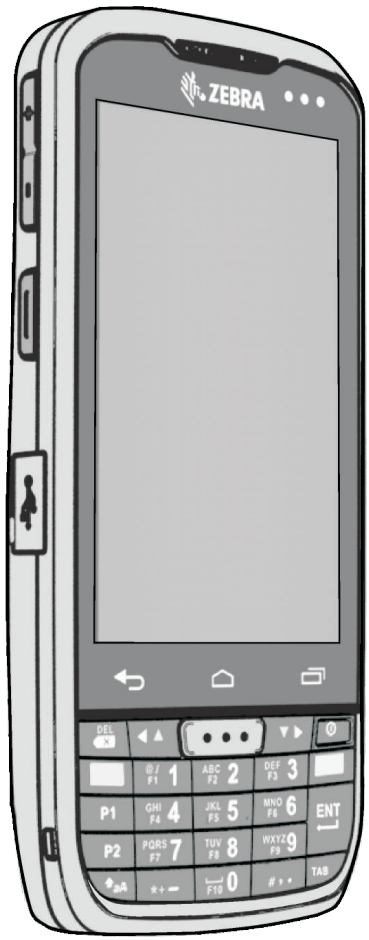
MC36



Mobile Computer

Integrator Guide

ZEBRA

MN002340A01

MC36
INTEGRATOR GUIDE

MN002340A01
Rev.A
October 2015

Copyrights

Disclaimer

Please note that certain features, facilities, and capabilities described in this document may not be applicable to or licensed for use on a particular system, or may be dependent upon the characteristics of a particular mobile subscriber unit or configuration of certain parameters. Please refer to your contact for further information.

Trademarks

Zebra and the Zebra head graphic are registered trademarks of ZIH Corp.

Revision History

Changes to the original guide are listed below:

Revision	Date	Description
A01 Rev. A	2015/10/29	Initial release.

Table of Contents

Chapter 2: Android Programming

Chapter 3: ADB USB Setup

Chapter 4: MTK Debug Logging

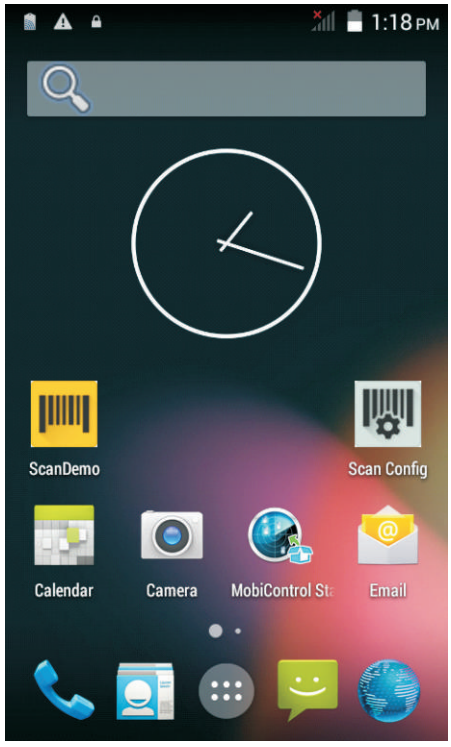
Chapter 5: Key Remap

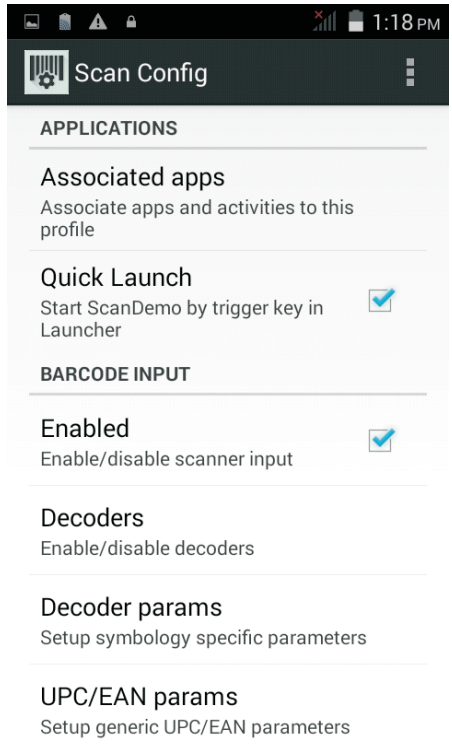
Chapter 1 Scan Config

Introduction

Scan Config is a utility to allow the user to scan bar codes in any application without adding any customization. It runs in the background and communicates with built-in barcode scanner. The function pictures give you basic idea about Scan Config's UI. More functions will be shown on UI along with the development of Scan Config tool. This document is to introduce what are the functionalities that Scan Config has now or will have.

Figure1-1:



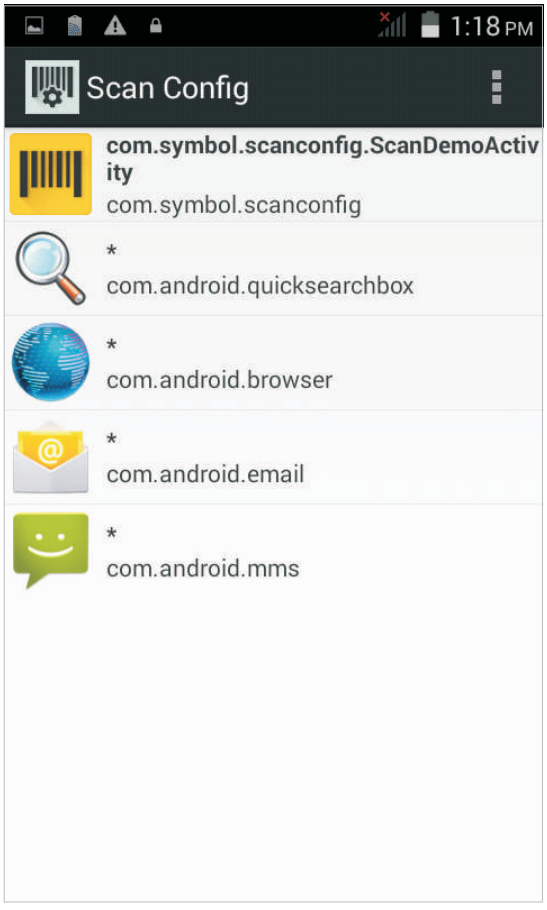


Application Association

Associated apps

Specify application that is able to receive the data in foreground from scanner. User can select one or more apps from all existed applications/activities on the device. All of the selected apps are represented in a list.
While the foreground application/activity is matched in the list, the captured barcode data is sent to the application via plug-in as typed on the keyboard.
Initially the list contains some stock Android apps and ScanDemo. (refer to the demo picture above)

Figure 1-2:

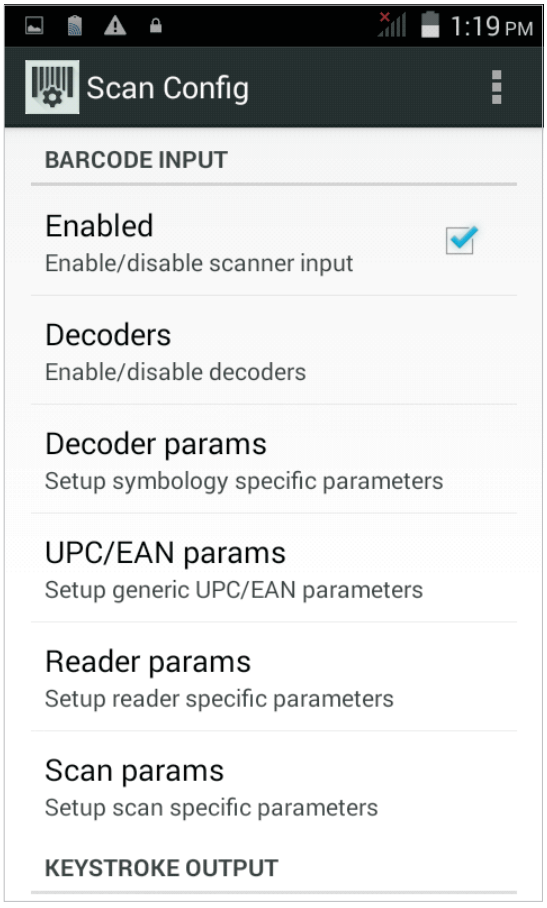


Quick Launch

Enabled for triggering ScanDemo in Launcher application with scan keys.
For example, we can trigger ScanDemo on home screen or programs by this.
It is enabled as default.

Barcode Input Option

Figure 1-3:



Scan Config provides below options to configure the barcode scanner input.

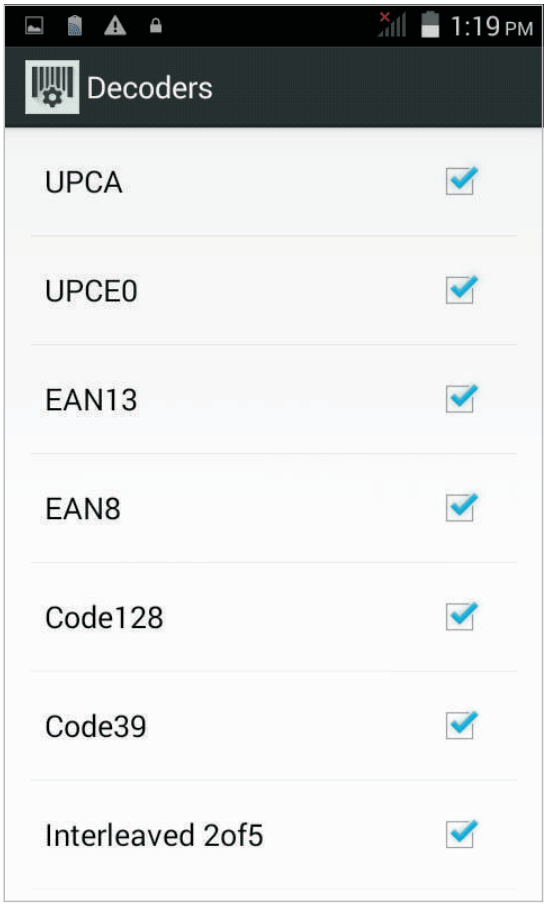
Enabled

Checked for enabling scanner decode input.
Scanner can not decode unless user-end decodes directly by API.
It is enabled as default.

Decoders

Configures which barcode decoders are enabled or disabled. A list showing all supported barcode decoders appears when the option is touched. A check in the checkbox indicates that the decoder is enabled.

Figure 1-4:



Below are the supported decoders for 1D/2D scanners:

UPC-A(*)	Code 93	MSI
UPC-E0(*)	Code 11	Trioptic 39
UPC-E1	Interleaved 2 of 5(*)	GS1 DataBar(*)
EAN-8(*)	Discrete 2 of 5	GS1 DataBar Limited(*)
EAN-13(*)	Chinese 2 of 5	GS1 DataBar Expanded(*)
Code 128(*)	Matrix 2 of 5	
Code 39(*)	Codabar	

Below are the supported decoders for 2D scanner only:

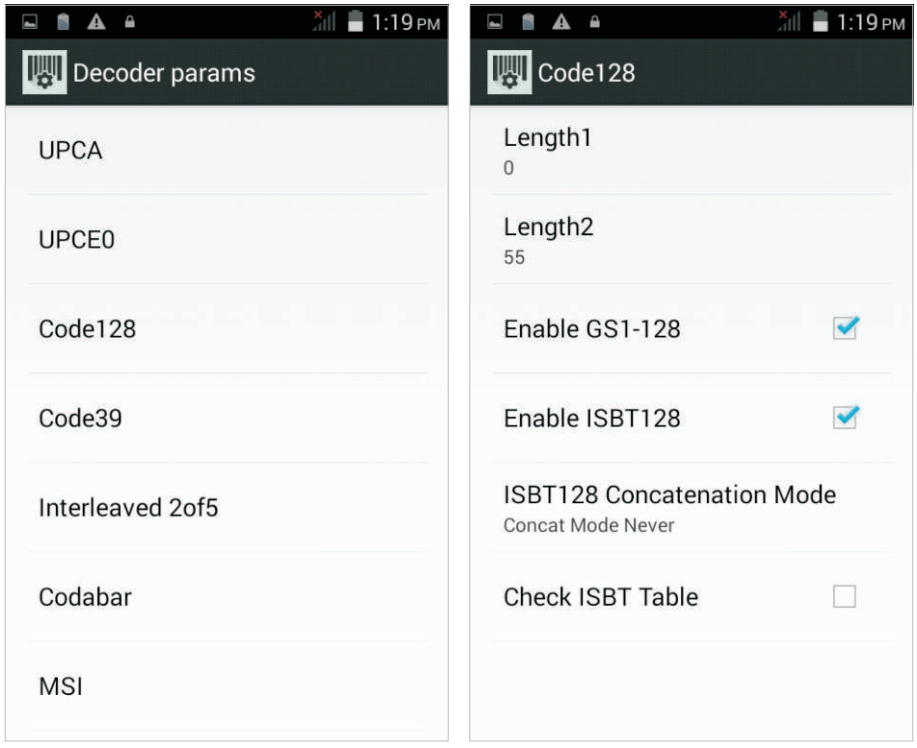
Composite CC-C	QR Code(*)	US Postnet
Composite CC-A/B	MicroQR(*)	US Planet
PDF417(*)	Aztec(*)	UK Postal
MicroPDF	Han Xin(*)	Japan Postal
Data Matrix(*)	US4state	Australia Post
Maxicode(*)	US4state FICS	TLC-39

Note: * means the decoder is enabled in default.

Decoder Params

Decode Params provides options to configure individual decoder parameters.

Figure 1-5:



Note: The default values are shown at the end of the description of each parameter.

