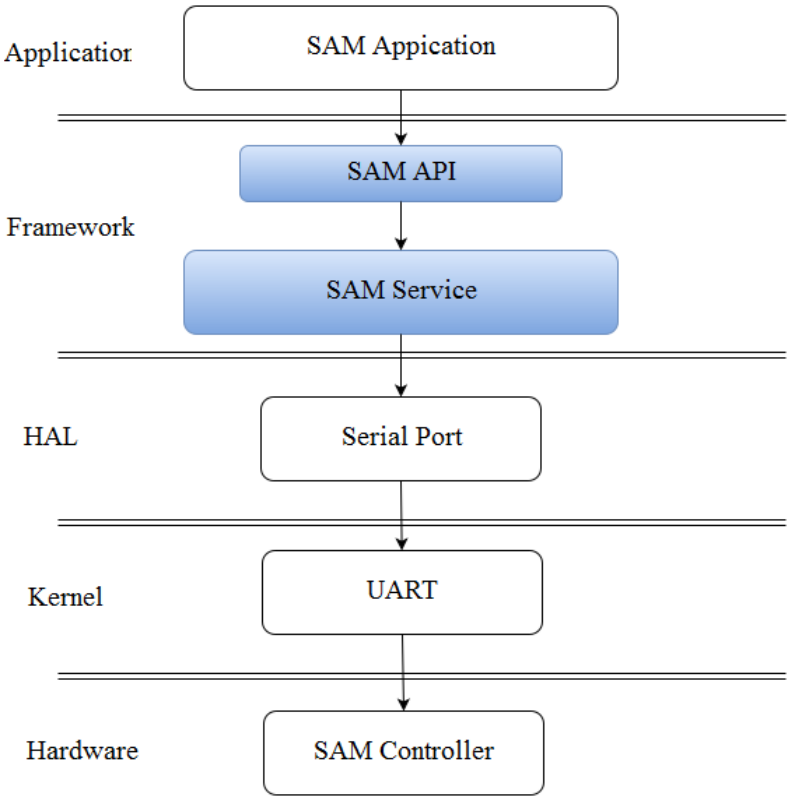
@Override
public void onDecodeComplete(String arg0) throws RemoteException {
     // TODO Auto-generated method stub
     //e1.setText(arg0);
     Toast.makeText(this, "Decode Data " + arg0, Toast.LENGTH_SHORT).show();
}