UPCA

- Report Check Digit - Enables to show the check digit. (*enabled)
- Preamble - Preamble characters are part of the UPC symbol consisting of Country Code and System Character. (*Preamble Sys Char)

UPCE0

- Report Check Digit - Enables to show the check digit. (*disabled)
- Preamble - Preamble characters are part of the UPC symbol consisting of Country Code and System Character. (*Preamble Sys Char)
- Convert UPCE0 To UPCA - Enable to convert UPCE0 (zero suppressed) decoded data to UPC-A format. (*disabled)

UPCE1

- Report Check Digit - Enables to show the check digit. (*enabled)
- Preamble - Preamble characters are part of the UPC symbol consisting of Country Code and System Character. (*Preamble None)
- Convert UPCE1 To UPCA - Enable to convert UPCE1 decoded data to UPC-A format. (*disabled)

Code128

- Length1 - To decode a Code 128 symbol with a specific length range set this value to the lower limit. (*0)
- Length2 - To decode a Code 128 symbol with a specific length range set this value to the upper limit. (*55)
- Enable GS1–128 - Set the GS1–128 subtype. (*enabled)
- Enable ISBT128 - Set the ISBT128 subtype. (*enabled)
- ISBT128 Concatenation Mode - Select an option for concatenating pairs of ISBT code types. (*Concat Mode Never)
- Check ISBT Table - Enable Check ISBT Table to concatenate only those pairs found in this table. (*disabled)

Code 39

- Length1 - To decode a Code 39 symbol with a specific length range set this value to the lower limit. (*0)
- Length2 - To decode a Code 39 symbol with a specific length range set this value to the upper limit. (*55)
- Verify Check Digit - Enable this feature to check the integrity of all Code 39 symbols. (*disabled)
- Report Check Digit - Transmit Code 39 data with or without the check digit. (*disabled)
- Full ASCII - Code 39 Full ASCII is a variant of Code 39 that pairs characters to encode the full ASCII character set. (*disabled)
- Convert Code39 To Code32 - Code 32 is a variant of Code 39 used by the Italian pharmaceutical industry. (*disabled)
- Report Code32 Prefix - Enable or disable adding the prefix character "A" to all Code 32 bar codes. (*disabled)

Code 93

- Length1 - To decode a Code 93 symbol with a specific length range set this value to the lower limit. (*0)
- Length2 - To decode a Code 93 symbol with a specific length range set this value to the upper limit. (*55)

Code 11

- Length1 - To decode a Code 11 symbol with a specific length range set this value to the lower limit. (*4)
- Length2 - To decode a Code 11 symbol with a specific length range set this value to the upper limit. (*55)
- Verify Check Digit - Selects the check digit mechanism for the decoded Code 11 bar code. (*No Check Digit)
- Report Check Digit - Transmit Code 11 data with or without the check digit. (*disabled)

Matrix 2 of 5

- Length1 - To decode a Matrix 2 of 5 symbol with a specific length range set this value to the lower limit. (*55)
- Length2 - To decode a Matrix 2 of 5 symbol with a specific length range set this value to the upper limit. (*0)
- Redundancy - Sets the reader to read the bar code twice before accepting data. (*disabled)
- Verify Check Digit - Enable this feature to check the integrity of all Matrix 2 of 5 symbols. (*disabled)
- Report Check Digit - Transmit M2of5 data with or without the check digit. (*disabled)

Interleaved 2 of 5

- Length1 - To decode an Interleaved 2 of 5 symbol with a specific length range set this value to the lower limit. (*14)
- Length2 - To decode an Interleaved 2 of 5 symbol with a specific length range set this value to the upper limit. (*10)
- Check Digit – Select the Check Digit type. (*No Check Digit)
- Report Check Digit - Transmit Interleaved 2 of 5 data with or without the check digit. (*disabled)
- Convert ITF-14 To EAN13 - Convert 14-character Interleaved 2 of 5 bar codes to EAN-13. (*disabled)

Discrete 2 of 5

- Length1 - To decode a Discrete 2 of 5 symbol with a specific length range set this value to the lower limit. (*12)
- Length2 - To decode a Discrete 2 of 5 symbol with a specific length range set this value to the upper limit. (*55)

Codebar

- Length1 - To decode a Codabar symbol with a specific length range set this value to the lower limit. (*6)
- Length2 - To decode a Codabar symbol with a specific length range set this value to the upper limit. (*55)
- CLSI Editing - Enable this parameter to strip the start and stop characters and insert a space after the first, fifth,

and tenth characters of a 14-character Codabar symbol. (*disabled)

- NOTIS Editing - Enable this parameter to strip the start and stop characters from a decoded Codabar symbol. (*disabled)

MSI

- Length 1 - To decode a MSI symbol with a specific length range set this value to the lower limit. (*4)
- Length 2 - To decode a MSI symbol with a specific length range set this value to the upper limit. (*55)
- Check Digit – Select one or two check digit. (*One Check Digit)
- Check Digit Scheme - Select the algorithm used to encode the check digit. (*Mod-10-10)
- Report Check Digit - Transmit MSI data with or without the check digit. (*disabled)

Data Matrix

- DataMatrix Inverse. - Set the Data Matrix inverse decoder setting. It makes the decoder decode regular or inverse bar codes. (*Disable)

Aztec

- Aztec Inverse - Set the Aztec inverse decoder setting. It makes the decoder decode regular or inverse bar codes. (*Disable)

QR Code

- QRCode Inverse - Set the QR code inverse decoder setting. It makes the decoder decode regular or inverse bar codes. (*Disable)

Composite CC-A/B

- UCC Link Mode – Select UCC Link Mode. (*Link Flag Ignored)

US Planet

- Report Check Digit - Transmit US Planet data with or without the check digit. (*disabled)

UK Postal

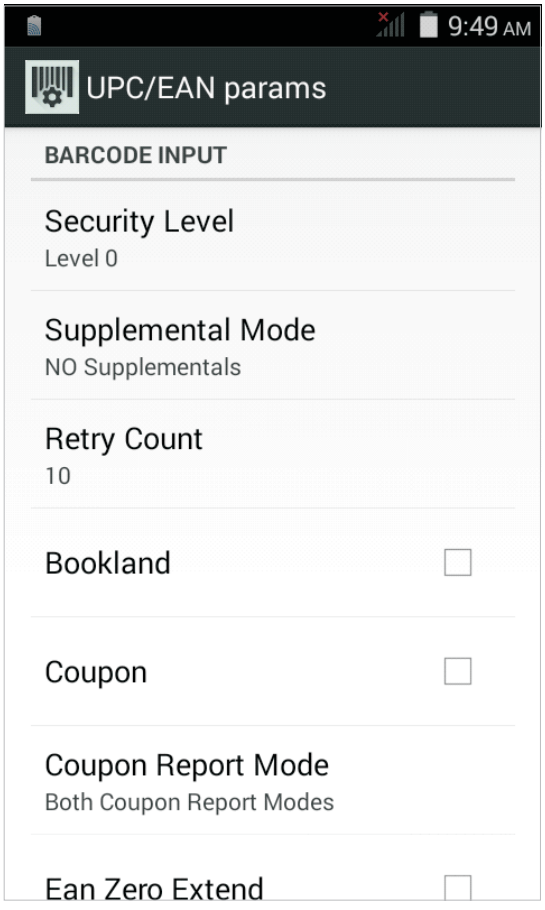
- Report Check Digit - Transmit UK Postal data with or without the check digit. (*disabled)

HAN XIN

- HAN XIN Inverse. - Set the HAN XIN inverse decoder setting. It makes the decoder decode regular or inverse bar codes. (*Disable)

UPC/EAN Params

Figure 1-6:

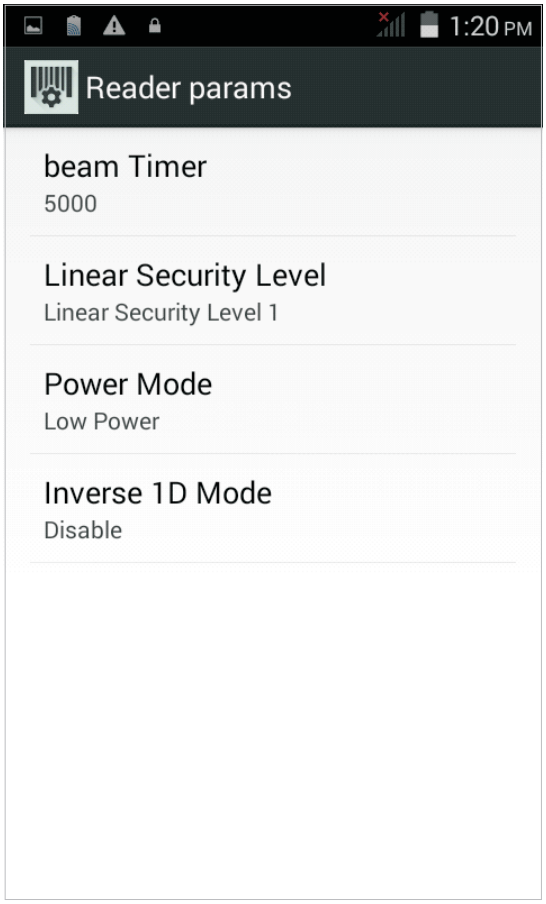


Note: The default values are shown at the end of the description of each parameter.

- Security Level – Select higher security levels for lower quality bar codes. (*Level 0)
- Supplemental Mode – select Supplemental Mode. (*No Supplementals)
- Retry Count – Retry count for auto-discriminating for supplemental. (*10)
- Bookland – Enable or disable this Bookland option. (*disabled)
- Coupon – Enables Coupon code decoding. (*disabled)
- Coupon Report Mode – Select Coupon Report Mode. (*Both Coupon Report Modes)
- EAN Zero Extend – Enable Ean Zero Extend. (*disabled)
- Bookland Format – If Bookland option is enabled, select one of the formats for Bookland data. (*Format ISBN-10)
- Convert DataBar to UPC EAN – converts DataBar barcodes to UPC/EAN format. (*disabled)

Reader Params

Figure 1-7:



Note: The default values are shown at the end of the description of each parameter.

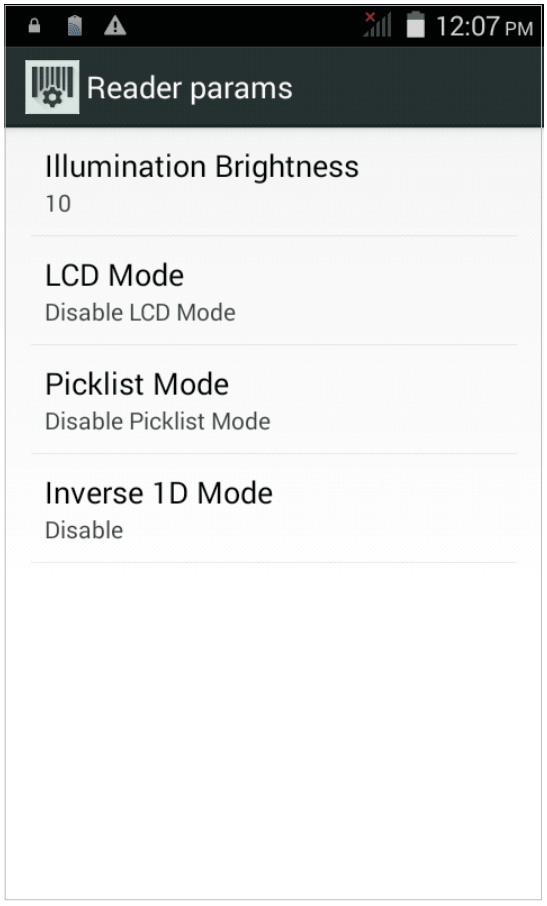
Laser Specific

- Beam Timer – Sets the maximum amount of time that the reader remains on.(*5000)
- Linear Security Level – Sets the number of times a bar code is read to confirm an accurate decode. (*Linear Security Level 1)
- Inverse 1D Mode – Allows the user to select decoding on inverse 1D barcodes. (*Disable)
- Power Mode –Set scanner power mode. (*Low Power)

Imager Specific

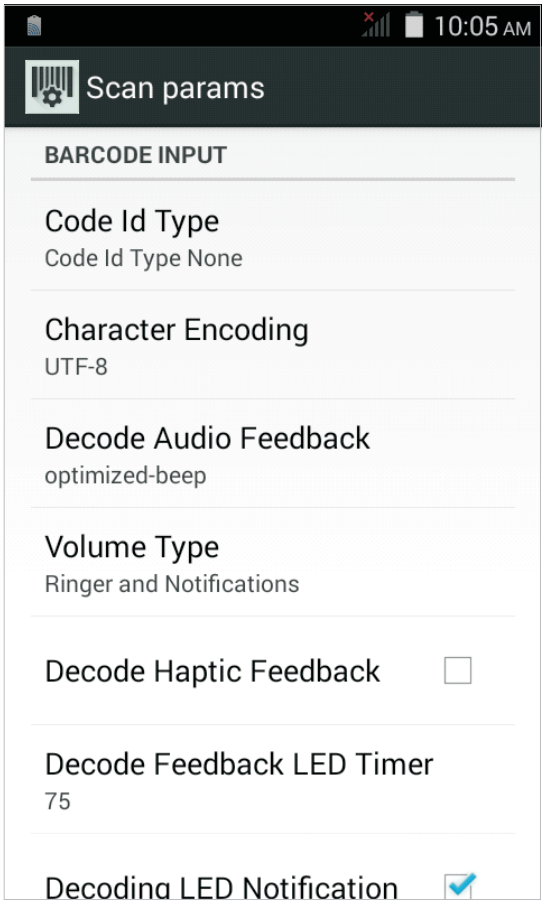
- Illumination Brightness –Set illumination Brightness of the Imager. (*10)
- Inverse 1D Mode – Allows the user to select decoding on inverse 1D barcodes. (*Disable)
- LCD Mode – Enables or disables LCD mode to enhance the ability of the imager to read bar codes from LCD displays. (*Disable LCD Mode)
- Pick List – Allows the imager to decode only the bar code that is directly under the cross-hair/reticle (+)
- part of the pattern. (*Disable Picklist Mode)

Figure 1-8:



Scan Params

Figure 1-9:

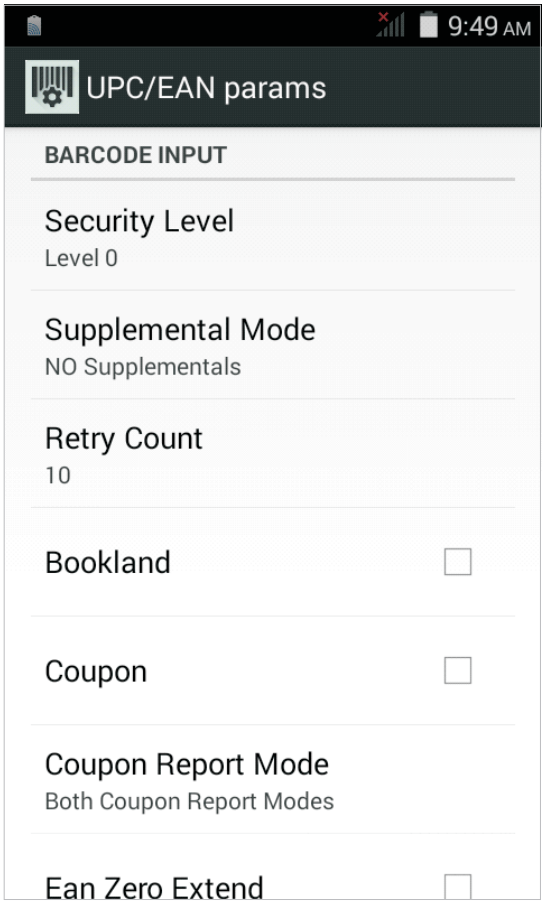


Note: The default values are shown at the end of the description of each parameter.

- Code Id Type – Insert the code ID identifying code type of a scanned barcode. (*Code Id Type None)
- Character Encoding – Set the encoding as UTF-8, Simplified Chinese or Traditional Chinese for the decoded data. (*UTF-8)
- Decode Audio Feedback – Select an audio tone to sound upon a good decode. (*optimized-beep)
- Volume Type – The audio stream type refers to type of streaming on which the scan beep should be played. (*Ringer and Notifications)
- Decode Haptic Feedback – Enable vibration upon a good decode. (*diabled)
- Decode Feedback LED Timer – Sets the duration of green-LED feedback upon a good decode. (*75)
- Decoding LED Notification – Enable red-LED notification upon a decode action. (*enabled)

UPC/EAN Params

Figure 1-10:



Note: The default values are shown at the end of the description of each parameter.

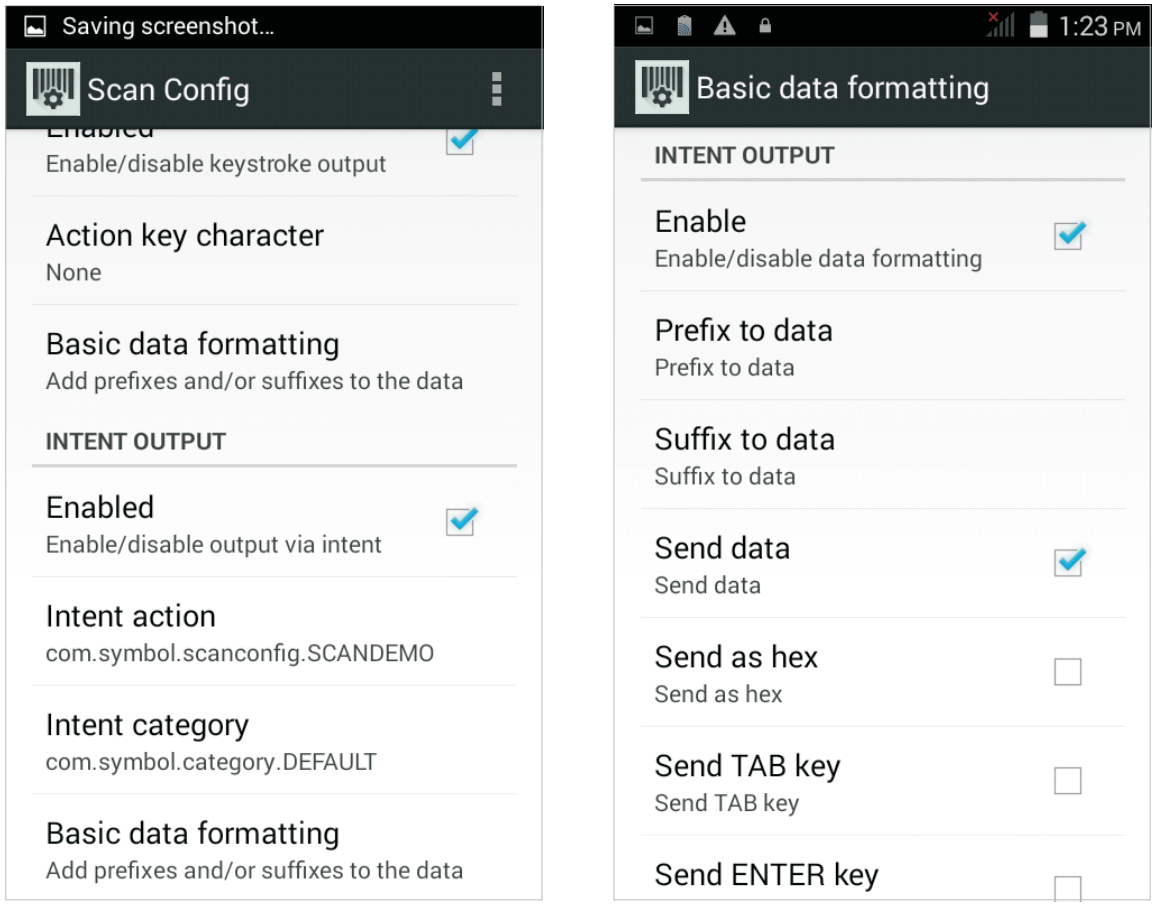
- Security Level – Select higher security levels for lower quality bar codes. (*Level 0)
- Supplemental Mode – select Supplemental Mode. (*No Supplementals)
- Retry Count – Retry count for auto-discriminating for supplemental. (*10)
- Bookland – Enable or disable this Bookland option. (*disabled)
- Coupon – Enables Coupon code decoding. (*disabled)
- Coupon Report Mode – Select Coupon Report Mode. (*Both Coupon Report Modes)
- EAN Zero Extend – Enable Ean Zero Extend. (*disabled)
- Bookland Format – If Bookland option is enabled, select one of the formats for Bookland data. (*Format ISBN-10)
- Convert DataBar to UPC EAN – converts DataBar barcodes to UPC/EAN format. (*disabled)

Plug-In Functions

Basic Data Format

The Basic Data Format plug-in allows Scan Config to add a prefix and/or a suffix to the captured data before passing it to an Output Plug-in.

Figure1-11:



Intent Output

Note: The default values are shown at the end of the description of each parameter.

The Intent Output plug-in allows the captured data to be sent to an application in the form of an implicit Intent. It provides below options in Scan Config UI:

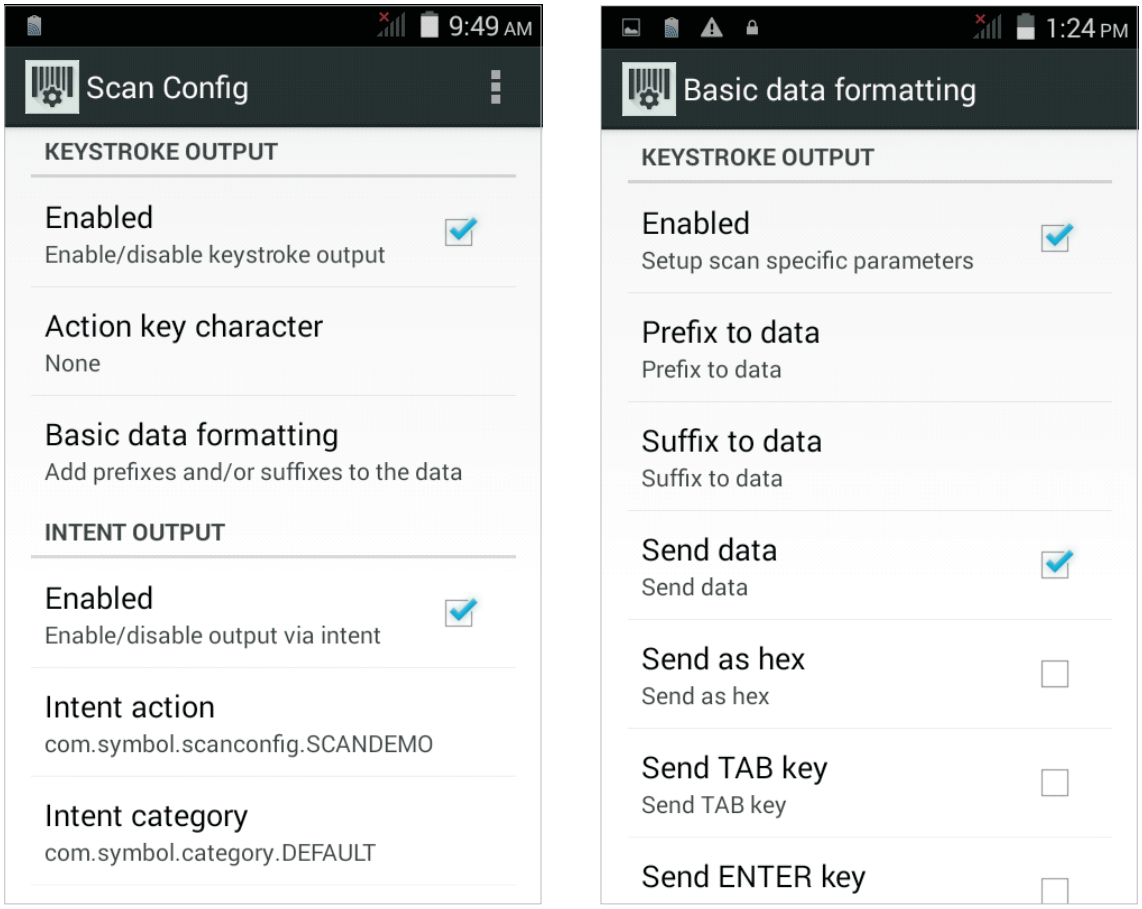
- Enabled - Enables or disables this plug-in. (*enabled)
- Intent action - Enter the Intent Action name (*com.symbol.scanconfig.SCANDEMO)
- Intent category?- Enter the Intent Category name. (*com.symbol.category)
- Basic data formatting - allows to configure data formatting for the Intent output
- Enabled - Enables or disables Basic Data Formatting. (*enabled)

- Prefix to data - Add characters to the beginning of the data when sent. Suffix to data - Add characters to the end of the data when sent.Send data - Set to transfer the captured data to the foreground application. Disabling this option prevents the actual data from being transmitted. However, the prefix and suffix strings, if present,are still transmitted
- even when this option is disabled. (*enabled)
- Send as hex - Set to send the data in hexadecimal format. (*disabled)
- Send TAB key - Set to append a tab character to the end of the processed data. (*disabled)
- Send ENTER key - Set to append an Enter character to the end of the processed data. (*disabled)

Keystroke Output

Note: The default values are shown at the end of the description of each parameter.

Figure1-12:



The Keystroke Plug-in captures and sends data received from the scanner to the foreground applications by emulating keystrokes.

- **Enabled -** Enables or disables this plug-in. (*enabled)
- **Action key character -** Enables or disables decoding of a special character embedded within a bar code or MSR data for use in native Android applications. (*None)
- **Basic data formatting -** allows to configure data formatting for the Intent output

  Enabled - Enables or disables Basic Data Format (*enabled)

  Prefix to data - Add characters to the beginning of the data when sent.

  Suffix to data - Add characters to the end of the data when sent.

  Send data - Set to transfer the captured data to the foreground application. Disabling this option prevents the actual data from being transmitted. However, the prefix and suffix strings, if present, are still transmitted even when this option is disabled. (*enabled)

  Send as hex - Set to send the data in hexadecimal format. (*disabled)

  Send TAB key - Set to append a tab character to the end of the processed data. (*disabled)

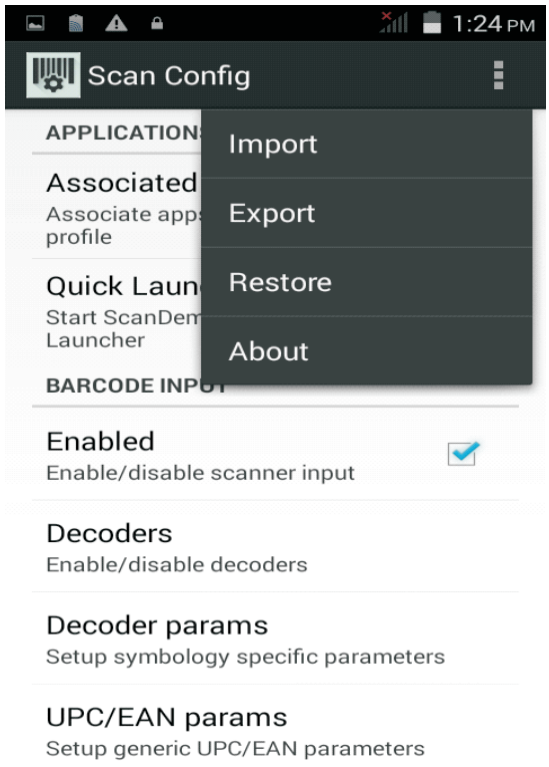
  Send ENTER key - Set to append an Enter character to the end of the processed data. (*disabled)

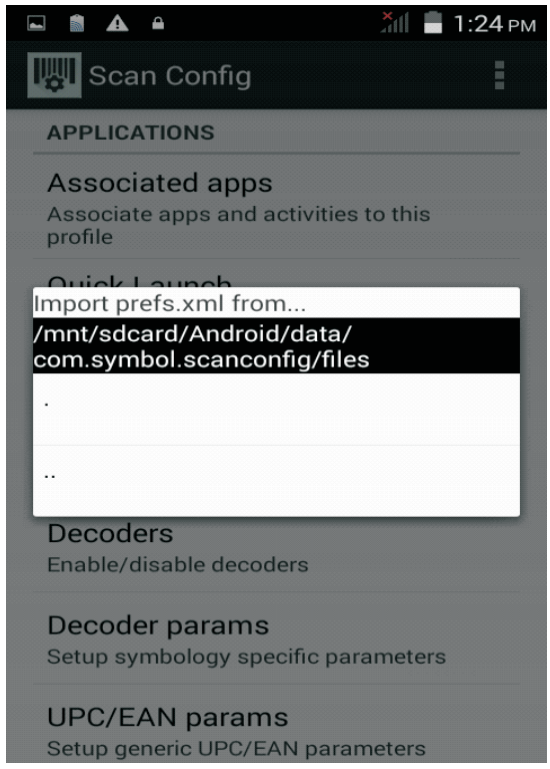
Configuration Management

Import

Allows the user to import Scan Config configuration file from specified path. The imported configuration overrides the current configuration.

Figure1-13:





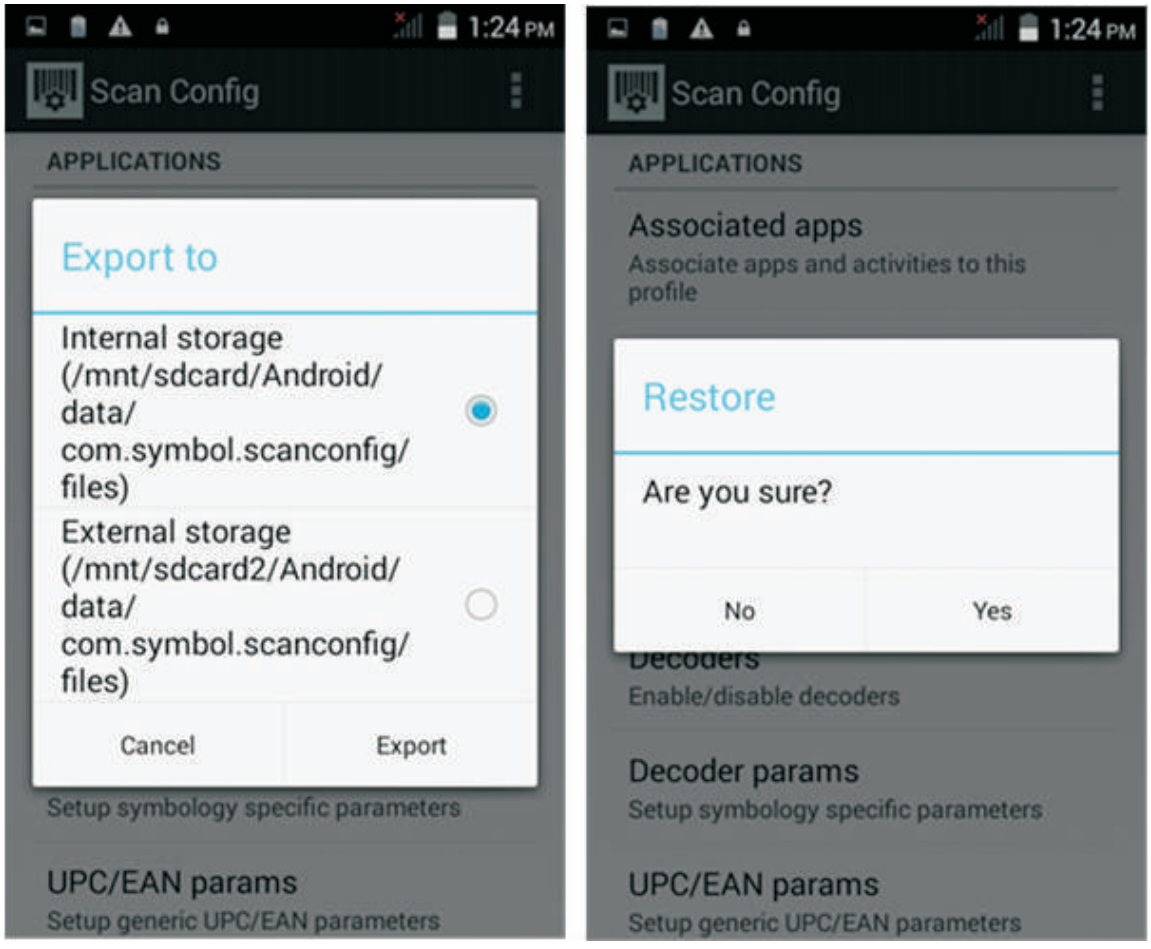
1. Click ⋮
2. Select "Import"
3. Choose the path for the file

The imported configuration overrides the current configuration.

Export

Allows the user to export the current Scan Config configuration to specified path.

Figure1-14:



1. Click ⋮
2. Select "Import"
3. Choose to save the file in internal storage or external storage (if it exists)

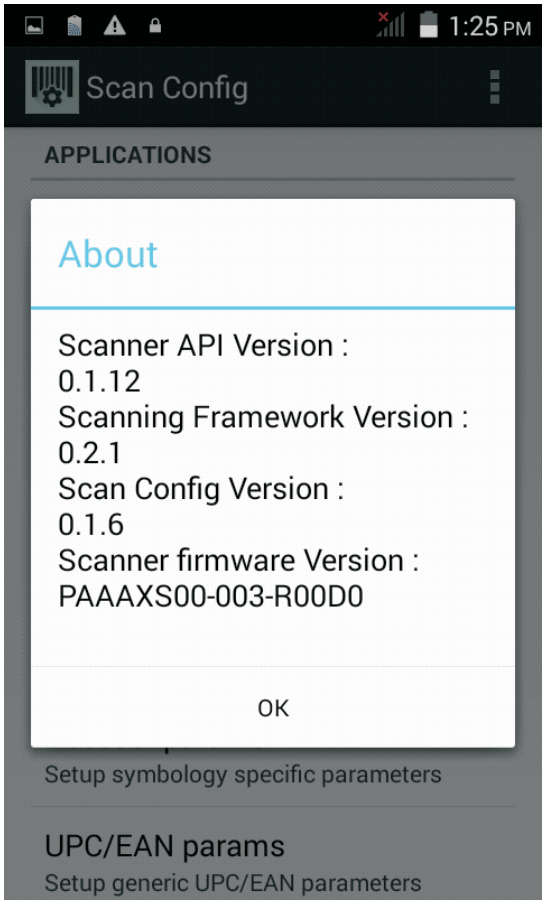
Restore

Restore the configuration to factory defaults.

About

Shows the versions of scanner API, scanning framework, ScanConfig and scanner firmware.

Figure 1-15:



Configuration File Management

Enterprise Folder

Internal storage contains an Enterprise folder(/enterprise). Enterprise folder is persistent and maintains data after an Enterprise reset.
After an Enterprise reset, ScanConfig checks folder /enterprise/devices/settings/scanconfig/enterprise for a configuration file, prefs.xml. If the file exists, ScanConfig imports the file to replace the configuration.

Note:

1.This is an implicit operation.

2.The permissions of prefs.xml should be set to 777.

3.Factory reset clears all files in Enterprise folder.

Auto Import

ScanConfig monitors the Enterprise folder /enterprise/device/settings/scanconfig/autoimport for prefs.xml file. Once the prefs.xml is found, ScanConfig imports the file and replaces the existing configuration. And then, ScanConfig deletes the prefs.xml after finishing import.
The configuration will be used the next time an associated application is opened.

Note:

1.This is an implicit operation.

2.The permissions of prefs.xml should be set to 777.

3.Factory reset clears all files in Enterprise folder.

Chapter 2
Android Programming

Introduction

This chapter provides an introduction to the MC36 Android SDK Add-on. It provides all the information you need to install and use the SDK add-on, as well as an introduction to the unique APIs available in MC36 devices. The MC36 Android Add-on can be download from https://www.zebra.com/support.

MC36 Android SDK Add-on

The SDK Add-on package includes:

- Libraries for Symbol APIs and Mediatek APIs.
- Sample file for Scan API

The naming of the package is:

- symbol_sdk_api_addon-< Android API Level>-< MTK API Level >-< Symbol API version >.zip

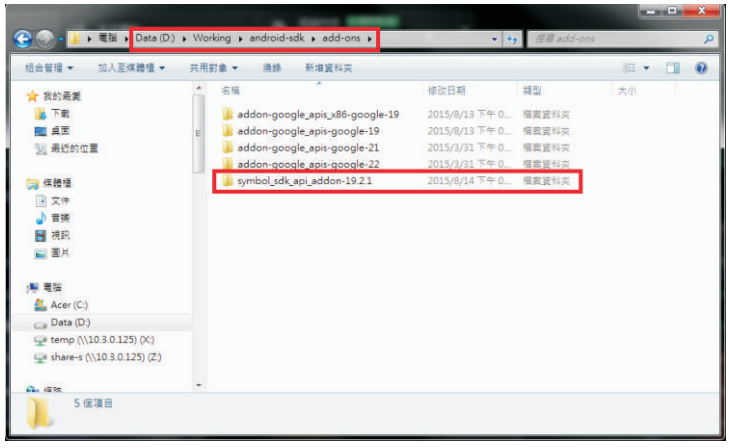
Prerequisites

Before installing the SDK add-on you should have an installation of the Android SDK, including the required supporting software. For more information see Get the Android SDK on the Android Developer website.

SDK Add-on Installation

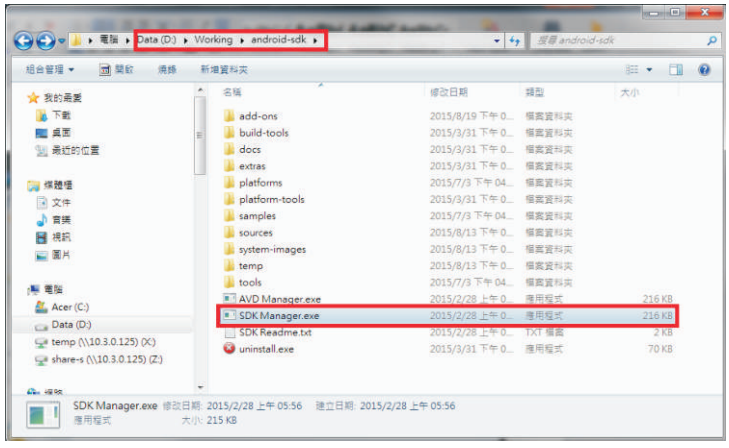
1.Make sure the Android SDK and Eclipse are installed.
2.Extract MC36 SDK Add-on file to “add-ons” folder of Android SDK Installation folder.Example :
MC36 SDK Add-on file : symbol_sdk_api_addon-19.2.1.zip
Android SDK Folder : D:\Working\android-sdk
Unpack symbol_sdk_api_addon-19.2.1.zip to D:\Working\android-sdk\symbol_sdk_api_addon-19.2.1

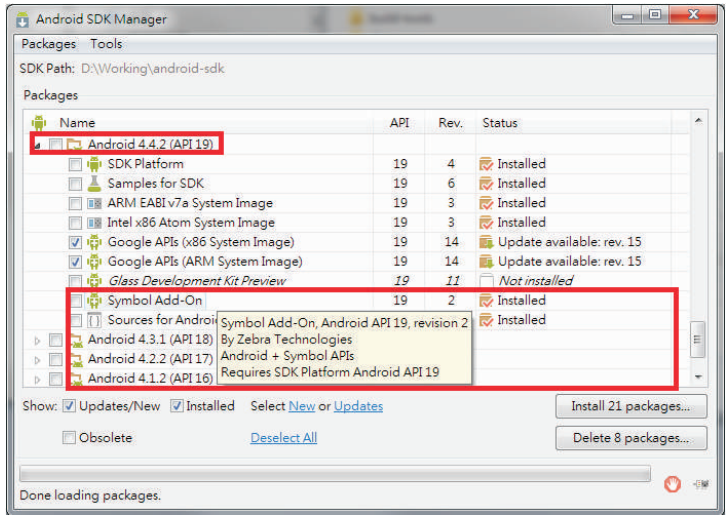
Figure 2-1:



3.Open Android SDK Manager to check if MC36 SDK add-on is installed.

Figure2-2:

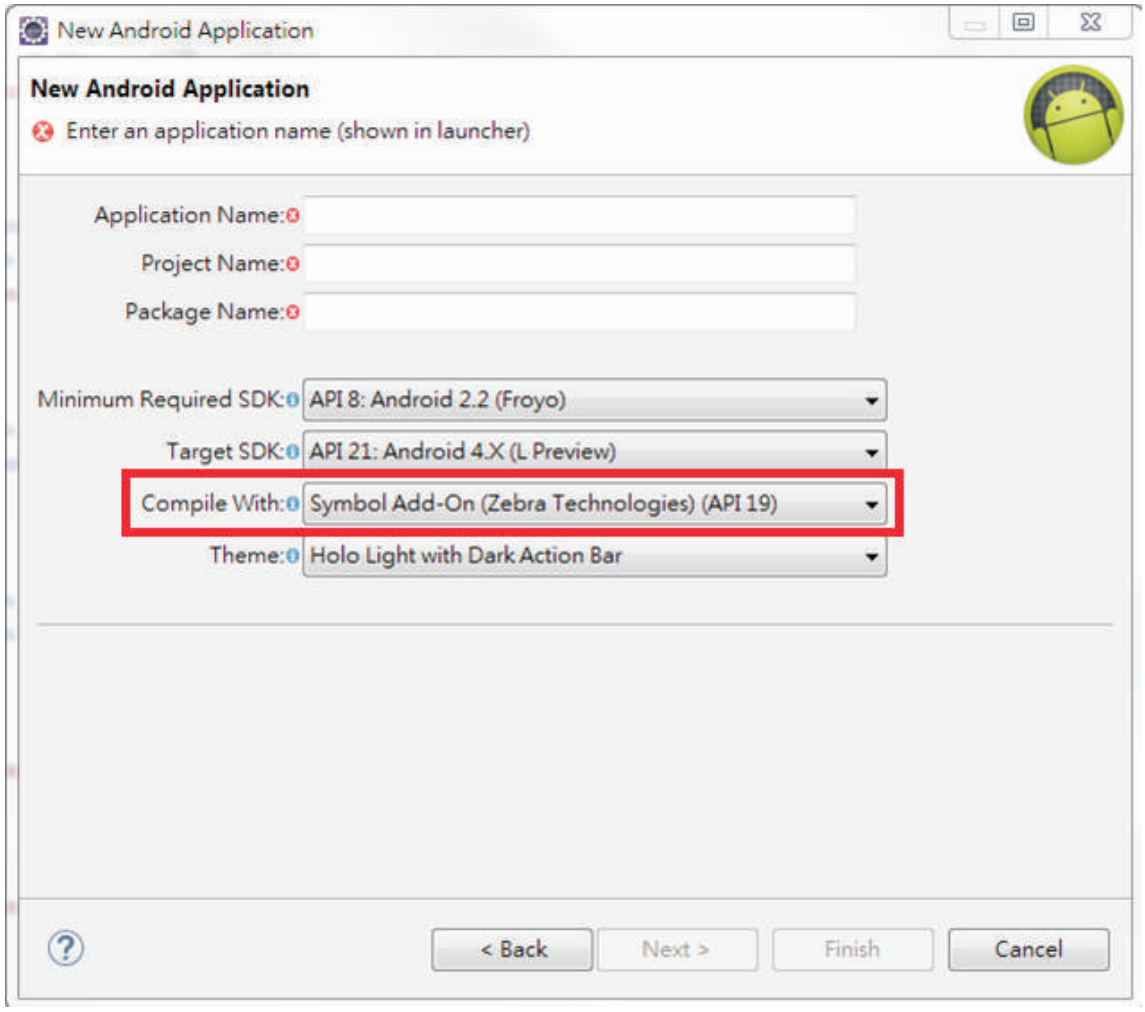




How to use in Eclipse

1. Open Eclipse IDE.
2. Create a new Android project; select the "Compile with" Symbol Add-on as the target.

Figure 2-3:



3. The shared libraries will be automatically added and ready to use.

MC36 APIs

Scanning API

The scanning SW would offer below java class for Android application developer:

com.symbol.scanning.Scanner

Scanner class will provide the access to the built-in scanner.
Example Usage:

```
Scanner mScanner = new Scanner();
try{
mScanner.enable();
boolean scannerEnable = mScanner.isEnable();
DataListener mDataListener = new DataListener(){ <omitted> };
if(scannerEnable)
{
    mScanner.addDataListener(mDataListener);
mScanner.read();
mScanner.cancelRead();
mScanner.removeDataListener(mDataListener);
mScanner.disable();
}
}
Catch(ScannerException se){ }
```

Return type	Method and Description
void	**enable()** Power on and enable built-in scanner hardware device.
boolean	**isEnable()** Return whether the scanner is enable or not.
void	**disable()** Disable and Power off built-in scanner hardware device.
void	**read()** Start a scan session. The amber light would be turned on until the scan is successful or session timeout.
void	**cancelRead()** Stop an active scan session.

void	**handsfreeRead ()**
	Start a hands-free scan session. The amber light would be turned on until the scan session is stopped.
	Example Usage:
	`if(scannerEnable)`
	`{`
	`  mScanner.handsfreeRead();`
	`  < waiting for 10 seconds >`
	`  mScanner.cancelRead();`
	`}`

Return type	Method and Description

ScannerInfo	**getScannerInfo()**
	Return the information of the scanner.
	Example Usage:
	`ScannerInfo mScannerInfo = mScanner.getScannerInfo();`
	`String name = mScannerInfo.getFriendlyName();`

String	**getFrameworkVer()** Return the version of the scanning framework. Example Usage: `String frameworkVersion = mScanner.getFrameworkVer();`
String	**getScannerFW()** Return the firmware version of the scanner. Example Usage: `String ScannerFrameworkVersion = mScanner.getScannerFW();`
int	**setEncoding(String)** Change the encoding charset used for the decoded data. Change the encoding charset used for the decoded data. Support : UTF-8, GB18030, GBK, HZ-GB-2312, Big5, Big5-HKSCS. Return 1 when successfully set; otherwise, return -1. Example Usage: `mScanner.setEncoding("GBK");`
ScannerConfig	**getConfig()** Get the current configuration for the scanner. Example Usage: `ScannerConfig mScannerConfig = mScanner.getConfig();`

void	**setConfig(ScannerConfig)**
	Change the configuration settings for the scanner.
	Example Usage:
	Following the example code of previous method :
	`mScannerConfig.decoderParams.upca.enabled = false;`
	`mScanner.setConfig(mScannerConfig);`

void	**resetToDefault()**
	Change the configuration of the scanner to scanner default settings.
	Example Usage:
	`if(scannerEnabled)`
	`{`
	`  mScanner.resetToDefault();`
	`}`

void	**addDataListener(DataListener)** Register callback to get data notification of scanning results.

void	**addStatusListener(StatusListener)**
	Register callback to get status notification of scanning results.
	Example Usage:
	`StatusListener mStatusListener = new StatusListener(){ <omitted> };`
	`mScanner.addStatusListener(mStatusListener);`

Return type	Method and Description
void	**removeDataListener (DataListener)** Un-register callback to get data notification of scanning results.

void	**removeStatusListener(StatusListener)**
	Un-register callback to get status notification of scanning results.
	Example Usage:
	`if(mStatusListener != null)`
	`mScanner.removeStatusListener(mStatusListener);`

com.symbol.scanning.Scanner.ScannerVersion

Class to indicate the scanner status.

Example Usage:

```
Scanner mScanner = new Scanner();
String scannerVersion = mScanner.scannerVersion.toString();
```

Return type	Method and Description
String	**toString()** Return the string of scanner API version number.

com.symbol.scanning.BarcodeManager

The primary object to access the barcode scanning feature.

Example Usage:

```
Scanner mScanner =
BarcodeManager.getDevice(BarcodeManager.DeviceIdentifier.INTERNAL_IMAGER1);
try{ <omitted> }
catch(ScannerException se){
   If(se.getResult() == BarcodeManager.ScannerResults.SCANNER_NOT_ENABLED)
{ <omitted> }
}
```

Return type	Method and Description
Scanner	**getDevice(ScannerInfo)** Return the scanner object with the valid ScannerInfo from the supported scanner device list.
Scanner	**getDevice(DeviceIdentifier)** Return the scanner object base on device identifier.
List<ScannerInfo>	**getSupportedDevicesInfo()** Return the list of supported scanner devices information.

Enum	Value
DeviceIdentifier	DEFAULT, INTERNAL_CAMERA1, INTERNAL_IMAGER1, INTERNAL_LASER1, BLUETOOTH_IMAGER1;
ScannerResults	SUCCESS, FAILURE, ALREADY_SCANNING, SCANNER_IN_USE,VF_ERROR, INVALID_VALUE, SCAN_PARAM_NOT_SUPPORTED,SCAN_DATA_FAILURE, SCAN_PARAM_READ_ONLY,LENGTH_MISMATCH, SCANNER_OPERATION_FAILURE,FEATURE_NOT_SUPPORTED, SCANNER_NOT_SUPPORTED,SCANNER_NOT_ENABLED, SCANNER_TIMED_OUT, INVALID_OBJECT,NO_DATA_LISTENER, TRIGGER_KEY_REG_FAILED,TRIGGER_KEY_IN_USE, TRIGGER_KEY_UNREG_FAILED,SCANNER_INIT_FAILURE, SCANNER_DEINIT_FAILURE,UNDEFINED;

com.symbol.scanning.ScannerException

Show the exception when an error occurs.
Example Usage:

```
try{ <omitted> }
catch(ScannerException se){
  If(se.getResult() == BarcodeManager.ScannerResults.SCANNER_NOT_ENABLED)
{ <omitted> }
}
```

Return type	Method and Description
BarcodeManager. ScannerResults	**getResult()** Return the ScannerResult enum value..

com.symbol.scanning.Scanner.DataListener

Callback to notify client on a good deocde.The decoded barcode string would be stored in the argument.
Example Usage:

```
Following the previous example :
DataListener mDataListener = new DataListener(){
  public void onData(ScanDataCollection sdc){<omitted> }
```

```
} ;
mScanner.addDataListener(mDataListener);
```

Return type	Method and Description
void	**onData(ScanDataCollection)** Callback function to deal with returned ScanDataCollection.

com.symbol.scanning.Scanner.StatusListener

Callback to notify client on the status of the scanner. The status string would be stored in the argument.
Example Usage:

```
Following the previous example:
StatusListener mStatusListener = new StatusListener(){
  public void onStatus(StatusData sd){ <omitted> }
};
mScanner.addStatusListener(mStatusListener);
```

Return type	Method and Description
StatusData	**onStatus(StatusData)** Callback function to deal with returned StatusData.

com.symbol.scanning.StatusData

Class to indicate the scanner status.
Example Usage:

```
Following the previous example :
  public void onStatus(StatusData sd){
   if(StatusData.ScannerStates.IDLE ==sd.getState())
{ … }
}
```

Return type	Method and Description
String	**getFriendlyName()** Return the name of scanner for which the status data is returned.
ScannerState	**getState()** Return the state of scanner.

Enum	Value
ScannerStates	IDLE, WAITING, SCANNING, DISABLED;

com.symbol.scanning.ScanDataCollection

The ScanDataCollection object gives scanning result and the collection of ScanData.
Example Usage:

```
public void onData(ScanDataCollection sdc)
    {
      ArrayList<ScanData> scanDataList = sdc.getScanData();
      ...
}
```

Return type	Method and Description
String	**getFriendlyName()** Return the name of the scanner for which the data is returned.
ScannerResults	**getResult()** Return the scanned result.
ArrayList<ScanData>	**getScanData()** Return the scan data.

Enum	Value
LabelType	AUSPOSTAL, AZTEC, BOOKLAND, CANPOSTAL, CHINESE_2OF5, CODABAR, CODE11, CODE128, CODE32, CODE39, CODE93, COMPOSITE_AB, COMPOSITE_C, COUPON, D2OF5, DATAMATRIX, DUTCHPOSTAL, DATABAR_COUPON, EAN128, EAN13, EAN8, GS1_DATABAR, GS1_DATABAR_EXP, GS1_DATABAR_LIM, I2OF5, IATA2OF5, ISBT128, JAPPOSTAL, KOREAN_3OF5, MATRIX_2OF5, MAXICODE, MICROPDF, MICROQR, MSI, OCR, PDF417, QRCODE, SIGNATURE, TLC39, TRIOPTIC39, UKPOSTAL, UPCA, UPCE0, UPCE1, US4STATE, US4STATE_FICS, USPLANET, USPOSTNET, WEBCODE, UNDEFINED;

com.symbol.scanning.ScanDataCollection.ScanData

Class to store the information on the scanned barcode data.
Example Usage:

```
String data = scanData.getData();
```

Return type	Method and Description
String	**getData()** Return barcode data.
LabelType	**getLabelType()** Return label type of scanned barcode data.
String	**getTimeStamp()** Return the time at which the barcode was scanned.

com.symbol.scanning.ScannerConfig

Class that provides access to scanner configuration settings.

Example Usage:

```
ScannerConfig scanconfig = mScanner.getConfig;
boolean upcaDecode =
scanconfig.isParamSupported( “scannerconfig.decoderParams.upca.enabled” );
```

Return type	Method and Description
boolean	**isParamSupported(String)** Return whether the specified parameter is supported or not.

Type	Member
DecoderParams	decoderParams
ReaderParams	readerParams
ScanParams	scanParams
SkipOnUnSupported	skipOnUnsupported

Enum	Value
AudioStreamType	RINGER, MEDIA, ALARMS;
BooklandFormat	ISBN_10, ISBN_13;
CheckDigit	ONE, TWO;
CheckDigitScheme	NO, USS, OPCC;
CodeIdType	NONE, AIM, SYMBOL;

Enum	Value
CouponReport	OLD, NEW, BOTH;
Inverse1DMode	DISABLED, ENABLED, AUTO;
Isbt128ContactMode	NEVER, ALWAYS, AUTO;
LcdMode	DISABLED, ENABLED;
LinearSecurityLevel	SHORT_OR_CODABAR, ALL_TWICE, LONG_AND_SHORT, ALL_THRICE;
PickList	DISABLED, ENABLED;
Preamble	NONE, SYS_CHAR, COUNTRY_AND_SYS_CHAR;
SecurityLevel	LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3;
SkipOnUnSupported	NONE, UNSUPPORTED_PARAM, UNSUPPORTED_VALUE, ALL;
SupplementalMode	NO, ALWAYS, AUTO, SMART, S_378_379, S_978_979, S_414_419_434_439, S_977;
UccLinkMode	LINK_FLAG_IGNORED, ALWAYS_LINKED, AUTO_DISCRIMINATE;
VerifyCheckDigit	NO, ONE, TWO;
ViewFinderMode	ENABLED, STATIC_RECTICLE;
PowerMode	LOW, OPTIMIZED, HIGH, ALWAYS_ON;

com.symbol.scanning.ScannerConfig.DecoderParams

Class contains the decoder parameters.
Example Usage:

```
ScannerConfig scanconfig = mScanner.getConfig;
```

Type	Member
Upca	upca
Upce0	upce0

Type	Member
EAN13	ean13
EAN8	ean8
Code128	code128
Code39	code39
I2of5	i2of5
GS1Databar	gs1Databar
GS1DatabarExp	gs1DatabarExp
GS1DatabarLim	gs1DatabarLim
CodaBar	codaBar
MSI	msi
Code93	code93
TriOptic39	triOptic39
D2of5	d2of5
Chinese2of5	chinese2of5
Code11	code11
Matrix2of5	matrix2of5
Upce1	upce1
DataMatrix	dataMatrix
QrCode	qrCode
Pdf417	pdf417
CompositeAB	compositeAB
CompositeC	compositeC

Type	Member
MicroQr	microQR
Aztec	aztec
MaxiCode	maxiCode
MicroPdf	microPDF
UsPostNet	usPostNet
UsPlanet	usPlanet
UkPostal	ukPostal
JapanesePostal	japanesePostal
Korean3of5	korean3of5
AustralianPostal	australianPostal
Us4State	us4State
Us4StateFics	us4StateFics
HanXin	hanXin
Tlc39	tlc39
UpcEanParams	upcEanParams

com.symbol.scanning.ScannerConfig.DecoderParams.Upca

Class that provides access to parameters available for the Upca decoder.
Example Usage:

```
Following the previous example code :
scanconfig.decoderParams.upca.enabled = true;
scanconfig.decoderParams.upca.preamble = ScannerConfig.Preamble.NONE;
```

Type	Member
boolean	enabled

Type	Member
Preamble	preamble
boolean	reportCheckDigit

com.symbol.scanning.ScannerConfig.DecoderParams.Upce0

Class that provides access to parameters available for the Upce0 decoder.
Example Usage:

```
scanconfig.decoderParams.upce0.enabled = true;
```

Type	Member
boolean	enabled
Preamble	preamble
boolean	reportCheckDigit
boolean	convertToUpca

com.symbol.scanning.ScannerConfig.DecoderParams.EAN13

Class that provides access to parameters available for the EAN13 decoder.
Example Usage:

```
scanconfig.decoderParams.ean13.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.EAN8

Class that provides access to parameters available for the EAN8 decoder.
Example Usage:

```
scanconfig.decoderParams.ean8.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.Code128

Class that provides access to parameters available for the Code128 decoder.

Example Usage:

```
scanconfig.decoderParams.code128.enabled = true;
```

Type	Member
boolean	enabled
int	length1
int	length2
boolean	redundancy
boolean	enableIsbt128
boolean	enableEan128
ScannerConfig. Isbt128ContactMode	isbt128ConcatMode
boolean	checkIsbtTable

com.symbol.scanning.ScannerConfig.DecoderParams.Code39

Class that provides access to parameters available for the Code39 decoder.

Example Usage:

```
scanconfig.decoderParams.code39.enabled = true;
```

Type	Member
boolean	enabled
int	length1

Type	Member
int	length2
boolean	verifyCheckDigit
boolean	reportCheckDigit
boolean	fullAscii
boolean	convertToCode32
boolean	reportCode32Prefix

com.symbol.scanning.ScannerConfig.DecoderParams.I2of5

Class that provides access to parameters available for the I2of5 decoder.
Example Usage:

```
scanconfig.decoderParams.i2of5.enabled = true;
```

Type	Member
boolean	enabled
int	length1
int	length2
VerifyCheckDigit	verifyCheckDigit
boolean	reportCheckDigit
boolean	convertToEan13

com.symbol.scanning.ScannerConfig.DecoderParams.GS1Databar

Class that provides access to parameters available for the GS1 Databar decoder.
Example Usage:

```
scanconfig.decoderParams.gs1Databar.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.GS1DatabarExp

Class that provides access to parameters available for the GS1 Databar Exp decoder.

Example Usage:

```
scanconfig.decoderParams.gs1DatabarExp.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.GS1DatabarLim

Class that provides access to parameters available for the GS1 Databar Lim decoder.

Example Usage:

```
scanconfig.decoderParams.gs1DatabarLim.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.CodaBar

Class that provides access to parameters available for the CodaBar decoder.

Example Usage:

```
scanconfig.decoderParams.codaBar.enabled = true;
```

Type	Member
boolean	enabled
int	length1
int	length2

Type	Member
boolean	clsiEditing
boolean	notisEditing

com.symbol.scanning.ScannerConfig.DecoderParams.MSI

Class that provides access to parameters available for the MSI decoder.

Example Usage:

```
scanconfig.decoderParams.msi.enabled = true;
```

Type	Member
boolean	enabled
int	length1
int	length2
CheckDigitScheme	checkdigitscheme
CheckDigit	checkDigits
boolean	reportCheckDigit

com.symbol.scanning.ScannerConfig.DecoderParams.Code93

Class that provides access to parameters available for the Code93 decoder.

Example Usage:

```
scanconfig.decoderParams.code93.enabled = true;
```

Type	Member
boolean	enabled
int	length1
int	length2

com.symbol.scanning.ScannerConfig.DecoderParams.TriOptic39

Class that provides access to parameters available for the TriOptic39 decoder.
Example Usage:

```
scanconfig.decoderParams.triOptic39.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.D2of5

Class that provides access to parameters available for the D2of5 decoder.
Example Usage:

```
scanconfig.decoderParams.d2of5.enabled = true;
```

Type	Member
boolean	enabled
int	length1
int	length2

com.symbol.scanning.ScannerConfig.DecoderParams.Chinese2of5

Class that provides access to parameters available for the Chinese2of5 decoder.
Example Usage:

```
scanconfig.decoderParams.chinese2of5.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.Code11

Class that provides access to parameters available for the Code11 decoder.decoder.
Example Usage:

```
scanconfig.decoderParams.code11.enabled = true;
```

Type	Member
boolean	enabled
int	length1
int	length2
boolean	reportCheckDigit
VerifyCheckDigit	verifyCheckDigit

com.symbol.scanning.ScannerConfig.DecoderParams.Matrix2of5

Class that provides access to parameters available for the Matrix2of5 decoder.

Example Usage:

```
scanconfig.decoderParams.matrix2of5.enabled = true;
```

Type	Member
boolean	enabled
int	length1
int	length2
boolean	reportCheckDigit
boolean	reportCheckDigit
boolean	verifyCheckDigit

com.symbol.scanning.ScannerConfig.DecoderParams.Upce1

Class that provides access to parameters available for the Upce1 decoder.

Example Usage:

```
scanconfig.decoderParams.upce1.enabled = true;
```

Type	Member
boolean	enabled

Type	Member
Preamble	Preamble
boolean	convertToUpca
boolean	reportCheckDigit

com.symbol.scanning.ScannerConfig.DecoderParams.DataMatrix

Class that provides access to parameters available for the DataMatrix decoder.

Example Usage:

```
scanconfig.decoderParams.dataMatrix.enabled = true;
```

Type	Member
boolean	enabled
Inverse1DMode	inverse

com.symbol.scanning.ScannerConfig.DecoderParams.QrCode

Class that provides access to parameters available for the QR Code decoder.

Example Usage:

```
scanconfig.decoderParams.qrCode.enabled = true;
```

Type	Member
boolean	enabled
Inverse1DMode	inverse

com.symbol.scanning.ScannerConfig.DecoderParams.Pdf417

Class that provides access to parameters available for the Pdf417 decoder.

Example Usage:

```
scanconfig.decoderParams.pdf417.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.CompositeAB

Class that provides access to parameters available for the CompositeAB decoder.
Example Usage:

```
scanconfig.decoderParams.compositeAB.enabled = true;
```

Type	Member
boolean	enabled
UccLinkMode	uccLinkMode

com.symbol.scanning.ScannerConfig.DecoderParams.CompositeC

Class that provides access to parameters available for the CompositeC decoder.
Example Usage:

```
scanconfig.decoderParams.compositeC.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.MicroQr

Class that provides access to parameters available for the MicroQr decoder.
Example Usage:

```
scanconfig.decoderParams.microQR.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.Aztec

Class that provides access to parameters available for the Aztec decoder.

Example Usage:

```
scanconfig.decoderParams.aztec.enabled = true;
```

Type	Member
boolean	enabled
Inverse1DMode	inverse

com.symbol.scanning.ScannerConfig.DecoderParams.MaxiCode

Class that provides access to parameters available for the MaxiCode decoder.

Example Usage:

```
scanconfig.decoderParams.maxiCode.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.MicroPdf

Class that provides access to parameters available for the MicroPdf decoder.

Example Usage:

```
scanconfig.decoderParams.microPDF.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.MicroPdf

Class that provides access to parameters available for the MicroPdf decoder.

Example Usage:

```
scanconfig.decoderParams.usPostNet.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.UsPlanet

Class that provides access to parameters available for the UsPlanet decoder.
Example Usage:

```
scanconfig.decoderParams.usPlanet.enabled = true;
```

Type	Member
boolean	enabled
boolean	reportCheckDigit

com.symbol.scanning.ScannerConfig.DecoderParams.UkPostal

Class that provides access to parameters available for the UkPostal decoder.
Example Usage:

```
scanconfig.decoderParams.ukPostal.enabled = true;
```

Type	Member
boolean	enabled
boolean	reportCheckDigit

com.symbol.scanning.ScannerConfig.DecoderParams.JapanesePostal

Class that provides access to parameters available for the JapanesePostal decoder.
Example Usage:

```
scanconfig.decoderParams.japanesePostal.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.Korean3of5

Class that provides access to parameters available for the Korean3of5 decoder.

Example Usage:

```
scanconfig.decoderParams.korean3of5.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.AustralianPostal

Class that provides access to parameters available for the AustralianPostal decoder.

Example Usage:

```
scanconfig.decoderParams.australianPostal.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.Us4State

Class that provides access to parameters available for the Us4State decoder.

Example Usage:

```
scanconfig.decoderParams.us4State.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.Us4StateFics

Class that provides access to parameters available for the Us4StateFics decoder

Example Usage:

```
scanconfig.decoderParams.us4StateFics.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.HanXin

Class that provides access to parameters available for the HanXin decoder.

Example Usage:

```
scanconfig.decoderParams.hanXin.enabled = true;
```

Type	Member
boolean	enabled
Inverse1DMode	inverse

com.symbol.scanning.ScannerConfig.DecoderParams.Tlc39

Class that provides access to parameters available for the Tlc39 decoder.

Example Usage:

```
scanconfig.decoderParams.tlc39.enabled = true;
```

Type	Member
boolean	enabled

com.symbol.scanning.ScannerConfig.DecoderParams.UpcEanParams

Class that provides access to parameters available for the UPC/EAN decoder.

Example Usage:

```
Following the previous example code:
scanconfig.decoderParams.upca.enabled == true;
scanconfig.decoderParams.  upcEanParams.securityLevel == ScannerConfig.SecurityLevel.LEVEL_0;
```

Type	Member
SecurityLevel	securityLevel
SupplementalMode	supplementalMode
boolean	booklandCode
boolean	couponCode
CouponReport	couponReport

Type	Member
boolean	eanZeroExtend
BooklandFormat	booklandFormat
boolean	convertDataBarToUpcEan
int	supplementalRetries

com.symbol.scanning.ScannerConfig.ReaderParams

Class holds the parameters for a scanner.

Type	Member
ReaderSpecific	readerSpecific

com.symbol.scanning.ScannerConfig.ReaderParams.ReaderSpecific

Class provides access to the reader specific parameters.

Type	Member
ImagerSpecific	imagerSpecific
LaserSpecific	laserSpecific

com.symbol.scanning.ScannerConfig.ReaderParams.ReaderSpecific.ImagerSpecific

Class provides access to the imager specific parameters.

Example Usage:

```
Following the previous example code:
scanconfig.readerParams.readerSpecific.imagerSpecific.inverse1DMode ==
ScannerConfig.Inverse1DMode.DISABLED;
```

Type	Member
int	illuminationBrightness

Type	Member
Inverse1DMode	inverse1DMode
LcdMode	lcdMode
PickList	pickList

com.symbol.scanning.ScannerConfig.ReaderParams.ReaderSpecific.LaserSpecific

Class provides access to the laser scanner specific parameters.

Example Usage:

```
Following the previous example code:
scanconfig.readerParams.readerSpecific.laserSpecific.linearSecurityLevel ==
ScannerConfig.LinearSecurityLevel.ALL_TWICE;
```

Type	Member
int	beamTimer
LinearSecurityLevel	linearSecurityLevel
Inverse1DMode	inverse1DMode
PowerMode	powerMode

com.symbol.scanning.ScannerConfig.ScanParams

Class provides access to scanning parameters available for all decoders.

Example Usage:

```
Following the previous example code:
scanconfig.scanParams.audioStreamType == ScannerConfig.AudioStreamType.MEDIA;
```

Type	Member
AudioStreamType	audioStreamType
CodeIdType	codeIdType
String	decodeAudioFeedbackUri

boolean	decodeHapticFeedback
boolean	decodeLEDFeedback
int	decodeLEDTime
String	encoding

com.symbol.scanning.ScannerInfo

Class allows enumeration of the scanners capabilities
Example Usage:

```
Scanner mScanner = new Scanner();
String name = mScanner.getScannerInfo().getFriendlyName();
```

Return type	Method and Description
ConnectionType	**getConnectionType()** Return the connection type to the mobile device.
DecoderType	**getDecoderType()** Return the barcode scanning type supported by scanner.
DeviceType	**getDeviceType()** Return the scanner device type.
String	**getFriendlyName()** Return the name of the scanner.
String	**getModelNumber()** Return the model number of the scanner.
boolean	**isConnected()** Return if the scanner connected to the device.
boolean	**isDefaultScanner()** Return if the scanner is the default scanner.

Enum	Description
ConnectionType	INTERNAL, BLUETOOTH_SSI, SERIAL_SSI, USB, UNDEFINED;
DecoderType	ONE_DIMENSIONAL, TWO_DIMENSIONAL,UNDEFINED;
DeviceType	CAMERA, IMAGER, LASER, UNDEFINED;

com.symbol.scanning.ProfileManager

Class handles all the config related functions.

Example Usage:

```
ProfileConfig mProfileConfig = new ProfileConfig();
ProfileManager.RESULT_STATUS mResultStatus;
mResultStatus = mProfileManager.processProfile(ProfileManager.PROFILE_FLAG.GET, mProfileConfig);
```

Return type	Method and Description
void	**release()** Release ProfileManager.
RESULT_STATUS	**processProfile(PROFILE_FLAG, ProfileConfig)** Processes the given profile based on the data provided and the flag and return status of the action.

Enum	Value
PROFILE_FLAG	SET, GET;
RESULT_STATUS	SUCCESS, FAILURE, SCANCONFIG_NOT_OPENED;

com.symbol.scanning.ProfileConfig

Class to indicate the config status.

Type	Member
ActivitySelection	ActivitySelection
QuickLaunch	QuickLaunch
DataCapture	DataCapture
String	String
String	String

Enum	Value
ACTION_KEY_CHAR	DEFAULT, NONE, TAB, LINEFEED, CARRIAGERETURN;

Enum	Value
BOOKLAND_FORMAT	DEFAULT, ISBN_10, ISBN_13;
CHARACTER_ENCODING	DEFAULT, UTF8, GB18030, GBK, BIG5, BIG5_HKSCS;
CHECK_DIGIT	DEFAULT, NO, ONE, TWO;
CHECK_DIGIT_SCHEME	DEFAULT, MOD_11_10, MOD_10_10;
CHECK_DIGIT_TYPE	DEFAULT, NO, USS, OPCC;
CODE_ID_TYPE	DEFAULT, NONE, AIM, SYMBOL;
COUPON_REPORT	DEFAULT, OLD, NEW, BOTH;
ENABLED_STATE	DEFAULT, FALSE, TRUE;
ILLUMINATION_MODE	DEFAULT, OFF, ON;
INVERSE_1D_MODE	DEFAULT, DISABLED, ENABLED, AUTO;
ISBT128_CONTACT_MODE	DEFAULT, NEVER, ALWAYS, AUTO;
LCD_MODE	DEFAULT, DISABLED, ENABLED;
LINEAR_SECURITY_LEVEL	DEFAULT, SHORT_OR_CODABAR, ALL_TWICE, LONG_AND_SHORT,ALL_THRICE;
PICK_LIST	DEFAULT, DISABLED,ENABLED;
POWER_MODE	DEFAULT, CONTINUOUS_POWER, LOW_POWER;
PREAMBLE	DEFAULT, NONE, SYS_CHAR, COUNTRY_AND_SYS_CHAR;
ACTION_KEY_CHAR	DEFAULT, NONE, TAB, LINEFEED, CARRIAGERETURN;
SECURITY_LEVEL	DEFAULT, LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3;
SUPPLEMENTAL_MODE	DEFAULT, NO, ALWAYS, AUTO, SMART, S_378_379, S_978_979, S_414_419_434_439, S_977;
UCC_LINK_MODE	DEFAULT, LINK_FLAG_IGNORED, ALWAYS_LINKED, AUTO_DISCRIMINATE;
VERIFY_CHECK_DIGIT	DEFAULT, NO, ONE, TWO;
VOLUME_TYPE	DEFAULT, RINGER, MEDIA, ALARMS;

com.symbol.scanning.ProfileConfig.ActivitySelection

Class for dealing with ActivitySelection of profile data.

Example Usage:

```
Following the previous example code:
ArrayList<ActivityElement> mAppList = mProfileConfig.activitySelection.activities;
```

Type	Member
ENABLED_STATE	activities

com.symbol.scanning.ProfileConfig.ActivitySelection.ActivityElement

Class to store application's package name and its activities to the profile.

Type	Member
String	activities
String	packageName

Constructor
ActivityElement (String activities, String packageName) Create a ActivityElement object with package name and activities of an application.

com.symbol.scanning.ProfileConfig.QuickLaunch

Class contains the enablement to trigger ScanDemo in Launch applications with scan keys.

Example Usage:

```
Following the previous example code:
mProfileConfig.quickLaunch.quick_launch = ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	quick_launch

com.symbol.scanning.ProfileConfig.DataCapture

Class for dealing with data capture profile data.

Type	Member
Barcode	barcode
DataDelivery	datadelivey

com.symbol.scanning.ProfileConfig.DataCapture.Barcode

Class that holds Barcode configuration settings.

Type	Member
DecoderParams	decoderParams
Decoders	decoders
ReaderParams	readerParams
ScanParams	scanParams
UpcEanParams	upcEanParams
ENABLED_STATE	scanner_input_enabled

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.Decoders

Class that provides access to enable or disable decoder barcode symbologies.
Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoders.aztec = ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	australian_postal
ENABLED_STATE	aztec
ENABLED_STATE	chinese_2of5

Type	Member
ENABLED_STATE	codabar
ENABLED_STATE	code11
ENABLED_STATE	code128
ENABLED_STATE	code39
ENABLED_STATE	code93
ENABLED_STATE	composite_ab
ENABLED_STATE	composite_c
ENABLED_STATE	d2of5
ENABLED_STATE	datamatrix
ENABLED_STATE	ean13
ENABLED_STATE	ean8
ENABLED_STATE	gs1_databar
ENABLED_STATE	gs1_databar_exp
ENABLED_STATE	gs1_databar_lim
ENABLED_STATE	i2of5
ENABLED_STATE	japanese_postal
ENABLED_STATE	korean_3of5
ENABLED_STATE	matrix_2of5
ENABLED_STATE	maxicode
ENABLED_STATE	micropdf
ENABLED_STATE	microqr
ENABLED_STATE	msi

Type	Member
ENABLED_STATE	pdf417
ENABLED_STATE	qrcode
ENABLED_STATE	tlc39
ENABLED_STATE	trioptic39
ENABLED_STATE	uk_postal
ENABLED_STATE	upca
ENABLED_STATE	upce0
ENABLED_STATE	upce1
ENABLED_STATE	us4state
ENABLED_STATE	us4state_fics
ENABLED_STATE	usplanet
ENABLED_STATE	uspostnet
ENABLED_STATE	hanxin

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams

Class that holds Barcode configuration settings.

Type	Member
Codabar	codabar
Code11	code11
Code128	code128
Code39	code39
Code93	code93

Type	Member
Composite_AB	composite_AB
Discrete_2of5	discrete_2of5
Interleaved_2of5	interleaved_2of5
Matrix_2of5	matrix_2of5
MSI	msi
UK_Postal	uk_Postal
US_Planet	us_Planet
UPCA	upca
UPCE0	upce0
UPCE1	upce1
HanXin	hanxin
DataMatrix	datamatrix
Aztec	aztec
QrCode	qrcode

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Codabar

Class that holds Decoder configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.codabar.clsi_editing =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	clsi_editing
Int	length1

Type	Member
Int	length2
ENABLED_STATE	notis_editing

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Code11

Class that holds Code 11 configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.code11.report_check_digit =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
int	length1
int	length2
ENABLED_STATE	report_check_digit
VERIFY_CHECK_DIGIT	verify_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Code128

Class that holds Code128 configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.code128.enable_plain =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	check_isbt_table
ENABLED_STATE	enable_ean128
ENABLED_STATE	enable_plain

Type	Member
ISBT128_CONTACT_MODE	isbt128_contact_mode
Int	length1
Int	length2

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Code39

Class that holds Code39 configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.code39.full_ascii =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	convert_to_code32
ENABLED_STATE	full_ascii
Int	length1
Int	length2
ENABLED_STATE	report_check_digit
ENABLED_STATE	report_code32_prefix
ENABLED_STATE	verify_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Code93

Class that holds Code93 configuration settings.

Example Usage:

```
Following the previous example code :
mProfileConfig.dataCapture.barcode.decoderParams.code93.length1 = 10;
```

Type	Member
Int	length1
Int	length2

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Composite_AB

Class that holds Composite AB configuration settings.
Example Usage:

```
Following the previous example code :
mProfileConfig.dataCapture.barcode.decoderParams.composite_AB.ucc_link_mode =
ProfileConfig.UCC_LINK_MODE.ALWAYS_LINKED;
```

Type	Member
UCC_LINK_MODE	ucc_link_mode

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Discrete_2of5

Class that holds Discrete 2 of 5 configuration settings.
Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.discrete_2of5.length1 = 10;
```

Type	Member
Int	length1
Int	length2

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Interleaved_2of5

Class that holds Interleaved 2 of 5 configuration settings.
Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.interleaved_2of5.length1 = 10;
```

Type	Member
CHECK_DIGIT	check_digit
ENABLED_STATE	convert_itf14_to_ean13
Int	length1
Int	length2
ENABLED_STATE	report_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Matrix_2of5

Class that holds Matrix 2 of 5 configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.matrix_2of5.redundancy =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
Int	length1
Int	length2
ENABLED_STATE	redundancy
ENABLED_STATE	report_check_digit
ENABLED_STATE	verify_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.MSI

Class that holds MSI configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.msi.report_check_digit =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
CHECK_DIGIT	check_digit
CHECK_DIGIT_SCHEME	check_digit_scheme
Int	length1
Int	length2
ENABLED_STATE	report_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.UK_Postal

Class that holds UK Postal configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.uk_Postal.report_check_digit =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	report_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.US_Planet

Class that holds US Planet configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.us_Planet.report_check_digit =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	report_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.UPCA

Class that holds UPCA configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.upca.report_check_digit =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
PREAMBLE	preamble
ENABLED_STATE	report_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.UPCE0

Class that holds UPCE0 configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.upce0.convert_to_upca =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	convert_to_upca
PREAMBLE	preamble
ENABLED_STATE	report_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.UPCE1

Class that holds UPCE1 configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.upce1.convert_to_upca =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	convert_to_upca
PREAMBLE	preamble
ENABLED_STATE	report_check_digit

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.HanXin

Class that holds HanXin configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.hanxin.inverse_1d_mode =
ProfileConfig.INVERSE_1D_MODE.DISABLED;
```

Type	Member
INVERSE_1D_MODE	inverse_1d_mode

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.DataMatrix

Class that holds DataMatrix configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.datamatrix.inverse_1d_mode =
ProfileConfig.INVERSE_1D_MODE.DISABLED;
```

Type	Member
INVERSE_1D_MODE	inverse_1d_mode

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.Aztec

Class that holds Aztec configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.aztec.inverse_1d_mode =
ProfileConfig.INVERSE_1D_MODE.DISABLED;
```

Type	Member
INVERSE_1D_MODE	inverse_1d_mode

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.DecoderParams.QrCode

Class that holds QR code configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.decoderParams.qrcode.inverse_1d_mode =
ProfileConfig.INVERSE_1D_MODE.DISABLED;
```

Type	Member
INVERSE_1D_MODE	inverse_1d_mode

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.ReaderParams

Class that holds reader configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.readerParams.beam_timer = 5000;
```

Type	Member
int	beam_timer
ILLUMINATION_MODE	illumination_mode
INVERSE_1D_MODE	inverse_1d_mode
LCD_MODE	lcd_mode
LINEAR_SECURITY_LEVEL	linear_security_level

Type	Member
POWER_MODE	power_mode
PICK_LIST	picklist
int	illumination_brightness

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.ScanParams

Class that holds scanning configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.scanParams.decode_haptic_feedback =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
CODE_ID_TYPE	code_id_type
String	decode_audio_feedback_uri
ENABLED_STATE	decode_haptic_feedback
int	good_decode_led_timer
ENABLED_STATE	led_feedback
VOLUME_TYPE	volume_type
CHARACTER_ENCODING	character_encoding

com.symbol.scanning.ProfileConfig.DataCapture.Barcode.UpcEanParams

Class that holds Upc Ean Params configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.upcEanParams.bookland =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	bookland
BOOKLAND_FORMAT	bookland_format
ENABLED_STATE	coupon
COUPON_REPORT	coupon_report
ENABLED_STATE	databar_to_upc_ean
int	retry_count
SECURITY_LEVEL	security_level
SUPPLEMENTAL_MODE	supplemental_mode
ENABLED_STATE	ean_zero_extend

com.symbol.scanning.ProfileConfig.DataCapture.DataDelivery

Class for dealing with DataDelivery profile data.

Type	Member
Intent	intent
Keystroke	keystroke

com.symbol.scanning.ProfileConfig.DataCapture.DataDelivery.BasicDataFormatting

Class that holds Basic Data Formatting configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.barcode.upcEanParams.bookland =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ENABLED_STATE	bdf_enabled

Type	Member
BOOKLAND_FORMAT	bdf_send_data
ENABLED_STATE	bdf_send_enter
ENABLED_STATE	bdf_send_hex
ENABLED_STATE	bdf_send_tab
String	bdf_prefix
String	bdf_suffix

com.symbol.scanning.ProfileConfig.DataCapture.DataDelivery.Intent

Class that holds Intent configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.datadelivey.intent.output_enabled =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ProfileConfig.DataCapture.DataDelivery.BasicDataFormatting	basicDataFormatting
String	action
String	category
ENABLED_STATE	output_enabled

com.symbol.scanning.ProfileConfig.DataCapture.DataDelivery.Keystroke

Class that holds Keystroke configuration settings.

Example Usage:

```
Following the previous example code:
mProfileConfig.dataCapture.datadelivey.keystroke.ime_output_enabled =
ProfileConfig.ENABLED_STATE.TRUE;
```

Type	Member
ProfileConfig.DataCapture.DataDelivery.BasicDataFormatting	basicDataFormatting
ACTION_KEY_CHAR	keystroke_action_char
ENABLED_STATE	ime_output_enabled

TouchInputManager API

com.symbol.touch.TouchInputManager

Class to get/set the operation mode of touch panel.

Return type	Method and Description

Void	**SetOperationMode()**
	Set the touch panel operation mode.
	Example Usage:
	`TouchInputManager mTouchInputManager = new TouchInputManager();`
	`mTouchInputManager.SetOperationMode(1); // 0 and 1: Finger mode, 2: Stylus (Glove) mode`

String	**GetOperationMode()**
	Return the touch panel operation mode.
	Example Usage:
	`TouchInputManager mTouchInputManager = new TouchInputManager();`
	`String mode = mTouchInputManager.GetOperationMode();`

Intent API

There are some customized Intents defined in MC36 to do system control:

com.symbol.actions.DISBLE_DEVICE_RESET

Class that holds Keystroke configuration settings.

```
Intent intent = new Intent();
String content = "com.symbol.actions.DISBLE_DEVICE_RESET"
intent.setAction(content);
sendBroadcast(intent);
```

com.symbol.actions.ENABLE_DEVICE_RESET

Enable Enterprise data reset option in Settings.

```
Intent intent = new Intent();
String content = "com.symbol.actions.ENABLE_DEVICE_RESET"
intent.setAction(content);
sendBroadcast(intent);
```

com.symbol.intent.action.HOMEKEY_MODE

Activate/Inactivate the capacitive Home button.

```
Intent intent = new Intent();
intent.setAction("com.symbol.intent.action.HOMEKEY_MODE");
intent.putExtra("enable", 1); // 1: Disable Home key, 0: Enable Home key
sendBroadcast(intent);
```

Mediatek API

MC36 is based on Mediatek chipsets platform which offers below extra packages:

-com.mediatek.build

-com.mediatek.camcorder

-com.mediatek.hardware

-com.mediatek.hotknot

-com.mediatek.media

-com.mediatek.telephony

A complementary Mediatek API Reference is provided on the MediaTek Labs site. The API level of MC36 devices is level 2.

Chapter 3 ADB USB Setup

To use the ADB, install the USB driver. This assumes that the development SDK has been installed on the host computer. Go to http://developer.android.com/sdk/index.html for details on setting up the development SDK. ADB driver for Windows and Linux are available on the Zebra Support Central web site at http://www.zebra.com/support. Download the ADB and USB Driver Setup package. Following the instructions with the package to install the ADB and USB drivers for Windows and Linux.

Chapter 4
MTK Debug Logging

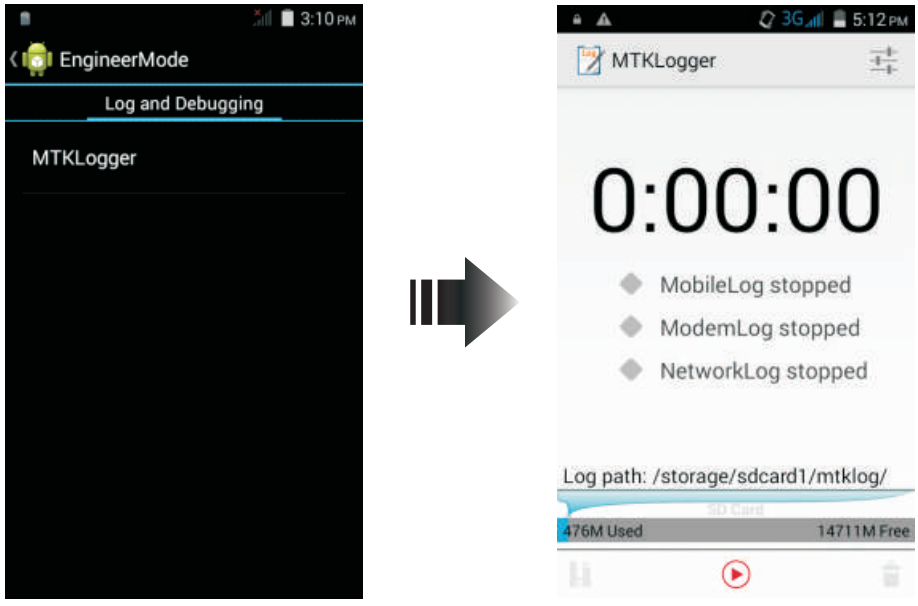
MTKLogger

MTKLogger is an application that contains Mobile Log, Modem Log, Network Log and System Logger. It allows users to implement the log operations with only one UI.

Open MTKLogger

1. Boot up MC36.
2. Enter Engineering mode and open MTKLogger.
   Phone->Dial *#*#3646633#*#*-> MTKLogger.

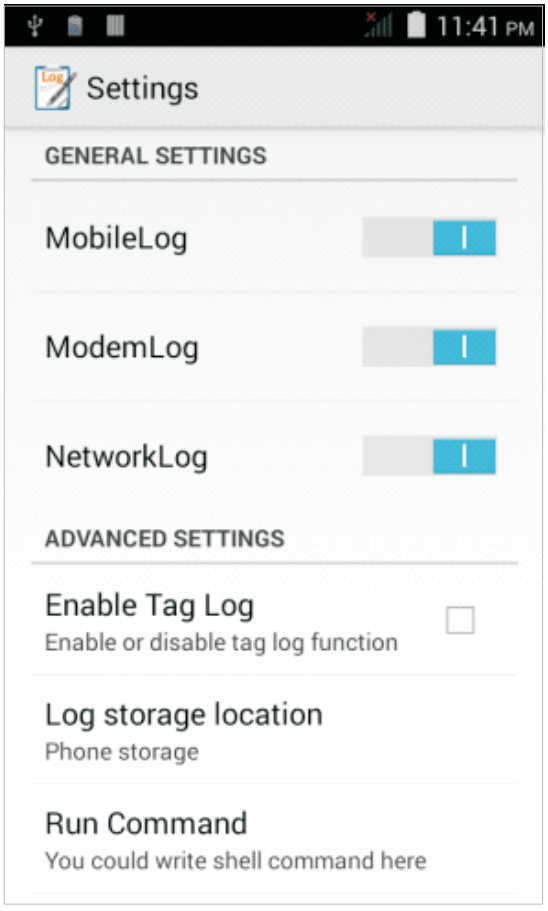
   Figure 4-1:



Configurations

Click the configuration button on MTKLogger UI.

Figure 4-2:



Mobile Log

Android Log

Set and android main log will be recorded while Mobile Log is on.

Kernel Log

Set and kernel log will be recorded while Mobile Log is on.

Bluetooth Log

Set and bluetooth log will be recorded while Mobile Log is on.

Limit Current Log Size(MB)

The maximum log size of Mobile Log for this start/stop period.

Limit Total Log Size(MB)

The total log size, including current recording log and former log, of Mobile Log.

Start Automatically

Set and Mobile Log will be turned on automatically when booting up.

Mobile Log

Log Mode
Select where the recorded log will be restored, usually SD mode.
USB mode is set if Modem Log will be caught with “Catcher”.
Limit Log Size(MB)
The total log folder size of Modem Log.
Start Automatically
Set and Modem Log will be turned on automatically when booting up.

Network Log

Enable environment check
Set to check current network connection status.
If it is set, it will ping 2 IP to confirm network connection status and take few seconds to stop log.
Enable package limitation
Set to limit the log size of each network log package.
Limited package size
Set the limitation size for each network package.
Limit Log Size(MB)
The total log folder size of Network Log.
Start Automatically
Set and Network Log will be turned on automatically when booting up.

Advance settings

Enable Tag Log
Set to check current network connection status.
If it is set, it will ping 2 IP to confirm network connection status and take few seconds to stop log.
Log storage location
Set to limit the log size of each network log package.
Run Command
Set the limitation size for each network package.

Start Logging

Press  button to start logging debug message.

Stop Logging

Press  button to stop logging and save debug log when test is completed.

Clear All Previous Logs

1.Press 
2.Press CLEAN ALL
3.Press OK

Extracting Log Files

1.Connect the device to host PC with USB cable.
2.Use a file explorer and navigate to /Phone storage/mtklog/
3.Copy the files from device to host PC

Chapter 5 Key Remap

Button Remapping

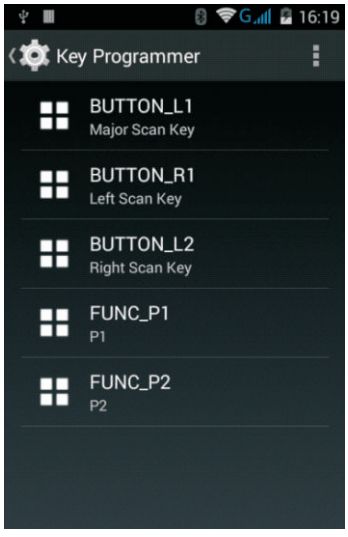
The MC36's buttons can be programmed to perform different functions or shortcuts to installed applications.

- Major Scan Key- Center scan button.
- Left Scan Key – Left scan button.
- Right Scan Key – Right scan button.
- P1 – IME Swich.
- P2 – LCD brightness bar.

Remapping a Button

1.Touch Settings.
2.Touch Key Programmer.

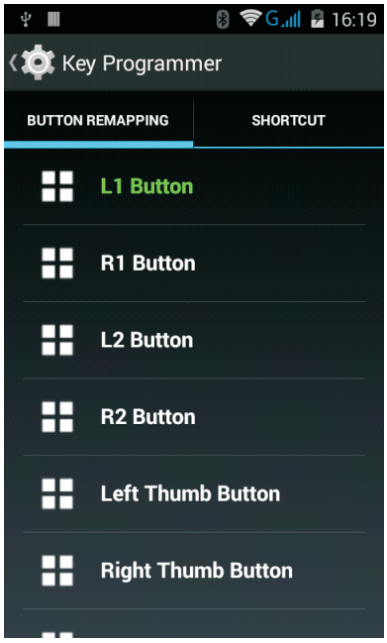
Figure 5-1:



3.Select the button to remap.
4.Touch the BUTTON REMAPPING tab or the SHORTCUT tab that lists the available functions and applications.
5.Touch a function or application shortcut to map to the button.

Note: If you select an application shortcut, the application icon appears next to the button on the Key Programmer screen.

Figure 5-2:



Exporting a Configuration File

The Button Remapping configuration can be exported to an xml file and imported into other MC36 devices.
1.Touch Settings.
2.Touch Key Programmer.
3.Touch virtual Menu button.
4.Touch Export.

The configuration file (key-config.xml) is saved in the folder: /enterprise/usr/.
5.Copy the xml file from the folder to a host computer.

Importing a Configuration File

1.Copy the configuration file (key-config.xml) from a host computer to the folder /enterprise/usr/.
2.Touch Settings.

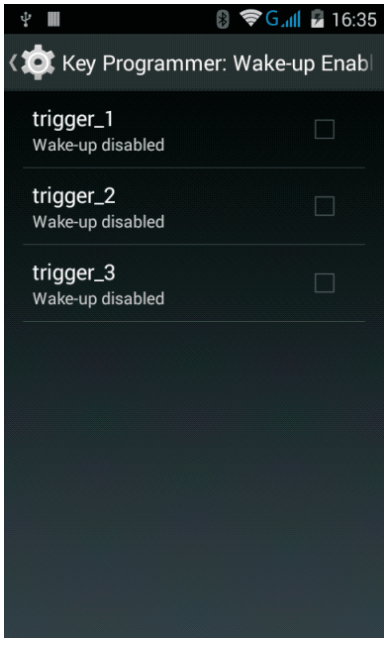
3.Touch Key Programmer.
4.Touch virtual Menu button.
5.Touch Import.

Wakeup Configuration

The three scan keys can be configured as wake-up trigger source:
Major Scan Key- trigger_1.
Left Scan Key – trigger_2.
Right Scan Key – trigger_3.

1.Touch Settings.
2.Touch Key Programmer.
3.Touch virtual Menu button.
4.Touch Wake-up enables.
5.Click the checkbox of the trigger key which is expected to be enabled.

Figure 5-3:



Creating a Remap File

The administrator can create an xml configuration file and import it into any MC36 device. Use any text editor to create the xml file with the filename: key-config.xml

```
<?xml version="1.0" encoding="UTF-8"?><Button_Remap>
 <trigger_1 mode="Remap Button" wakeup="0">
  <REMAP_CODE>BUTTON_L1</REMAP_CODE>
  <EXTRA_SHORTCUT>MPA3_TRIGGER_1</EXTRA_SHORTCUT>
  <EXTRA_TITLE>ScanDemo</EXTRA_TITLE>
  <EXTRA_PACKAGE_NAME>ScanDemo</EXTRA_PACKAGE_NAME>
 </trigger_1>
 <trigger_2 mode="Remap Button" wakeup="0">
  <REMAP_CODE>BUTTON_R1</REMAP_CODE>
  <EXTRA_SHORTCUT>MPA3_TRIGGER_2</EXTRA_SHORTCUT>
  <EXTRA_TITLE/>
  <EXTRA_PACKAGE_NAME/>
 </trigger_2>
 <trigger_3 mode="Remap Button" wakeup="0">
  <REMAP_CODE>BUTTON_L2</REMAP_CODE>
  <EXTRA_SHORTCUT>MPA3_TRIGGER_3</EXTRA_SHORTCUT>
  <EXTRA_TITLE/>
  <EXTRA_PACKAGE_NAME/>
 </trigger_3>
 <trigger_4 mode="Remap Button" wakeup="0">
  <REMAP_CODE>FUNC_P1</REMAP_CODE>
  <EXTRA_SHORTCUT>MPA3_TRIGGER_4</EXTRA_SHORTCUT>
  <EXTRA_TITLE/>
  <EXTRA_PACKAGE_NAME/>
 </trigger_4>
 <trigger_5 mode="Remap Button" wakeup="0">
  <REMAP_CODE>FUNC_P2</REMAP_CODE>
  <EXTRA_SHORTCUT>MPA3_TRIGGER_5</EXTRA_SHORTCUT>
  <EXTRA_TITLE/>
  <EXTRA_PACKAGE_NAME/>
 </trigger_5>
</Button_Remap>
```

Replace the options for each trigger. See Keypad Remap Strings on page 5-5 for a list of available button functions.

Enterprise Reset

To ensure that the configuration persists after an Enterprise Reset:
1.Export the settings before an Enterprise Reset and then import the settings after an Enterprise Reset or
2.Push the configuration file through USB or a third-party MDM to the /enterprise/device/settings/keypad/ folder before the Enterprise Reset. After the Enterprise Reset the key configuration will be automatically applied from this file.

Key Remap Strings

Key Event	Scancode
HOME	102
BACK	158
CALL	231
ENDCALL	107
0	11
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
STAR	227
POUND	228

Key Event	Scancode
DPAD_UP	103
DPAD_DOWN	108
DPAD_LEFT	105
DPAD_RIGHT	106
DPAD_CENTER	353
A	30
B	48
C	46
D	32
E	18
F	33
G	34
H	35
I	23
J	36
K	37
L	38
M	50
N	49
O	24
P	25
Q	16

Key Event	Scancode
R	19
S	31
T	20
U	22
V	47
W	17
X	45
Y	21
Z	44
COMMA	51
PERIOD	52
ALT_LEFT	56
ALT_RIGHT	100
SHIFT_LEFT	42
SHIFT_RIGHT	54
TAB	15
SPACE	57
EXPLORER	150
ENTER	28
DEL	14
GRAVE	41
MINUS	12

Key Event	Scancode
EQUALS	13
LEFT_BRACKET	26
RIGHT_BRACKET	27
BACKSLASH	43
SEMICOLON	39
APOSTROPHE	40
SLASH	53
AT	215
PLUS	78
MENU	139
SEARCH	217
PAGE_UP	177
PAGE_DOWN	178
BUTTON_A	304
BUTTON_B	305
BUTTON_C	306
BUTTON_X	307
BUTTON_Y	308
BUTTON_Z	309
BUTTON_L1	310
BUTTON_R1	311
BUTTON_L2	312

Key Event	Scancode
BUTTON_START	315
BUTTON_SELECT	314
BUTTON_MODE	316
FUNC_P1 (IME Switch)	250
FUNC_P2 (Brightness Bar)	251
BUTTON_R2	313
BUTTON_THUMBL	317
BUTTON_THUMBR	318

Zebra Technologies Corporation, Inc.
3 Overlook Point
LincoInshire, IL 60069, U.S.A.
http://www.zebra.com

MN002340A01 Revision A- October 2015