}
```

Chapter 2

SAM API

Because Google hasn't defined the class relating to SAM, this object provides the SAM Controller access method for the application.



Before developing your self-made application, the offered "***SamAPI.jar***" library file has to be imported into your project. Please refer to 1.1 Import Library on how to import the library.

Library required:
SamAPI.jar

IN THIS CHAPTER

2.1 Bind SAM Service 70
2.2 Service Information 71

2.1 BIND SAM SERVICE

SamManager InitInstance(Context context)	
Purpose	Binds SAM service.
Parameters	*context* is Activity

Example	`private SamManager m_SM;`
	`@Override`
	`protected void onCreate(Bundle savedInstanceState) {`
	`super.onCreate(savedInstanceState);`
	`setContentView(R.layout.activity_main);`
	
	`m_SM = SamManager.InitInstance(this);`
	`}`

Return Value	Gets a SamManager instance if successful, else null.

ClResult ExecuteApdu(int[] cmd, ApduOutputData outputData, int timeOut)	
Purpose	Sends APDU command and get response of IC card.
Parameters	*cmd* – [in] An integer arrary (APDU command). *outputData* – [out] An ApduOutputData object. time*out* – [in] An integer value (ms).

Example	`int[] cmd={0x00,0x01,0x02,0x03,0x04,0x05};`
	
	`ApduOutputData outputData=new ApduOutputData();`
	
	`if (ClResult.S_OK == m_SM.ExecuteApdu(cmd, outputData, 2000))`
	`{`
	`tvOutput.setText("Len is " + outputData.length + "\nData is " + intArrayToHex(outputData.outputData));`
	`}`
	`//Details of the ApduOutputData object`
	`//public class ApduOutputData {`
	`//  public int[] outputData;`
	`//  public int length;`
	`//}`

Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.

Release	
Purpose	Unbinds SAM service.
Syntax	**void Release ();**

Example	`@Override`
	`protected void onDestroy() {`
	`super.onDestroy();`
	`m_SM.Release();`
	`}`

See Also	SamManager InitInstance

ClResult Reset	
Purpose	Reset SAM controller.
Syntax	**ClResult Reset()**
Return Value	If successful, it returns ClResult.S_OK. Otherwise, it returns ClResult.S_ERR.

2.2 SERVICE INFORMATION

Get_SamServiceVer	
Purpose	Gets SAM service version of the device.
Syntax	**String Get_SamServiceVer ();**
Example	`String ver = m_SM.Get_SamServiceVer()`

Appendix I

RESPONSE CODE INSTRUCTIONS

Value	Instruction
ClResult.S_OK	Successful completion of request
ClResult.S_ERR	Unknown error
ClResult.ERR_NotSupport	Symbology not supported
ClResult.ERR_InvalidParameter	Invalid parameter

Appendix II

SCAN ENGINE SETTINGS

RS30 Series Mobile Computer supports the following reader types. Reader availability depends on the hardware integrated on the mobile computer.

Scan Engine		ID
1D	*CCD*	*SM1*
1D	*Laser*	*SE955*
2D	*Imager*	*SE4500*

IN THIS CHAPTER

Symbologies Supported..... 76

SYMBOLOGIES SUPPORTED

Depending on the scan engine integrated on the mobile computer, supported symbologies will differ as listed below.

		CCD	Laser	2D
Codabar		✓	✓	✓
Code 11		✘	✓	✓
Code 39	Code 39	✓	✓	✓
	Trioptic Code 39	✘	✓	✓
	Italian Pharmacode (Code 32)	✓	✓	✓
Code 93		✓	✓	✓
Code 128	Code 128	✓	✓	✓
	GS1-128 (EAN-128)	✓	✓	✓
	ISBT 128	✓	✓	✓
Code 2 of 5	Chinese 25	✘	✓	✓
	Industrial 25 (Discrete 25)	✓	✓	✓
	Interleaved 25	✓	✓	✓
	Convert Interleaved 25 to EAN-13	✘	✓	✓
	Matrix 25	✘	✘	✓
Composite Code	Composite CC-A/B	✘	✘	✓
	Composite CC-C	✘	✘	✓
	Compostie TLC 39	✘	✘	✓
GS1 DataBar (RSS)	GS1 DataBar-14 (RSS-14)	✓	✓	✓
	GS1 DataBar Limited (RSS Limited)	✓	✓	✓
	GS1 DataBar Expanded (RSS Expanded)	✓	✓	✓
	Convert to UPC/EAN	✘	✓	✓
Inverse	Inverse 1D barcodes	✘	✘	✓
Korean 3 of 5		✘	✘	✓
MSI		✓	✓	✓
Postal Codes	Australian Postal	✘	✘	✓
	Japan Postal	✘	✘	✓
	Netherlands KIX Code	✘	✘	✓

	US Postnet	✘	✘	✓
	US Planet	✘	✘	✓
	UK Postal	✘	✘	✓
EAN / UPC	EAN-8	✓	✓	✓
	EAN-8 Extend	✓	✓	✓
	EAN-13	✓	✓	✓
	Bookland EAN (ISBN)	✓	✓	✓
	ISSN EAN	✘	✘	✓
	UPC-A	✓	✓	✓
	UPC-E	✓	✓	✓
	Convert UPC-E to UPC-A	✓	✓	✓
	UPC-E1	✓	✓	✓
	Convert UPC-E1 to UPC-A	✓	✓	✓
2D Symbologies	Aztec	✘	✘	✓
	Data Matrix	✘	✘	✓
	Maxicode	✘	✘	✓
	MicroPDF417	✘	✘	✓
	MicroQR	✘	✘	✓
	PDF417	✘	✘	✓
	QR Code	✘	✘	✓

Appendix III

CODE TYPE & SYMBOLOGY

The table below lists symbologies and the corresponding code types.

Hex	ASCII	Symbology
0x2F	47 (/)	Composite CC-A
0x37	55 (7)	Composite CC-B
0x38	56 (8)	Korean 3 of 5
0x39	57 (9)	ISSN
0x3F	63 (?)	ISBT 128 Concatenation
0x40	64 (@)	ISBT 128
0x41	65 (A)	Code 39
0x42	66 (B)	Italian Pharmacode (Code 32)
0x43	67 (C)	French Pharmacode (CIP 39)
0x44	68 (D)	Industrial 25
0x45	69 (E)	Interleaved 25
0x46	70 (F)	Matrix 25
0x47	71 (G)	Codabar (NW7)
0x48	72 (H)	Code 93
0x49	73 (I)	Code 128
0x4A	74 (J)	UPC-E0
0x4B	75 (K)	UPC-E0 with Addon 2
0x4C	76 (L)	UPC-E0 with Addon 5
0x4D	77 (M)	EAN-8
0x4E	78 (N)	EAN-8 with Addon 2
0x4F	79 (O)	EAN-8 with Addon 5
0x50	80 (P)	EAN-13 (also UPC-A on CCD/Laser scan engine)
0x51	81 (Q)	EAN-13 with Addon 2
0x52	82 (R)	EAN-13 with Addon 5
0x53	83 (S)	MSI
0x54	84 (T)	Plessey
0x55	85 (U)	GS1-128 (EAN-128)
0x56	86 (V)	Undefined
0x57	87 (W)	Undefined
0x58	88 (X)	Undefined

0x59	89 (Y)	Undefined
0x5A	90 (Z)	Telepen
0x5B	91 ([ )	GS1 DataBar Omnidirectional (RSS-14)
0x5C	92 (\)	GS1 DataBar Limited (RSS Limited)
0x5D	93 ( ])	GS1 DataBar Expanded (RSS Expanded)
0x5E	94 (^)	UPC-A
0x5F	95 (_)	UPC-A with Addon 2
0x60	96 (‘)	UPC-A with Addon 5
0x61	97 (a)	UPC-E1
0x62	98 (b)	UPC-E1 with Addon 2
0x63	99 (c)	UPC-E1 with Addon 5
0x64	100 (d)	TLC 39 (TCIF Linked Code 39)
0x65	101 (e)	Trioptic (Code 39)
0x66	102 (f)	Bookland (EAN)
0x67	103 (g)	Code 11
0x68	104 (h)	Code 39 Full ASCII
0x69	105 (i)	IATA[Note] (Code 25 used on flight tickets)
0x6A	106 (j)	Industrial 25 (Discrete 25)
0x6B	107 (k)	PDF417
0x6C	108 (l)	MicroPDF417
0x6D	109 (m)	Data Matrix
0x6E	110 (n)	Maxicode
0x6F	111 (o)	QR Code
0x70	112 (p)	US Postnet
0x71	113 (q)	US Planet
0x72	114 (r)	UK Postal
0x73	115 (s)	Japan Postal
0x74	116 (t)	Australian Postal
0x75	117 (u)	Dutch Postal
0x76	118 (v)	Composite CC-C
0x77	119 (w)	Macro PDF
0x78	120 (x)	Coupon Code
0x79	121 (y)	Chinese 25
0x7A	122 (z)	Aztec
0x7B	123 ({)	MicroQR
0x7C	124 (|)	USPS 4CB / One Code / Intelligent Mail

0x7D	125 (})	UPU FICS Postal
0x7E	126 (~)	Macro MicroPDF417

Appendix IV

SAMPLE CODE

```
package com.example.cipherlab;

import android.app.Activity;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends Activity {

	private TextView tv1 = null;
	private IntentFilter filter;

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);

		tv1 = (TextView)findViewById(R.id.tv1);

		// Register an intent filter to get the intent we want.
		filter = new IntentFilter();
		filter.addAction("com.cipherlab.barcodebaseapi.PASS_DATA_2_APP");
		registerReceiver(myDataReceiver, filter);
	}

	@Override
	protected void onDestroy() {
		super.onDestroy();
		unregisterReceiver(myDataReceiver);
	}

	// Create a broadcast object to get the intent sent from the service.
	private final BroadcastReceiver myDataReceiver = new BroadcastReceiver() {
		@Override
		public void onReceive(Context context, Intent intent) {
		// If the intent of the Intent_SOFTTRIGGER_DATA string is received,
		// the following statements are excuted.
			if
(intent.getAction().equals("com.cipherlab.barcodebaseapi.PASS_DATA_2_APP")) {

				tv1.setText("");

				// Fetch the data along with the intent.
				String data = intent.getStringExtra("Decoder_Data");
				// Fetch the original data along with the intent (not change
				// to UTF-8 Format)
				byte [] buffer =
				intent.getByteArrayExtra("Decoder_DataArray");
				// Fetch the code type along with the intent
				int iCodeType = intent.getIntExtra("Decoder_CodeType", 0);

				// Display the data.
				tv1.setText(data);
```

```
                    }

                }
        };
}
```

Note: The enableKeyboardEmulation is supposed to be set to “KeyboardEmulationType.InputMethod” or “KeyboardEmulationType.KeyEvent” so this sample can then receive the broadcasting intent message. For further information, please refer to Set_ReaderOutputConfiguration.

Appendix V

OPEN SOURCE LICENSE

The Reader/SAM API involves the android-serialport-api project of Apache License Version 2.0, January 2004.

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

a. You must give any other recipients of the Work or Derivative Works a copy of this License; and

b. You must cause any modified files to carry prominent notices stating that You changed the files; and

c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS