

**SOFTWARE DECODE SDK
FOR ANDROID
DEVELOPER GUIDE**

SOFTWARE DECODE SDK FOR ANDROID DEVELOPER GUIDE

72E-162670-06

Revision A

November 2016

No part of this publication may be reproduced or used in any form, or by any electrical or mechanical means, without permission in writing from Zebra. This includes electronic or mechanical means, such as photocopying, recording, or information storage and retrieval systems. The material in this manual is subject to change without notice.

The software is provided strictly on an “as is” basis. All software, including firmware, furnished to the user is on a licensed basis. Zebra grants to the user a non-transferable and non-exclusive license to use each software or firmware program delivered hereunder (licensed program). Except as noted below, such license may not be assigned, sublicensed, or otherwise transferred by the user without prior written consent of Zebra. No right to copy a licensed program in whole or in part is granted, except as permitted under copyright law. The user shall not modify, merge, or incorporate any form or portion of a licensed program with other program material, create a derivative work from a licensed program, or use a licensed program in a network without written permission from Zebra. The user agrees to maintain Zebra’s copyright notice on the licensed programs delivered hereunder, and to include the same on any authorized copies it makes, in whole or in part. The user agrees not to decompile, disassemble, decode, or reverse engineer any licensed program delivered to the user or any portion thereof.

Zebra reserves the right to make changes to any software or product to improve reliability, function, or design.

Zebra does not assume any product liability arising out of, or in connection with, the application or use of any product, circuit, or application described herein.

No license is granted, either expressly or by implication, estoppel, or otherwise under any Zebra Technologies Corp. intellectual property rights. An implied license only exists for equipment, circuits, and subsystems contained in Zebra products.

Portions of this software are based in part on the work of the Independent JPEG Group.

Zebra Technologies Corporation
Lincolnshire, IL U.S.A.
<http://www.zebra.com>

Warranty

For the complete hardware product warranty statement, go to:
<http://www.zebra.com/warranty>.

Revision History

Changes to the original manual are listed below:

Change	Date	Description
-01 Rev A	7/2012	Initial release
-02 Rev A	11/2013	<ul style="list-style-type: none"> • Added support for the SE4750 engine • Updated Theory of Operation • Updated Software Installation and Setting Up Build Environment sections • Added Enabling and Disabling All Symbologies, Get Decode Count, and Get Last Decoded Image • Added following parameters: Mobile Phone/Display Mode, Multi Decode Mode, Bar Codes to Read, Full Read Mode, Illumination Power Level, Video Scaling, Retrieve Last Decoded Image, Han Xin, and Han Xin Inverse • Updated Sample Application section • Added Signature Capture Code appendix
-03 Rev A	7/2014	<ul style="list-style-type: none"> • Added support for the SE4710 engine • Updated processors and operating systems supported • Updated Software Decode SDK File Listing • Updated Video Capture section • Added Engine Interface configuration key • Added Fuzzy 1D parameter • Updated following parameters: Image Cropping, Signature Capture Width, Video Scaling • Added Any Length option to Set Lengths for I 2 of 5 • Added I 2 of 5 Security parameter
-04 Rev A	2/2015	<ul style="list-style-type: none"> • Rebranded • Added Ignore Code 128 <FNC4> • Added Quiet Zone parameters
-05 Rev A	3/2016	<ul style="list-style-type: none"> • Updated configuration file location information • Removed Note from Composite CC-C parameter • Added OCR chapter
-06 Rev A	11/2016	<ul style="list-style-type: none"> • Added Notes regarding SE2100 support of parameters

TABLE OF CONTENTS

About This Guide

Introduction	xi
Chapter Descriptions	xi
Notational Conventions	xii
Related Documents	xii
Service Information	xiii

Chapter 1: Getting Started

Introduction	1-1
Theory of Operation	1-2
Supported Symbolologies	1-3
Operating Modes	1-3
System Requirements	1-4
Software Installation	1-5
Software Decode SDK File Listing	1-5
Setting Up the Build Environment	1-10

Chapter 2: Software Decode Library Operation

Introduction	2-1
Initialization	2-1
Scanner Connection and Setup	2-1
Properties and Parameters	2-2
Image Frame Management	2-3
Bar Code Decoding	2-4
Snapshot Capture	2-5
Video Capture	2-5
Enabling all Symbolologies	2-5
Disabling all Symbolologies	2-5
Get Decode Count	2-5
Get Last Decoded Image	2-5
Shutdown	2-5

Chapter 3: Software Decode Library API

Introduction	3-1
Configuration	3-1
sdl.conf Sample File	3-2

Chapter 4: User and Imaging Parameters

Introduction	4-1
User and Imaging Parameter Defaults	4-1
User Preferences	4-3
Trigger Mode	4-3
Picklist Mode	4-3
Decode Session Timeout	4-3
Timeout Between Decodes, Same Symbol	4-4
Transmit Code ID Character	4-4
Transmit “No Read” Message	4-4
Fuzzy 1D Processing	4-4
Mobile Phone/Display Mode	4-5
Multi Decode Mode	4-5
Bar Codes to Read	4-5
Full Read Mode	4-5
Imager Preferences	4-6
Illumination Power Level	4-6
Decoding Illumination	4-6
Decode Aiming Pattern	4-6
Hands-free Decode Aiming Pattern	4-6
Image Capture Illumination	4-7
Motion Illumination	4-7
Snapshot Mode Timeout	4-7
Snapshot Aiming Pattern	4-7
Image Cropping	4-7
Crop to Pixel Addresses	4-8
Image Resolution	4-9
Image File Format Selector	4-9
JPEG Quality Value	4-9
Image Enhancement	4-9
Bits per Pixel	4-10
Signature Capture	4-10
Output File Format	4-10
Signature Capture File Format Selector	4-11
Signature Capture Bits per Pixel	4-11
Signature Capture Width and Height	4-11
Signature Capture Width	4-11
Signature Capture Height	4-11
Signature Capture JPEG Quality	4-12
Video View Finder	4-12
Video Scaling	4-12
Retrieve Last Decoded Image	4-12

Chapter 5: Symbology Parameters

Introduction	5-1
Symbology Parameter Defaults	5-1
Symbologies	5-7
Enable/Disable UPC-A	5-7
Enable/Disable UPC-E	5-7
Enable/Disable UPC-E1	5-7
Enable/Disable EAN-8/JAN-8	5-7
Enable/Disable EAN-13/JAN-13	5-7
Enable/Disable Bookland EAN	5-8
Decode UPC/EAN/JAN Supplementals	5-8
User-Programmable Supplementals	5-9
UPC/EAN/JAN Supplemental Redundancy	5-9
UPC/EAN/JAN Supplemental AIM ID Format	5-9
UPC Reduced Quiet Zone	5-10
Transmit UPC-A Check Digit	5-10
Transmit UPC-E Check Digit	5-10
Transmit UPC-E1 Check Digit	5-10
UPC-A Preamble	5-11
UPC-E Preamble	5-11
UPC-E1 Preamble	5-11
Convert UPC-E to UPC-A	5-11
Convert UPC-E1 to UPC-A	5-12
EAN-8/JAN-8 Extend	5-12
Bookland ISBN Format	5-12
UCC Coupon Extended Code	5-12
Coupon Report	5-13
ISSN EAN	5-13
Enable/Disable Code 128	5-13
Set Lengths for Code 128	5-14
Enable/Disable GS1-128 (formerly UCC/EAN-128)	5-14
Enable/Disable ISBT 128	5-14
ISBT Concatenation	5-15
Check ISBT Table	5-15
ISBT Concatenation Redundancy	5-15
Code 128 Reduced Quiet Zone	5-15
Ignore Code 128 <FNC4>	5-16
Enable/Disable Code 39	5-16
Enable/Disable Trioptic Code 39	5-16
Convert Code 39 to Code 32	5-16
Code 32 Prefix	5-17
Set Lengths for Code 39	5-17
Code 39 Check Digit Verification	5-17
Transmit Code 39 Check Digit	5-17
Code 39 Full ASCII Conversion	5-18
Code 39 Reduced Quiet Zone	5-18
Enable/Disable Code 93	5-18
Set Lengths for Code 93	5-18
Code 11	5-19
Set Lengths for Code 11	5-19
Code 11 Check Digit Verification	5-19

Transmit Code 11 Check Digits	5-19
Enable/Disable Interleaved 2 of 5	5-20
Set Lengths for Interleaved 2 of 5	5-20
1 2 of 5 Check Digit Verification	5-20
Transmit 1 2 of 5 Check Digit	5-21
Convert 1 2 of 5 to EAN-13	5-21
1 2 of 5 Security Level	5-21
1 2 of 5 Reduced Quiet Zone	5-21
Enable/Disable Discrete 2 of 5	5-22
Set Lengths for Discrete 2 of 5	5-22
Enable/Disable Codabar	5-23
Set Lengths for Codabar	5-23
CLSI Editing	5-23
NOTIS Editing	5-23
Enable/Disable MSI	5-24
Set Lengths for MSI	5-24
MSI Check Digits	5-24
Transmit MSI Check Digit(s)	5-25
MSI Check Digit Algorithm	5-25
Enable/Disable Chinese 2 of 5	5-25
Enable/Disable Korean 3 of 5	5-25
Enable/Disable Matrix 2 of 5	5-26
Set Lengths for Matrix 2 of 5	5-26
Matrix 2 of 5 Redundancy	5-26
Matrix 2 of 5 Check Digit	5-26
Transmit Matrix 2 of 5 Check Digit	5-27
Inverse 1D	5-27
US Postnet	5-27
US Planet	5-27
Transmit US Postal Check Digit	5-27
UK Postal	5-28
Transmit UK Postal Check Digit	5-28
Japan Postal	5-28
Australia Post	5-28
Australia Post Format	5-28
Netherlands KIX Code	5-29
USPS 4CB/One Code/Intelligent Mail	5-29
UPU FICS Postal	5-29
GS1 DataBar-14	5-29
GS1 DataBar Limited	5-29
GS1 DataBar Limited Security Level	5-30
GS1 DataBar Expanded	5-30
Convert GS1 DataBar to UPC/EAN	5-30
Composite CC-C	5-31
Composite CC-A/B	5-31
Composite TLC-39	5-31
UPC Composite Mode	5-31
GS1-128 Emulation Mode for UCC/EAN Composite Codes	5-31
Enable/Disable PDF417	5-32
Enable/Disable MicroPDF417	5-32
Code 128 Emulation	5-32

Data Matrix	5-33
Data Matrix Inverse	5-33
Decode Mirror Images (Data Matrix Only)	5-33
Maxicode	5-33
QR Code	5-34
QR Inverse	5-34
MicroQR	5-34
Aztec	5-34
Aztec Inverse	5-34
Han Xin	5-35
Han Xin Inverse	5-35
Redundancy Level	5-35
Security Level	5-36
1D Quiet Zone Level	5-36
Intercharacter Gap Size	5-36

Chapter 6: OCR Programming

Introduction	6-1
OCR Parameter Defaults	6-2
OCR Programming Parameters	6-3
Enable/Disable OCR-A	6-3
OCR-A Variant	6-3
Enable/Disable OCR-B	6-4
OCR-B Variant	6-4
Enable/Disable MICR E13B	6-5
Enable/Disable US Currency Serial Number	6-6
OCR Orientation	6-6
OCR Lines	6-6
OCR Minimum Characters	6-6
OCR Maximum Characters	6-6
OCR Subset	6-7
OCR Quiet Zone	6-7
OCR Template	6-8
Required Digit (9)	6-8
Required Alpha (A)	6-8
Optional Alphanumeric (1)	6-9
Optional Alpha (2)	6-9
Alpha or Digit (3)	6-9
Any Including Space & Reject (4)	6-9
Any except Space & Reject (5)	6-9
Optional Digit (7)	6-9
Digit or Fill (8)	6-9
Alpha or Fill (F)	6-9
Optional Space ()	6-10
Optional Small Special (.)	6-10
Other Template Operators	6-10
Repeat Previous (R)	6-12
Multiple Templates	6-13
Template Examples	6-13
OCR Check Digit Modulus	6-13

OCR Check Digit Multiplier	6-14
OCR Check Digit Validation	6-15
No Check Digit (0)	6-15
Product Add Left to Right (3)	6-15
Product Add Right to Left (1)	6-15
Digit Add Left to Right (4)	6-16
Digit Add Right to Left (2)	6-16
Product Add Right to Left Simple Remainder (5)	6-17
Digit Add Right To Left Simple Remainder (6)	6-17
Health Industry - HIBCC43 (9)	6-17
Inverse OCR	6-17

Appendix A: SDK Demonstration Program

Software Decode SDK Demonstration Programs	A-1
Sample Source Code	A-1
Sample Application	A-2
Sample Application Source Code	A-2

Appendix B: Programming Reference

Symbol Code Identifiers	B-1
AIM Code Identifiers	B-3
SDL Bar Code Type Identifiers	B-8
Decode Data	B-11

Appendix C: Signature Capture Code

Introduction	C-1
Code Structure	C-1
Signature Capture Area	C-1
CapCode Pattern Structure	C-2
Start / Stop Patterns	C-2
Dimensions	C-3
Data Format	C-3
Additional Capabilities	C-4
Signature Boxes	C-4

Index

ABOUT THIS GUIDE

Introduction

The *Software Decode SDK for Android Developer Guide* provides installation and programming information for the Software Decode Software Developer Kit (SDK) for Android operating systems.

Chapter Descriptions

This guide includes the following topics:

- [Chapter 1, Getting Started](#) provides an overview and a theory of operation of the product and its components, and includes information on installing *Software Decode SDK for Android* and a file list for the system library.
- [Chapter 2, Software Decode Library Operation](#) describes how the Software Decode Library interacts with scanning devices and provides operational information.
- [Chapter 3, Software Decode Library API](#) provides a reference to the `BarcodeReader` Java class, and includes configuration information.
- [Chapter 4, User and Imaging Parameters](#) describes the programmable user preference and imaging parameters available via Software Decode.
- [Chapter 5, Symbology Parameters](#) describes the programmable symbology parameters available via Software Decode.
- [Chapter 6, OCR Programming](#) describes how to set up the scanner for OCR programming.
- [Appendix A, SDK Demonstration Program](#) describes a sample application that demonstrates how to interact with the Software Decode Library.
- [Appendix B, Programming Reference](#) provides reference information such as Symbol and AIM code identifiers and SDL bar code type identifiers.
- [Appendix C, Signature Capture Code](#) describes CapCode, a signature capture code that is a special pattern that encloses a signature area on a document and allows a scanner to capture a signature.

Notational Conventions

This document uses the following conventions:

- *Italics* are used to highlight chapters and sections in this and related documents
- bullets (•) indicate:
 - Action items
 - Lists of alternatives
 - Lists of required steps that are not necessarily sequential
- Sequential lists (e.g., those that describe step-by-step procedures) appear as numbered lists.



NOTE This symbol indicates something of special interest or importance to the reader. Failure to read the note will not result in physical harm to the reader, equipment or data.



CAUTION This symbol indicates that if this information is ignored, the possibility of data or material damage may occur.



WARNING! This symbol indicates that if this information is ignored the possibility that serious personal injury may occur.

Related Documents

- *SE4500 Integration Guide*, p/n 72E-112996-xx
- *SE3300 Integration Guide*, p/n 72E-148589-xx
- *SE4750 Integration Guide*, p/n 72E-171726-xx
- *SE4710 Integration Guide*, p/n MN000130Axx
- *SE2100 Integration Guide*, p/n MN-xxxxxx-xx
- Java Class *BarcodeReader*.html document, available with the Software Decode Software Developer Kit (SDK) for Android

For the latest version of this guide and all guides, go to: www.zebra.com/support.

Service Information

If you have a problem using the equipment, contact your facility's technical or systems support. If there is a problem with the equipment, they will contact the Global Customer Support Center at:

www.zebra.com/support.

When contacting support, please have the following information available:

- Serial number of the unit
- Model number or product name
- Software type and version number.

Zebra responds to calls by e-mail, telephone or fax within the time limits set forth in support agreements.

If your problem cannot be solved by support, you may need to return your equipment for servicing and will be given specific directions. Zebra is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty.

If you purchased your business product from a Zebra business partner, contact that business partner for support.

CHAPTER 1 GETTING STARTED

Introduction

The Zebra Software Decode SDK for Android empowers devices to receive and decode images from the imager engine. The Software Decode Library enables host-resident applications to access data decoded from captured images and set system parameters using the Software Decode API.

The imager engine is used with the SDK's software decode library and low-level acquisition drivers to make up a complete bar code decoding solution, including the camera data stream interface and I²C interface for sensor control. Developers can update the sample camera drivers to accommodate their hardware, provided drivers are based on industry standard board support packages.

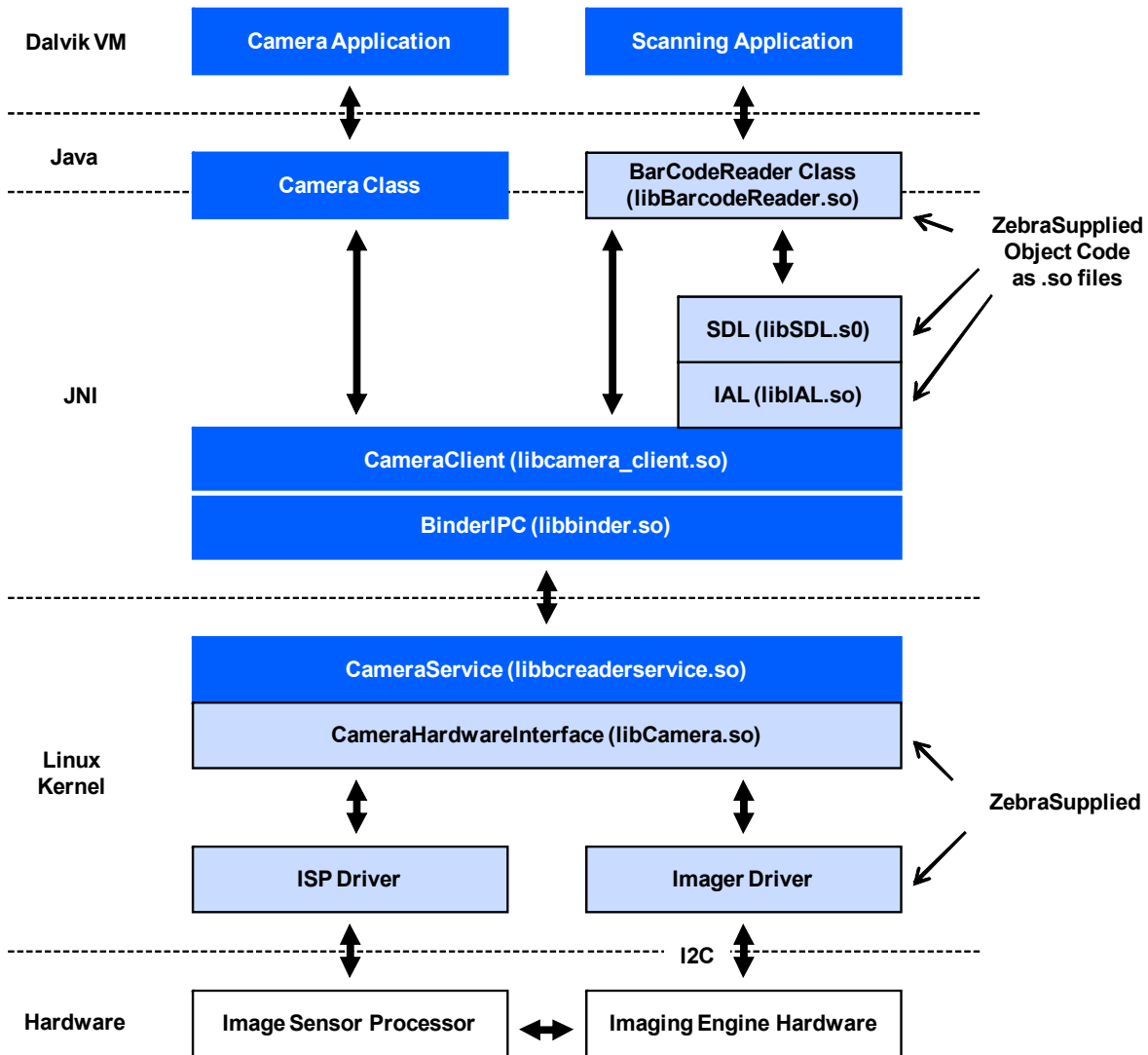


Figure 1-1 Software Decode Solution for Android

This guide describes the installation and configuration of the Software Decode SDK for Android.

Theory of Operation

During image capture:

1. The image sensor in the imager engine captures an image of the bar code through the engine's optical lens. If necessary, the engine automatically adjusts illumination, exposure, and other parameters to obtain the best quality image.
2. The imager engine transmits the image through the V4L2 compliant camera driver, through the hardware abstraction layer, to the JNI layer.
3. The decoding libraries at the JNI layer receive the image and decode any bar code found in the image.
4. The bar code data is sent to the Java host application using a callback mechanism.

Supported Symbologies

The following bar code types are supported and can be individually enabled or disabled:

1D Symbologies

UPC/EAN
Bookland EAN
UCC Coupon Code
ISSN EAN
Code 128
GS1-128
ISBT 128
Code 39
Trioptic Code 39
Code 32
Code 93
Code 11
Interleaved 2 of 5
Discrete 2 of 5
Codabar
MSI
Chinese 2 of 5
Korean 3 of 5
Matrix 2 of 5
Inverse 1D
GS1 DataBar
Composite Codes

2D Symbologies

PDF417
MicroPDF417
Data Matrix
Data Matrix Inverse
Maxicode
QR Code
MicroQR
QR Inverse
Aztec
Aztec Inverse
Han Xin
Han Xin Inverse

Postal Codes

US Postnet
US Planet
UK Postal
Japan Postal
Australian Postal
Netherlands KIX Code
USPS 4CB/One Code/Intelligent Mail
UPU FICS Postal

Operating Modes

The Software Decode SDK supports the following operating modes.

- Decode (default mode) - for decoding a bar code
- Snapshot - for capturing an image
- Video - provides a video of the subject

System Requirements

The SDK supports the following:

- Processor: TI OMAP3, OMAP4, Samsung SOCs, Marvell SOCs, Qualcomm SOCs, MTK SOCs, Freescale SOCs
- Engines:
 - SE4500 imager engine (refer to the *SE4500 Integration Guide*, p/n 72E-112996-xx)
 - SE3300 imager engine (refer to the *SE3300 Integration Guide*, p/n 72E-148589-xx)
 - SE4750 imager engine (refer to the *SE4750 Integration Guide*, p/n 72E-171726-xx)
 - SE4710 imager engine (refer to the *SE4710 Integration Guide*, p/n MN000130Axx)
- Operating Systems: Android versions 2.3, 4.0, 4.1, 4.2, 4.3, 4.4
- RAM (decoding system): 3 MB RAM

Software Installation

The Software Decode SDK for Android is supplied as a tarball file. This section provides information on installing the SDK as well as the contents of the tarball.

Software Decode SDK File Listing

[Table 1-1](#) lists and defines the files within each directory.

Table 1-1 Software Decode SDK Files

Directory	Filename	Description
Android_Sources/Texas Instruments/Ginger Bread (2.3x)		This folder has a structure and layout identical to the Android GB Source root folder
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/kernel/arch/arm/configs	omap3_evm_android_defconfig	Add kernel menu setting for SE4500/SE3300 engine
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/kernel/arch/arm/mach-omap2	omap3_evm_defconfig	Add kernel menu setting for SE4500/SE3300 engine
	board-omap3evm.c	Changed I ² C bus speed to communicate to SE4500/SE3300 to 100KHz
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/kernel/drivers/media/video	board-omap3evm-camera.c	Add support for SE4500/SE3300 engine
	KConfig	Add menu option for SE4500/SE3300 engine
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/kernel/drivers/media/video/isp	Makefile	Support to build the driver file for SE4500/SE3300 engine
	se4500.c	Driver support for SE4500/SE3300 engine
		Support for the SE4500/SE3300 engine
Android_Sources//Texas InstrumentsGinger Bread (2.3x)/kernel/include/media	se4500.h	Header file to support the SE4500/SE3300 driver
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/kernel/Drivers/system/core/rootdir	ueventd.rc	Add device support for SE4500/SE3300 engine
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/OMAP3/device/ti/omap3evm	init.rc	Changes to support debugging over ethernet
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/OMAP3/ frameworks/base/ include/camera	CameraParameters.h	Support for grayscale picture format

Table 1-1 Software Decode SDK Files (Continued)

Directory	Filename	Description
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/ OMAP3/ frameworks/base/libs/ camera	CameraParameters.cpp	Support for grayscale picture format
Android_Sources/Texas Instruments/Ginger Bread (2.3x)/OMAP3/ hardware/ti/omap3/ camera		Add support inside the camera HAL driver for SE4500/SE3300 engine
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)		This folder has a structure and layout identical to the Android ICS Source root folder
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/kernel/ arch/arm/ mach-omap2	Kconfig	Add support for pre-allocating memory for OMAP4 video buffers
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/kernel/arch/ arm/plat-omap/ include/plat	Omap44xx.h	Add OMAP4 support for SE4500
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/kernel/arch/arm/ mach-omap2/include/mach	Omap4-cam.h	Add omap4 header file. TI doesn't include OMAP4 files by default in the Android source.
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/kernel/drivers/media/video	Omap4xiss folder	Add the omap4xiss folder to add OMAP4 support. It includes files for ISP support and OMAP4 registers.
	Soc_camera.c	Add support for SE4500
	SE4500.c	Add the camera driver file for SE4500
	Makefile	Add support for compiling SE4500 and OMAP4
	Kconfig	Add support for compiling SE4500 and OMAP4
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/kernel/include/media/video	V4l2-subdev.h	Add support for SE4500
	Soc_camera.h	Add support for SE4500
	Omap4_camera.h	Add this file to support OMAP4
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/OMAP4/mydroid/frameworks/ base/include/camera	CameraParameters.h	Support for grayscale picture format

Table 1-1 Software Decode SDK Files (Continued)

Directory	Filename	Description
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/OMAP4/mydroid/frameworks/base/libs/camera	CameraParameters.cpp	Support for grayscale picture format
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/OMAP4/mydroid/hardware/ti/omap3/camera		Add support inside the Camera HAL driver for SE4500 engine
Android_Sources/Texas Instruments/Ice Cream Sandwich (4.0.x)/OMAP4/mydroid/system/core/rootdir	ueventd.rc	Add support to expose I ² C device to the application
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)		This folder has a structure and layout identical to the Android JB source root folder
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/kernel/android-3.0/arch/arm/mach-omap2	Kconfig	Add support for pre-allocating memory for OMAP4 video buffers
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/kernel/android-3.0/arch/arm/mach-omap2/include/mach	Omap4-cam.h	Add omap4 header file. TI doesn't include OMAP4 files by default in the Android source.
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/kernel/arch/arm/plat-omap/include/plat	Omap44xx.h	Add OMAP4 support for SE4500/SE4750/SE4710
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/kernel/android-3.0/drivers/media/video	Omap4xiss folder	Add the omap4xiss folder to add OMAP4 support. It includes files for ISP support and OMAP4 registers.
	Soc_camera.c	Add support for SE4500/SE4750/SE4710
	SE4500.c	Add the camera driver file for SE4500/SE4750/SE4710
	Makefile	Add support for compiling SE4500/SE4750/SE4710 and OMAP4
	Kconfig	Add support for compiling SE4500/SE4750/SE4710 and OMAP4
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/kernel/android-3.0/include/media/video	V4l2-subdev.h	Add support for SE4500/SE4750/SE4710
	Soc_camera.h	Add support for SE4500/SE4750/SE4710
	Omap4_camera.h	Add this file to support OMAP4

Table 1-1 Software Decode SDK Files (Continued)

Directory	Filename	Description
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/OMAP4/mydroid/frameworks/av/camera	CameraParameters.cpp	Support for grayscale picture format
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/OMAP4/mydroid/frameworks/av/ include/camera	CameraParameters.h	Support for grayscale picture format
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/ OMAP4/mydroid/hardware/ti/omap4xxx/camera		Add support inside the camera HAL driver for SE4500 engine
Android_Sources/Texas Instruments/Jelly Bean (4.1.x 4.2.x)/OMAP4/mydroid/system/core/rootdir	ueventd.rc	Add support to expose I ² C device to the application
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x)		Source code for Qualcomm based Android JB ad KK
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x)/kernel/arch/arm/mach-msm	board-8960-camera.c	Platform specific board file changes for Qualcomm 8960. Add support for I ² C and SE4750/SE4710. SURF board used as reference.
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x)/kernel//arch/arm/mach-msm	board-8960-regulator.c	Platform specific board file for Qualcomm 8960. Add support for I ² C and SE4750/SE4710. SURF board used as reference.
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x)/kernel/drivers/media/video/msm	Kconfig	Add support for SE4750/SE4710
	Msm_mctl.c	Add support for grey scale
	Sensors folder	Camera driver and Makefile.
	Server folder	Add support to bypass pixel check
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x)/kernel/include/linux	v4l2-mediabus.h	Add support for greyscale
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x) hardware/qcom/camera/QCamera/HAL/core/src	CameraHWI.cpp, QCameraHWI_Parm.cpp, QCameraHWI_Rdi.cpp, QCameraStream.cpp	Necessary changes to support SE4750/SE4710 using RDI path and vision mode
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x)/hardware/qcom/camera/QCamera/stack/mm-camera-test	inc and src folder	Support for scanner resolution and changing the default to test case 3 for running the RDI test app

Table 1-1 Software Decode SDK Files (Continued)

Directory	Filename	Description
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x)/vendor/qcom/proprietary/mm-camera/server	Core, hardware and plugin foders	Adds support for the user space driver for the camera as well as necessary changes for supporting SE4750/SE4710
Qualcomm/JellyBean Kitkat (4.3.x 4.4.x)/system/core/rootdir	ueventd.rc	Add support to expose I ² C device to the application
Applications/SDL_GUI		Sample application files
Documentation	EULA_SDK Zebra	License agreement
	SWDecodeAndroid_DevGuide.pdf	Developer Guide
Documentation/BarCodeReader.javadoc		Help files for the BarCodeReader Java class
Libraries	libbarcodereader.so	Shared library for barcodereader class
	libbarcodereader43.so	Shared library for barcodereader class on Android 4.3
	libbarcodereader44.so	Shared library for barcodereader class on Android 4.4
	libSDL.so	Shared library for Software Decode
	libIAL.so	Shared library for Imager Access
Sources/com/zebra/adc/decoder	BarCodeReader.java	Source code for the BarCodeReader class

Setting Up the Build Environment

Refer to the following developer guide to set up the build environment for TI OMAP3 and OMAP4 with Ubuntu:

- Ginger Bread 2.3.4
Platform: OMAP3
http://processors.wiki.ti.com/index.php/TI-Android-GingerBread-2.3.4-DevKit-2.1_DeveloperGuide#Using_gitorious.org.2Frowboat
- Ice Cream Sandwich 4.0.x
Platform: OMAP4
http://www.omappedia.com/wiki/4AI.1.7_OMAP4_Icecream_Sandwich_Release_Notes#Preparing_Android_binaries
- Jelly Bean 4.1.x and 4.2.x
Platform: OMAP4
http://www.omappedia.com/wiki/4AJ.1.1_OMAP4_Jelly_Bean_Release_Notes

Use the following links to download Qualcomm Android sources and for information on compilation and setup:

- <https://www.codeaurora.org/xwiki/bin/QAEP/release>
- <https://www.codeaurora.org/xwiki/bin/QAEP/>

CHAPTER 2 SOFTWARE DECODE LIBRARY OPERATION

Introduction

This chapter describes how the Software Decode Library interacts with scanning devices and provides operational information to Android Java applications.

Initialization

The user application initializes the Software Decode Library. The Software Decode Library is accessed via the **BarcodeReader** class (see the Java Class **BarcodeReader** .html document) which interfaces with the imager. The user Java class (e.g., the Android Activity class) must implement **BarcodeReader.DecodeCallback**, **BarcodeReader.PictureCallback**, and **BarcodeReader.PreviewCallback** and load the required libraries using:

```
static
{
    System.loadLibrary("IAL");
    System.loadLibrary("SDL");
    System.loadLibrary("barcodereader");
}
```

Scanner Connection and Setup

When an application accesses the Software Decode Library to scan a bar code or perform other scanner-specific operations, it first calls **BarcodeReader.open()** which attempts to locate a scanner and connect to it. If **BarcodeReader.open()** returns success, the connection is established between the application and the scanner, and the application can request various configuration and image processing functions.

Properties and Parameters

To provide maximum flexibility and compatibility with various imaging devices, the Software Decode Library and the application must be able to determine the properties and configuration settings associated with a specific scanner device. Since some hardware maintains its own property and configuration information while others do not, the various levels of software may extend or emulate pieces of this information as it progresses through the call stack.

This SDK defines two types of information about the Software Decode Library and the scanner objects: properties and parameters. Properties are a fixed list of immutable data objects that identify aspects and capabilities of the system or a specific scanner object. Parameters are configurable data objects that can alter the operation of the device.

The **BarcodeReader.PropertyNum** class lists the available scanner-specific properties. This list defines the properties available from each scanner object and is mainly used for identification or for altering the software operation to appropriately manage the device, such as model name and serial number. Properties can be of various data types including signed and unsigned integers and character strings. The application uses **BarcodeReader.getNumProperty()** to access numeric scanner properties and **BarcodeReader.getStrProperty()** to access scanner string properties.

A property can be read after successfully calling **BarcodeReader.open()** and connecting to the scanner. Properties are read-only and therefore cannot be changed.

To control and monitor a scanner device, access device-specific parameters with calls to **BarcodeReader.getNumParameter()**, **BarcodeReader.getStrParameter()**, **BarcodeReader.setParameter()**, and **BarcodeReader.setDefaultParameters()**.

The parameter numbering scheme is unique to each device, but must fit within a 32-bit field. Although all Zebra scan engines use a common list of parameter numbers, parameters supported are unique to each device. Since property IDs are common to all implemented scanner devices, they can be used to identify the specific scanner type and therefore aid in interpreting the parameter IDs used.

For the interfaces defined in this SDK, the parameter number is used while the SSI format for extended (larger than 8-bit) number identification is not.

The Software Decode Library implements a set of parameters that alters decoding and image manipulation processing in various ways. This subset of parameters the scanner provides is defined in the **BarcodeReader.ParamNum** class.

The **BarcodeReader.Parameters** class parallels the Android **Camera.Parameters** class. **BarcodeReader.getParameters()** and **BarcodeReader.setParameters()** configure the camera service. These parameters relate to camera settings and do not affect bar code decoder settings.

BarcodeReader.getNumParameter() and **BarcodeReader.getStrParameter()** are used to get specific numeric and string decoder parameters, respectively. **BarcodeReader.setParameter()** with the correct argument is used to set either the numeric or string decoder parameter. The **BarcodeReader.ParamNum** class contains constants for all decoder parameters. The **BarcodeReader.ParamVal** class contains constants for parameters that do not have simple boolean or range values.

To access parameters for a scanner device, first establish a connection to the device with a call to **BarcodeReader.open()**.

Image Frame Management

When a snapshot image frame is available, the user's **onPictureTaken()** method is called. When video frames are received, the user's **onVideoFrame()** method is called. When preview frames are received, the user's **onPreviewFrame()** method is called.

To pass image frames efficiently through the various layers of software, a set of buffers is allocated to hold the raw image data. Additionally, a pool of frame descriptors is allocated to hold meta data for each frame.

The low level camera interface in the Software Decode Library allocates the frame buffer and frame descriptor memory when **BarcodeReader.open()** is called. The application is not directly involved in memory allocation for raw frames. The low level camera interface performs this function based on its knowledge of the needs of the Software Decode Library.

Once the frame buffers and descriptors are allocated, the Software Decode Library and application can begin processing frames as necessary by calling BarcodeReader **takePicture()** for snapshot or **startVideoCapture()** for video which enable low level acquisition and delivery of frames.

Once the Software Decode Library has processed a frame that satisfies the current operating parameters, it reformats the image data according to the parameter settings. The Software Decode Library then issues the **onPictureTaken()** callback for snapshot mode or the **onVideoFrame()** callback for video mode with the frame data.

BarcodeReader **startPreview()** bypasses the Software Decode Library's frame processing and passes the raw frames to the user's application via the **onPreviewFrame()** callback. The Software Decode Library continues to process incoming frames while the application has not returned from the callback. During this time, the additional frames are discarded by the low level camera interface and are not available to the application.

Bar Code Decoding

The application must implement the **BarcodeReader.DecodeCallback()** and call the **setDecodeCallback()** method to register for decode data callbacks.

Various parameters alter the operation of a decode session.

The **BarcodeReader.ParamNum.LASER_ON_PRIM** parameter sets a timeout value for the session from 0.5 to 9.9 seconds.

The **BarcodeReader.ParamNum.PRIM_TRIG_MODE** controls when and how to initiate a decode session and how to process images. It offers the following options:

- **Level mode** - initiates a decode session when the application calls **BarcodeReader.startDecode()**. When the session starts, the decode session timer is set with the timeout value specified in the **BarcodeReader.ParamNum.LASER_ON_PRIM** parameter and the Software Decode Library starts a video session from the scanner and begins processing frames through its decoding algorithms. The decode session terminates when an image decodes, the session timer expires, or the application calls **BarcodeReader.stopDecode()**. In each case the **onDecodeComplete()** method is called with the length parameter set to the decoded bar code length, **BarcodeReader.DECODE_STATUS_TIMEOUT** or **BarcodeReader.DECODE_STATUS_CANCELED** respectively.
- **Presentation mode** - an application call to **BarcodeReader.startHandsFreeDecode(int mode)** with the mode parameter set to **BarcodeReader.ParamVal.HANDSFREE** initiates a video session from the low level camera interface and monitors the video stream for the presence of an object using motion detection algorithms. When motion is detected, the **DecodeCallback.onEvent** function is called with an event argument of **BCRDR_EVENT_MOTION_DETECTED**. The video images are then processed through the decoding algorithms. On a successful decode, the Software Decode Library issues a call to the **DecodeCallback.onDecodeComplete()** method and returns to motion detection mode.
- **Auto-aim mode** - an application call to **BarcodeReader.startHandsFreeDecode(int mode)** with the mode parameter set to **BarcodeReader.ParamVal.AUTO_AIM** initiates a video session from the low level camera interface and monitors the video stream for the presence of an object using motion detection algorithms. When motion is detected, the aim pattern is enabled, the decode session timer is set with the timeout value specified in the **BarcodeReader.ParamNum.LASER_ON_PRIM** parameter, and the **DecodeCallback.onEvent** function is called with an event argument of **BCRDR_EVENT_MOTION_DETECTED**. When the bar code is within the aiming pattern, the application issues a call to **BarcodeReader.startDecode()** and the Software Decode Library processes video images through its decode algorithms. If a successful decode occurs before the timer expires, the Software Decode Library issues a call to the **DecodeCallback.onDecodeComplete()** method and returns to motion detection mode. If the decode session timer expires, the Software Decode Library reverts to motion detection mode and issues a call to the **DecodeCallback.onDecodeComplete()** method function with the timeout status code **BarcodeReader.DECODE_STATUS_TIMEOUT** set in the length parameter.

Snapshot Capture

The Software Decode Library can also operate in snapshot mode where it does not attempt to decode images but simply formats the images based on the current parameter settings and passes them to the application.

The application must implement the **BarcodeReader.PictureCallback()** interface and call **BarcodeReader.takePicture()** to start this mode. The snapshot image is delivered by a callback to the application's **onPictureTaken()** method.

The session terminates when the application's **onPictureTaken()** returns.

To implement a snapshot preview (i.e., video viewfinder), call **BarcodeReader.setPreviewDisplay()** to set the live preview surface and **BarcodeReader.startViewFinder()** to start previewing frames. To take the snapshot call **BarcodeReader.stopPreview()** followed by **BarcodeReader.takePicture()** as described above.

Video Capture

To initiate video capture, the application must implement **BarcodeReader.VideoCallback()** and call **BarcodeReader.startVideoCapture()**. Once the Software Decode Library has processed a frame that satisfies the current operating parameters, it reformats the image data according to the parameter settings then calls the application's **onVideoFrame()** method to deliver the frame data. To terminate a video session the application must call the **BarcodeReader.stopPreview()** method.

Enabling all Symbologies

To enable the decoding of all bar code types, call **BarcodeReader.enableAllCodeTypes()**.

Disabling all Symbologies

To disable the decoding of all bar code types, call **BarcodeReader.disableAllCodeTypes()**.

Get Decode Count

To retrieve the number of bar codes decoded during a multiple bar code decoding session, call **BarcodeReader.getDecodeCount()**.

Get Last Decoded Image

To retrieve the last decoded image, call **BarcodeReader.getLastDecImage()**.

Shutdown

When the application is about to pause or exit it should call **BarcodeReader.release()** to free any system resources.

CHAPTER 3 SOFTWARE DECODE LIBRARY API

Introduction

This chapter demonstrates use of the `BarcodeReader` Java class. `BarcodeReader` is a generic Android camera interface with special additions for the Zebra imager engine. For more information refer to [Documentation/BarcodeReader.javadoc/overview-tree.html](#).

Configuration

The Android Software Decode Library reads system configuration values from the configuration file `/sdl.conf`, which can be stored at `/mnt/sdcard` or `/system/etc` or `/data`, at startup. If the file is not found, system defaults are used. [Table 3-1](#) lists the configuration key names. Also see [sdl.conf Sample File](#).

Table 3-1 Configuration Keys

Key	Type	Default	Value Range	Description
DebugMode	byte	0	0 = Off 1 = On, Serial 2 = On, File	Off Debug output is sent to the serial port Debug output is sent to the <DebugFile>
DebugLevel	byte	0	0 - 255	0 = off, 255 = most detail
DebugFile	string	(none)		Debug filename, used when DebugMode is set to 2
CamDevice	string	/dev/video0		Name of the camera device to acquire frames
I2CDevice	string	/dev/sdl_control		Name of the device used for I ² C communications
EngineInterface	byte	0x04	0x03 Select MIPI interface on the engine 0x04 Select parallel interface on the engine	Selects the camera interface on the engine. Applicable only for SE4710.

sdl.conf Sample File

```

#-----
# Linux/Android SDL Property/Config file
#
# format:
# <property name> = <hex>
# <property name> = <decimal>
# <property name> = <string>
#
# where:
# <hex> = 0x64
# <decimal> = 100
# <string> = "string"
#
#-----

# decoding

# system
PlatformName = "Android"
EngineModel = 0

##DecodeThreadPriority = 100
##SystemThreadPriority = 100

IALPriority = 0

# devices
CamDevName = "/dev/video0"
I2CDevName = "/dev/sdl_control"

# debug
DebugMode = 2
DebugLevel = 50
DebugFile = "mnt/sdcard/sdl.log"

#----- [end]

```

CHAPTER 4 USER AND IMAGING PARAMETERS

Introduction

This chapter describes the programmable user preference and imaging parameters available via the *Software Decode SDK for Android*, and includes the parameter numbers and option values to use.

- ✓ **NOTE** Parameter bar codes found in other guides can not be used to program these parameters.
- ✓ **NOTE** Some of the parameters in this chapter accept values which are non-contiguous. Entering unsupported values can cause unpredictable behavior.

User and Imaging Parameter Defaults

[Table 4-1](#) lists defaults for all supported user and imaging parameters.

Table 4-1 *User Preferences and Imaging Options Default Table*

Parameter	Parameter Number (Decimal)	Default	Page Number
User Preferences			
Trigger Mode	138	Level	4-3
Picklist Mode	402	Disabled Always	4-3
Decode Session Timeout	136	9.9 Sec	4-3
Timeout Between Decodes, Same Symbol	137	0.6 Sec	4-4
Transmit Code ID Character	45	None	4-4
Transmit “No Read” Message	94	Disable	4-4
Fuzzy 1D Processing	514	Enable	4-4
Mobile Phone/Display Mode	716	0	4-5

Table 4-1 User Preferences and Imaging Options Default Table

Parameter	Parameter Number (Decimal)	Default	Page Number
Multi Decode Mode	900	0	4-5
Bar Codes to Read	902	1	4-5
Full Read Mode	901	1	4-5
Imaging Options			
Illumination Power Level	764	10	4-6
Decoding Illumination	298	Enable	4-6
Decode Aiming Pattern	306	Enable	4-6
Hands-free Decode Aiming Pattern	590	Enable	4-6
Image Capture Illumination	361	Enable	4-7
Motion Illumination	762	Enable	4-7
Snapshot Mode Timeout	323	0 (30 seconds)	4-7
Snapshot Aiming Pattern	300	Enable	4-7
Image Cropping	301	Disable	4-7
Crop to Pixel Addresses	315; 316; 317; 318	0 top, 0 left, 479 bottom, 751 right	4-8
Image Resolution	302	Full	4-9
Image File Format Selection	304	JPEG	4-9
JPEG Quality Value	305	65	4-9
Image Enhancement	564	Off	4-9
Bits per Pixel (BPP)	303	8 BPP	4-10
Signature Capture	93	Disable	4-10
Signature Capture Image File Format Selection	313	JPEG	4-11
Signature Capture Bits per Pixel (BPP)	314	8 BPP	4-11
Signature Capture Width	366	400	4-11
Signature Capture Height	367	100	4-11
Signature Capture JPEG Quality	421	65	4-12
Video View Finder	324	Disable	4-12
Video Scaling	761	2	4-12
Retrieve Last Decoded Image	905	Disable	4-12

User Preferences

In this section, * indicates the default option.

Trigger Mode

Parameter # 138

Select a trigger mode:

- ***0 - Level** - A trigger event activates decode processing, which continues until the trigger event ends, a valid decode, or the decode session time-out occurs.
- **7 - Presentation Mode** - When the imager engine detects an object in its field of view, it triggers and attempts to decode. The range of object detection does not vary under normal lighting conditions. This applies to decode mode only.
- **9 - Auto Aim** - This trigger mode turns on the red laser aiming pattern when the imager engine senses motion. A trigger pull activates decode processing. After 2 seconds of inactivity the red laser aiming pattern automatically shuts off.

Picklist Mode

Parameter # 402

Picklist mode enables the decoder to decode only bar codes aligned under the center of the laser aiming pattern. Select one of the following picklist modes:

- ***0 - Disabled Always** - Picklist mode is always disabled.
- **2 - Enabled Always** - Picklist mode is always enabled.



NOTES Picklist mode works via an approximation of the aiming pattern center. In most cases this approximation is fully accurate. However, decodes can occur when the target bar code is near but not directly under the center of the aiming pattern.

The Zebra SE2100 scan engine does not produce an aiming pattern. For this engine, Picklist Mode works via an approximation of the center of the field of view and therefore works best if the target bar code is placed in the center of the field of view.

Decode Session Timeout

Parameter # 136

Set the maximum time decode processing continues during a scan attempt, available in 0.1 second increments from 0.5 to 9.9 seconds. The default timeout is 9.9 seconds.

For example, to set a decode session timeout of 0.5 seconds, set this parameter to a value of **5**. To set a timeout of 2.5 seconds, enter the value **25**.

Timeout Between Decodes, Same Symbol

Parameter # 137

Use this option in presentation mode to prevent multiple reads of a symbol left in the imager engine's field of view. The timeout begins when you remove the symbol from the field of view.

Set the timeout between decodes for the same symbol, available in 0.1 second increments from 0.0 to 9.9 seconds. The default interval is 0.6 seconds. For example, to set this timeout to 0.5 seconds, enter a value of **5**. To set a timeout of 2.5 seconds, enter the value **25**.

Transmit Code ID Character

Parameter # 45

A Code ID character identifies the code type of a scanned bar code. This is useful when decoding more than one code type. In addition to any single character prefix already selected, the Code ID character is inserted between the prefix and the decoded symbol.

✓ **NOTE** If you enable Symbol Code ID Character or AIM Code ID Character, and enable [Transmit "No Read" Message](#), the decoder appends the code ID for Code 39 to the NR message.

Select one of the following Code ID options:

- ***0 - None**
- **1 - AIM Code ID Character**
- **2 - Symbol Code ID Character**

Transmit "No Read" Message

Parameter # 94

Select whether or not to transmit a No Read message.

✓ **NOTE** If you enable **Transmit No Read**, and also enable Symbol Code ID Character or AIM Code ID Character for [Transmit Code ID Character](#), the decoder appends the code ID for Code 39 to the NR message.

Select one of the following options:

- ***0 - Disable No Read** - the decoder sends nothing to the host if a symbol does not decode.
- **1 - Enable No Read** - the decoder sends the characters **NR** when a successful decode does not occur before trigger release or the **Decode Session Timeout** expires. See [Decode Session Timeout on page 4-3](#).

Fuzzy 1D Processing

Parameter # 514

This option is enabled by default to optimize decode performance on 1D bar codes, including damaged and poor quality symbols. Disable this only if you experience time delays when decoding 2D bar codes, or in detecting a no decode

- **0 - Disable Fuzzy 1D Processing**
- ***1 - Enable Fuzzy 1D Processing**

Mobile Phone/Display Mode

Parameter # 716

✓ **NOTE** The SE2100 does not support Mobile Phone/Display Mode.

This mode improves bar code reading performance on mobile phones and electronic displays. Enabling this mode improves accuracy by reducing the probability of no-decodes or mis-decodes, but may increase decode time.

Select one of the following options:

- ***0 - Disable Mobile Phone/Display Mode**
- **1 - Enable Mobile Phone/Display Mode**

Multi Decode Mode

Parameter # 900

This mode enables decoding multiple bar codes within the scanner's field of view. Select one of the following options:

- ***0 - Disable Multi Decode Mode**
- **1 - Enable Multi Decode Mode**

Bar Codes to Read

Parameter # 902

This parameter sets the number of bar codes to read when **Multi Decode Mode** is enabled. The range is 1 to 10 bar codes. The default is 1.

Full Read Mode

Parameter # 901

Select when to generate a decode event to the calling application when **Multi Decode Mode** is enabled.

- **0** - Generate a decode event after one or more bar codes are decoded.
- ***1** - Only generate the callback to **onDecodeComplete()** when at least the number of bar codes set in **Bar Codes to Read** are decoded.

Imager Preferences

In this section, * indicate the default option.

Illumination Power Level

Parameter # 764

✓ **NOTE** The SE2100 does not support Illumination Power Level.

This parameter sets the level of illumination by altering laser/LED power. The default is 10, which is maximum illumination. For values from 0 to 10, illumination varies from lowest to highest level. This parameter affects both decoding and motion illumination.

Decoding Illumination

Parameter # 298

Enable or disable illumination:

- ***1 - Enable Decoding Illumination** - the decoder turns on illumination every image capture to aid decoding.
- **0 - Disable Decoding Illumination** - the decoder does not use decoding illumination.

Enabling illumination usually results in superior images. The effectiveness of illumination decreases as the distance to the target increases.

Decode Aiming Pattern

Parameter # 306

✓ **NOTE** The SE2100 does not support Decode Aiming Pattern.

This parameter only applies in Decode Mode.

- ***1 - Enable Decode Aiming Pattern** - this projects the aiming pattern during bar code capture.
- **0 - Disable Decode Aiming Pattern** - this turns off the aiming pattern.

Hands-free Decode Aiming Pattern

Parameter # 590

✓ **NOTE** The SE2100 does not support Hands-free Decode Aiming Pattern.

This parameter only applies in hands-free mode.

- ***1 - Enable Hands-free Decode Aiming Pattern** - this projects the aiming pattern during hands-free mode.
- **0 - Disable Hands-free Decode Aiming Pattern** - this turns off the aiming pattern during hands-free mode.

Image Capture Illumination

Parameter # 361

Enable or disable image capture illumination:

- ***1 - Enable Image Capture Illumination** - the decoder turns on illumination during every image capture.
- **0 - Disable Image Capture Illumination** - prevents the decoder from using image capture illumination.

Enabling illumination usually results in superior images. The effectiveness of illumination decreases as the distance to the target increases.

Motion Illumination

Parameter # 762

This parameter only applies to hands-free and auto aim trigger modes.

- ***1 - Enable Motion Illumination** - turns on motion illumination in hands-free and auto aim trigger modes.
- **0 - Disable Motion Illumination** - turns off motion illumination.

Snapshot Mode Timeout

Parameter # 323

This parameter sets the amount of time the decoder remains in Snapshot Mode. The decoder exits Snapshot Mode upon a trigger event, or when the Snapshot Mode Timeout elapses. The default value is 0 which represents 30 seconds; values increment by 30. For example, 1 = 60 seconds, 2 = 90 seconds, etc.

Snapshot Aiming Pattern

Parameter # 300

✓ **NOTE** The SE2100 does not support Snapshot Aiming Pattern.

Enable or disable the Snapshot Aiming Pattern:

- ***1 - Enable Snapshot Aiming Pattern** - projects the aiming pattern when in Snapshot Mode.
- **0 - Disable Snapshot Aiming Pattern** - turns the aiming pattern off.

Image Cropping

Parameter # 301

Enable or disable the Image Cropping:

- ***0 - Disable Image Cropping** - presents the full 752 x 480 pixels for the SE3300/SE4500, 1280 x 960 for the SE4750, 1280 x 800 for the SE4710, and 640 x 480 for the SE2100.
- **1 - Enable Image Cropping** - crops the image to the pixel addresses set in [Crop to Pixel Addresses on page 4-8](#).

✓ **NOTE** The decoder has a cropping resolution of 4 pixels. Setting the cropping area to less than 3 pixels transfers the entire image.

Crop to Pixel Addresses

Parameter # 315 (Top)

Parameter # 316 (Left)

Parameter # 317 (Bottom)

Parameter # 318 (Right)

If you selected **Enable Image Cropping**, set the pixel addresses to crop to. Specify four values for Top, Left, Bottom, and Right, where Top and Bottom correspond to row pixel addresses, and Left and Right correspond to column pixel addresses.

Table 4-2 lists engines with the allowed value ranges and numbering.

Table 4-2 Pixel Address Ranges by Engine

Engine	Value Range	Column Numbering	Row Numbering
SE3300/SE4500	(0,0) to (751, 479)	0 to 751	0 to 479
SE4750	(0,0) to (1279, 959)	0 to 1279	0 to 959
SE4710	(0,0) to (1279, 799)	0 to 1279	0 to 799
SE2100	(0,0) to (639, 479)	0 to 639	0 to 479

For example, for a 4 row x 8 column image in the extreme bottom-right section of the image, set the following values:

SE3300/SE4500: Top = 476, Bottom = 479, Left = 744, Right = 751

SE4750: Top = 955, Bottom = 959, Left = 1271, Right = 1279

SE4710: Top = 795, Bottom = 799, Left = 1271, Right = 1279

SE2100: Top = 475, Bottom = 479, Left = 631, Right = 639

- ✓ **NOTE** The decoder has a minimum cropping resolution of four pixels; increment and decrement cropping addresses in multiples of four. Other values are rounded up. For example, choosing to crop from the top at addresses 0, 1, or 2 (removing 1, 2, or 3 pixels) has the same result as cropping at address 3; this removes four rows from the top.

Image Resolution

Parameter # 302

This option alters image resolution before compression. Rows and columns are removed from the image, resulting in a smaller image containing the original content with reduced resolution.

Select one of the following values:

Table 4-3 *Image Resolution by Engine*

Value	Resolution	Uncropped Image Size			
		SE3300, SE4500	SE4750	SE4710	SE2100
*0	Full	752 x 480	1280 x 960	1280 x 800	640 x 480
1	1/2	376 x 240	640 x 480	640 x 400	320 x 240
3	1/4	188 x 120	320 x 240	320 x 200	160 x 120

Image File Format Selector

Parameter # 304

Select an image format appropriate for the system. The decoder stores captured images in the selected format:

- *1 - JPEG File Format
- 3 - BMP File Format
- 4 - TIFF File Format

JPEG Quality Value

JPEG Quality = Parameter # 305

Set the **JPEG Quality** to a value from 5 to 100, where 100 represents the highest quality image. The default is 65.

Image Enhancement

Parameter # 564

This feature uses a combination of edge sharpening and contrast enhancement to produce an image that is visually pleasing. Select a level of image enhancement:

- *0 - Off
- 1 - Low
- 2 - Medium
- 3 - High

Bits per Pixel

Parameter # 303

Select the number of significant bits per pixel (BPP) to use when capturing an image:

- *2 - 8 BPP to assign 1 of 256 levels of grey to each pixel
- 0 - 1 BPP for a black and white image
- 1 - 4 BPP to assign 1 of 16 levels of grey to each pixel

The decoder ignores these settings for JPEG files, which always use 8 BPP.

Signature Capture

Parameter # 93

✓ **NOTE** The SE2100 does not support Signature Capture.

A signature capture bar code is a special-purpose symbology which delineate a signature capture area in a document with a machine-readable format. The recognition pattern is variable so it can optionally provide an index to various signatures. The region inside the bar code pattern is considered the signature capture area. See [Appendix C, Signature Capture Code](#) for more information.

Output File Format

Decoding a signature capture bar code de-skews the signature image and converts the image to a BMP, JPEG, or TIFF file format. The output data includes the file descriptor followed by the formatted signature image.

Table 4-4 Signature Capture Image Format

File Descriptor			Signature Image
Output Format (1 byte)	Signature Type (1 byte)	Signature Image Size (4 bytes) (BIG Endian)	
JPEG - 1 BMP - 3 TIFF - 4	1-8	0x00000400	0x00010203.....

Enable or disable **Signature Capture**:

- *0 - Disable Signature Capture
- 1 - Enable Signature Capture

Signature Capture File Format Selector

Parameter # 313

Select a signature file format appropriate for the system (BMP, TIFF, or JPEG). The decoder stores captured signatures in the selected format.

- *1 - JPEG Signature Format
- 3 - BMP Signature Format
- 4 - TIFF Signature Format

Signature Capture Bits per Pixel

Parameter # 314

Select the number of significant bits per pixel (BPP) to use when capturing a signature:

- *2 - 8 BPP to assign 1 of 256 levels of grey to each pixel
- 0 - 1 BPP for a black and white image
- 1 - 4 BPP to assign 1 of 16 levels of grey to each pixel

The decoder ignores these settings for JPEG files, which always use 8 BPP.

Signature Capture Width and Height

The aspect ratio of the Signature Capture Width and Signature Capture Height parameters must match that of the signature capture area. For example, a 4 x 1 inch signature capture area requires a 4 to 1 aspect ratio of width to height.

Signature Capture Width

Parameter # 366

Set the width of the signature capture box to a value within the appropriate range below:

- SE3300/SE4500: 001 to 752 decimal
- SE4710/SE4750: 001 to 1280 decimal

The default is 400.

Signature Capture Height

Parameter # 367

Set the height of the signature capture box to a value within the appropriate range below:

- SE3300/SE4500: 001 to 480 decimal
- SE4750: 001 to 960 decimal
- SE4710: 001 to 800 decimal

The default is 100.

Signature Capture JPEG Quality

Parameter # 421

Set the **Signature Capture JPEG Quality** to a value from 005 to 100, where 100 represents the highest quality image. The default is 65.

Video View Finder

Parameter # 324

Select a Video View Finder option:

- ***0 - Disable Video View Finder** - turns the video view finder off.
- **1 - Enable Video View Finder** - projects the video view finder while in Image Mode.

Video Scaling

Parameter # 761

Set the resolution of the image in video mode.

- **0 - Full Resolution:**
 - SE3300/SE4500: 752 x 480
 - SE4750: 1280 x 960
 - SE4710: 1280 x 800
 - SE2100: 640 to 480
- **1 - 1/2 Resolution**
- ***2 - 1/3 Resolution**
- **3 - 1/4 Resolution**

Retrieve Last Decoded Image

Parameter # 905

This parameter retrieves the last decoded frame in the most recent decode session. To receive the last decoded image call **barcodeReader.getLastDecImage ()**. Enable or disable this parameter:

- ***0 - Disable Retrieve Last Decoded Image**
- **1 - Enable Retrieve Last Decoded Image**

CHAPTER 5 SYMBOLOGY PARAMETERS

Introduction

This chapter describes the programmable symbology parameters available via the *Software Decode SDK for Android*, and includes the parameter numbers and option values to use.

Symbology Parameter Defaults

[Table 5-1](#) lists defaults for all supported symbology parameters.

Table 5-1 *Symbology Preferences Default Table*

Parameter	Parameter Number (Decimal)	Default	Page Number
UPC/EAN			
UPC-A	1	Enable	5-7
UPC-E	2	Enable	5-7
UPC-E1	12	Disable	5-7
EAN-8/JAN 8	4	Enable	5-7
EAN-13/JAN 13	3	Enable	5-7
Bookland EAN	83	Disable	5-8
Decode UPC/EAN/JAN Supplementals (2 and 5 digits)	16	Ignore	5-8
User-Programmable Supplementals			5-9
Supplemental 1:	579		
Supplemental 2:	580		
UPC/EAN/JAN Supplemental Redundancy	80	10	5-9
Decode UPC/EAN/JAN Supplemental AIM ID	672	Combined	5-9

Table 5-1 *Symbology Preferences Default Table (Continued)*

Parameter	Parameter Number (Decimal)	Default	Page Number
UPC Reduced Quiet Zone	1289	Disable	5-10
Transmit UPC-A Check Digit	40	Enable	5-10
Transmit UPC-E Check Digit	41	Enable	5-10
Transmit UPC-E1 Check Digit	42	Enable	5-10
UPC-A Preamble	34	System Character	5-11
UPC-E Preamble	35	System Character	5-11
UPC-E1 Preamble	36	System Character	5-11
Convert UPC-E to A	37	Disable	5-11
Convert UPC-E1 to A	38	Disable	5-12
EAN-8/JAN-8 Extend	39	Disable	5-12
Bookland ISBN Format	576	ISBN-10	5-12
UCC Coupon Extended Code	85	Disable	5-12
Coupon Report	730	New Coupon Symbols	5-13
ISSN EAN	617	Disable	5-13
Code 128			
Code 128	8	Enable	5-13
Set Length(s) for Code 128	209 210	Any Length	5-14
GS1-128 (formerly UCC/EAN-128)	14	Enable	5-14
ISBT 128	84	Enable	5-14
ISBT Concatenation	577	Disable	5-15
Check ISBT Table	578	Enable	5-15
ISBT Concatenation Redundancy	223	10	5-15
Code 128 Reduced Quiet Zone	1208	Disable	5-15
Ignore Code 128 <FNC4>	1254	Disable	5-16
Code 39			
Code 39	0	Enable	5-16
Trioptic Code 39	13	Disable	5-16
Convert Code 39 to Code 32 (Italian Pharmacy Code)	86	Disable	5-16
Code 32 Prefix	231	Disable	5-17

Table 5-1 Symbology Preferences Default Table (Continued)

Parameter	Parameter Number (Decimal)	Default	Page Number
Set Length(s) for Code 39	18 19	2 to 55	5-17
Code 39 Check Digit Verification	48	Disable	5-17
Transmit Code 39 Check Digit	43	Disable	5-19
Code 39 Full ASCII Conversion	17	Disable	5-18
Code 39 Reduced Quiet Zone	1209	Disable	5-18
Code 93			
Code 93	9	Disable	5-18
Set Length(s) for Code 93	26 27	4 to 55	5-18
Code 11			
Code 11	10	Disable	5-19
Set Lengths for Code 11	28 29	4 to 55	5-19
Code 11 Check Digit Verification	52	Disable	5-19
Transmit Code 11 Check Digit(s)	47	Disable	5-19
Interleaved 2 of 5 (ITF)			
Interleaved 2 of 5 (ITF)	6	Enable	5-20
Set Lengths for I 2 of 5	22 23	14	5-20
I 2 of 5 Check Digit Verification	49	Disable	5-20
Transmit I 2 of 5 Check Digit	44	Disable	5-21
Convert I 2 of 5 to EAN 13	82	Disable	5-21
I 2 of 5 Security Level	1121	Security Level 1	5-21
I 2 of 5 Reduced Quiet Zone	1210	Disable	5-21
Discrete 2 of 5 (DTF)			
Discrete 2 of 5	5	Disable	5-22
Set Length(s) for D 2 of 5	20 21	12	5-22
Codabar (NW - 7)			
Codabar	7	Disable	5-23
Set Lengths for Codabar	24 25	5 to 55	5-23

Table 5-1 Symbology Preferences Default Table (Continued)

Parameter	Parameter Number (Decimal)	Default	Page Number
CLSI Editing	54	Disable	5-23
NOTIS Editing	55	Disable	5-23
MSI			
MSI	11	Disable	5-24
Set Length(s) for MSI	30 31	4 to 55	5-24
MSI Check Digits	50	One	5-24
Transmit MSI Check Digit	46	Disable	5-25
MSI Check Digit Algorithm	51	Mod 10/Mod 10	5-25
Chinese 2 of 5			
Chinese 2 of 5	408	Disable	5-25
Korean 3 of 5			
Korean 3 of 5	581	Disable	5-25
Matrix 2 of 5			
Matrix 2 of 5	618	Disable	5-26
Matrix 2 of 5 Lengths	619 620	14	5-26
Matrix 2 of 5 Redundancy	621	Disable	5-26
Matrix 2 of 5 Check Digit	622	Disable	5-26
Transmit Matrix 2 of 5 Check Digit	623	Disable	5-27
Inverse 1D	586	Regular	5-27
Postal Codes			
US Postnet	89	Enable	5-27
US Planet	90	Enable	5-27
Transmit US Postal Check Digit	95	Enable	5-27
UK Postal	91	Enable	5-28
Transmit UK Postal Check Digit	96	Enable	5-28
Japan Postal	290	Enable	5-28
Australia Post	291	Enable	5-28
Australia Post Format	718	Autodiscriminate	5-28
Netherlands KIX Code	326	Enable	5-29

Table 5-1 Symbology Preferences Default Table (Continued)

Parameter	Parameter Number (Decimal)	Default	Page Number
USPS 4CB/One Code/Intelligent Mail	592	Disable	5-29
UPU FICS Postal	611	Disable	5-29
GS1 DataBar (formerly RSS, Reduced Space Symbology)			
GS1 DataBar-14	338	Enable	5-29
GS1 DataBar Limited	339	Disable	5-29
GS1 DataBar Limited Security Level	728	3	5-30
GS1 DataBar Expanded	340	Disable	5-30
Convert GS1 DataBar to UPC/EAN	397	Disable	5-30
Composite			
Composite CC-C	341	Disable	5-31
Composite CC-A/B	342	Disable	5-31
Composite TLC-39	371	Disable	5-31
UPC Composite Mode	344	Never Linked	5-31
GS1-128 Emulation Mode for UCC/EAN Composite Codes	427	Disable	5-31
2D Symbologies			
PDF417	15	Enable	5-32
MicroPDF417	227	Disable	5-32
Code 128 Emulation	123	Disable	5-32
Data Matrix	292	Enable	5-33
Data Matrix Inverse	588	Regular	5-33
Decode Mirror Images (Data Matrix Only)	537	Never	5-33
Maxicode	294	Enable	5-33
QR Code	293	Enable	5-34
QR Inverse	587	Regular	5-34
MicroQR	573	Enable	5-34
Aztec	574	Enable	5-34
Aztec Inverse	589	Regular	5-34
Han Xin	1167	Disable	5-35
Han Xin Inverse	1168	Regular	5-35

Table 5-1 *Symbology Preferences Default Table (Continued)*

Parameter	Parameter Number (Decimal)	Default	Page Number
Symbology-Specific Security Levels			
Redundancy Level	78	1	5-35
Security Level	77	1	5-36
1D Quiet Zone Level	1288	1	5-36
Intercharacter Gap Size	381	Normal	5-36

Symbologies

In this section, * indicates the default option.

Enable/Disable UPC-A

Parameter # 1

Enable or disable UPC-A:

- *1 - Enable UPC-A
- 0 - Disable UPC-A

Enable/Disable UPC-E

Parameter # 2

Enable or disable UPC-E:

- *1 - Enable UPC-E
- 0 - Disable UPC-E

Enable/Disable UPC-E1

Parameter # 12

Enable or disable UPC-E1. UPC-E1 is disabled by default.

- *0 - Disable UPC-E1
- 1 - Enable UPC-E1

✓ **NOTE** UPC-E1 is not a UCC (Uniform Code Council) approved symbology.

Enable/Disable EAN-8/JAN-8

Parameter # 4

Enable or disable EAN-8/JAN-8:

- *1 - Enable EAN-8/JAN-8
- 0 - Disable EAN-8/JAN-8

Enable/Disable EAN-13/JAN-13

Parameter # 3

Enable or disable EAN-13/JAN-13:

- *1 - Enable EAN-13/JAN-13
- 0 - Disable EAN-13/JAN-13

Enable/Disable Bookland EAN

Parameter # 83

Enable or disable Bookland EAN:

- ***0 - Disable Bookland EAN**
- **1 - Enable Bookland EAN**

✓ **NOTE** If you enable Bookland EAN, select a [Bookland ISBN Format on page 5-12](#). Also select either Decode UPC/EAN Supplementals, Autodiscriminate UPC/EAN Supplementals, or Enable 978/979 Supplemental Mode in [Decode UPC/EAN/JAN Supplementals on page 5-8](#).

Decode UPC/EAN/JAN Supplementals

Parameter # 16

Supplementals are bar codes appended according to specific format conventions (e.g., UPC A+2, UPC E+2, EAN 13+2). Select one of the following options:

- ***0 - Ignore UPC/EAN with Supplementals** - if the decoder is presented with a UPC/EAN plus supplemental symbol, the decoder decodes UPC/EAN and ignores the supplemental characters.
- **1 - Decode UPC/EAN with Supplementals** - the decoder only decodes UPC/EAN symbols with supplemental characters, and ignores symbols without supplementals.
- **2 - Autodiscriminate UPC/EAN Supplementals** - decoder decodes UPC/EAN symbols with supplemental characters immediately. If the symbol does not have a supplemental, the decoder must decode the bar code the number of times set via [UPC/EAN/JAN Supplemental Redundancy on page 5-9](#) before transmitting its data to confirm that there is no supplemental.

If you select one of the following **Supplemental Mode** options, the decoder immediately transmits EAN-13 bar codes starting with that prefix that have supplemental characters. If the symbol does not have a supplemental, the decoder must decode the bar code the number of times set via [UPC/EAN/JAN Supplemental Redundancy on page 5-9](#) before transmitting its data to confirm that there is no supplemental. The decoder transmits UPC/EAN bar codes that do not have that prefix immediately.

- **4 - Enable 378/379 Supplemental Mode**
- **5 - Enable 978/979 Supplemental Mode.**

✓ **NOTE** If you select 978 Supplemental Mode and are scanning Bookland EAN bar codes, see [Enable/Disable Bookland EAN on page 5-8](#) to enable Bookland EAN, and select a format using [Bookland ISBN Format on page 5-12](#).

- **7 - Enable 977 Supplemental Mode**
- **6 - Enable 414/419/434/439 Supplemental Mode**
- **8 - Enable 491 Supplemental Mode**
- **3 - Enable Smart Supplemental Mode** - applies to EAN-13 bar codes starting with any prefix listed previously.
- **9 - Supplemental User-Programmable Type 1** - applies to EAN-13 bar codes starting with a 3-digit user-defined prefix. Set this 3-digit prefix using [User-Programmable Supplementals on page 5-9](#).
- **10 - Supplemental User-Programmable Type 1 and 2** - applies to EAN-13 bar codes starting with either of two 3-digit user-defined prefixes. Set the 3-digit prefixes using [User-Programmable Supplementals on page 5-9](#).

- **11 - Smart Supplemental Plus User-Programmable 1** - applies to EAN-13 bar codes starting with any prefix listed previously or the user-defined prefix set using [User-Programmable Supplementals on page 5-9](#).
- **12 - Smart Supplemental Plus User-Programmable 1 and 2** - applies to EAN-13 bar codes starting with any prefix listed previously or one of the two user-defined prefixes set using [User-Programmable Supplementals on page 5-9](#).

✓ **NOTE** To minimize the risk of invalid data transmission, select either to decode or ignore supplemental characters.

User-Programmable Supplementals

Supplemental 1: Parameter # 579

Supplemental 2: Parameter # 580

If you selected a Supplemental User-Programmable option from [Decode UPC/EAN/JAN Supplementals](#), select **User-Programmable Supplemental 1** to set the 3-digit prefix. Select **User-Programmable Supplemental 2** to set a second 3-digit prefix.

UPC/EAN/JAN Supplemental Redundancy

Parameter # 80

With **Autodiscriminate UPC/EAN/JAN Supplementals** selected, this option adjusts the number of times a symbol without supplementals is decoded before transmission. The range is from two to 30 times. Five or above is recommended when decoding a mix of UPC/EAN/JAN symbols with and without supplementals, and the autodiscriminate option is selected. The default is set at 10.

UPC/EAN/JAN Supplemental AIM ID Format

Parameter # 672

Select an output format when reporting UPC/EAN/JAN bar codes with Supplementals with [Transmit Code ID Character on page 4-4](#) set to **AIM Code ID Character**:

- **0 - Separate** - transmit UPC/EAN with supplementals with separate AIM IDs but one transmission, i.e.:
]E<0 or 4><data>]E<1 or 2>[supplemental data]
- ***1 - Combined** – transmit UPC/EAN with supplementals with one AIM ID and one transmission, i.e.:
]E3<data+supplemental data>
- **2 - Separate Transmissions** - transmit UPC/EAN with supplementals with separate AIM IDs and separate transmissions, i.e.:
]E<0 or 4><data>
]E<1 or 2>[supplemental data]

UPC Reduced Quiet Zone

Parameter # 1289

Enable or disable decoding UPC bar codes with reduced quiet zones. If you select **Enable**, select a *1D Quiet Zone Level* on page 5-36.

- *0 - Disable UPC Reduced Quiet Zone
- 1 - Enable UPC Reduced Quiet Zone

Transmit UPC-A Check Digit

Parameter # 40

The check digit is the last character of the symbol used to verify the integrity of the data. Select whether to transmit the bar code data with or without the UPC-A check digit. It is always verified to guarantee the integrity of the data.

- *1 - Transmit UPC-A Check Digit
- 0 - Do Not Transmit UPC-A Check Digit

Transmit UPC-E Check Digit

Parameter # 41

The check digit is the last character of the symbol used to verify the integrity of the data. Select whether to transmit the bar code data with or without the UPC-E check digit. It is always verified to guarantee the integrity of the data.

- *1 - Transmit UPC-E Check Digit
- 0 - Do Not Transmit UPC-E Check Digit

Transmit UPC-E1 Check Digit

Parameter # 42

The check digit is the last character of the symbol used to verify the integrity of the data. Select whether to transmit the bar code data with or without the UPC-E1 check digit. It is always verified to guarantee the integrity of the data.

- *1 - Transmit UPC-E1 Check Digit
- 0 - Do Not Transmit UPC-E1 Check Digit

UPC-A Preamble

Parameter # 34

Preamble characters are part of the UPC symbol, and include Country Code and System Character. There are three options for transmitting a UPC-A preamble to the host device. Select the appropriate option to match the host system:

- ***1** - Transmit System Character Only (<SYSTEM CHARACTER> <DATA>)
- **2** - Transmit System Character and Country Code ("0" for USA)
(< COUNTRY CODE> <SYSTEM CHARACTER> <DATA>)
- **0** - Transmit no preamble (<DATA>)

UPC-E Preamble

Parameter # 35

Preamble characters are part of the UPC symbol, and include Country Code and System Character. There are three options for transmitting a UPC-E preamble to the host device. Select the appropriate option to match the host system.

- ***1** - Transmit System Character Only (<SYSTEM CHARACTER> <DATA>)
- **2** - Transmit System Character and Country Code ("0" for USA)
(< COUNTRY CODE> <SYSTEM CHARACTER> <DATA>)
- **0** - Transmit no preamble (<DATA>)

UPC-E1 Preamble

Parameter # 36

Preamble characters are part of the UPC symbol, and include Country Code and System Character. There are three options for transmitting a UPC-E1 preamble to the host device. Select the appropriate option to match the host system.

- ***1** - Transmit System Character Only (<SYSTEM CHARACTER> <DATA>)
- **2** - Transmit System Character and Country Code ("0" for USA)
(< COUNTRY CODE> <SYSTEM CHARACTER> <DATA>)
- **0** - Transmit no preamble (<DATA>)

Convert UPC-E to UPC-A

Parameter # 37

Enable this to convert UPC-E (zero suppressed) decoded data to UPC-A format before transmission. After conversion, the data follows UPC-A format and is affected by UPC-A programming selections (e.g., Preamble, Check Digit). When disabled, UPC-E decoded data is transmitted as UPC-E data, without conversion.

- ***0** - Do Not Convert UPC-E to UPC-A (Disable)
- **1** - Convert UPC-E to UPC-A (Enable)

Convert UPC-E1 to UPC-A

Parameter # 38

Enable this to convert UPC-E1 decoded data to UPC-A format before transmission. After conversion, the data follows UPC-A format and is affected by UPC-A programming selections (e.g., Preamble, Check Digit). When disabled, UPC-E1 decoded data is transmitted as UPC-E1 data, without conversion.

- ***0 - Do Not Convert UPC-E to UPC-A (Disable)**
- **1 - Convert UPC-E to UPC-A (Enable)**

EAN-8/JAN-8 Extend

Parameter # 39

Enable this parameter to add five leading zeros to decoded EAN-8 symbols to make them compatible in format to EAN-13 symbols. Disable this to transmit EAN-8 symbols as is.

- ***0 - Disable EAN/JAN Zero Extend**
- **1 - Enable EAN/JAN Zero Extend**

Bookland ISBN Format

Parameter # 576

If you enabled Bookland EAN using [Enable/Disable Bookland EAN on page 5-8](#), select one of the following formats for Bookland data:

- ***0 - Bookland ISBN-10** - The decoder reports Bookland data starting with 978 in traditional 10-digit format with the special Bookland check digit for backward-compatibility. Data starting with 979 is not considered Bookland in this mode.
- **1 - Bookland ISBN-13** - The decoder reports Bookland data (starting with either 978 or 979) as EAN-13 in 13-digit format to meet the 2007 ISBN-13 protocol.

✓ **NOTE** For Bookland EAN to function properly, first enable Bookland EAN using [Enable/Disable Bookland EAN on page 5-8](#), then select either Decode UPC/EAN Supplementals, Autodiscriminate UPC/EAN Supplementals, or Enable 978/979 Supplemental Mode in [Decode UPC/EAN/JAN Supplementals on page 5-8](#).

UCC Coupon Extended Code

Parameter # 85

Enable this parameter to decode UPC-A bar codes starting with digit '5', EAN-13 bar codes starting with digit '99', and UPC-A/EAN-128 Coupon Codes. UPCA, EAN-13, and EAN-128 must be enabled to scan all types of Coupon Codes.

- ***0 - Disable UCC Coupon Extended Code**
- **1 - Enable UCC Coupon Extended Code**

✓ **NOTE** Use the Decode UPC/EAN Supplemental Redundancy parameter to control autodiscrimination of the EAN128 (right half) of a coupon code.

Coupon Report

Parameter # 730

Traditional coupon symbols (old coupon symbols) are composed of two bar codes: UPC/EAN and Code128. A new coupon symbol is composed of a single Databar Expanded bar code. The new coupon format offers more options for purchase values (up to \$999.99) and supports complex discount offers such as a second purchase requirement.

An interim coupon symbol also exists that contains both types of bar codes: UPC/EAN and Databar Expanded. This format accommodates both retailers that do not recognize or use the additional information included in the new coupon symbol, as well as those who can process new coupon symbols.

Select one of the following options for decoding coupon symbols:

- **0 - Old Coupon Symbols** - Scanning an old coupon symbol reports both UPC and Code 128, scanning an interim coupon symbol reports UPC, and scanning a new coupon symbol reports nothing (no decode).
- ***1 - New Coupon Symbols** - Scanning an old coupon symbol reports either UPC or Code 128, and scanning an interim coupon symbol or a new coupon symbol reports Databar Expanded.
- **2 - Both Coupon Formats** - Scanning an old coupon symbol reports both UPC and Code 128, and scanning an interim coupon symbol or a new coupon symbol reports Databar Expanded.

ISSN EAN

Parameter # 617

Enable or disable ISSN EAN.

- ***0 - Disable ISSN EAN**
- **1 - Enable ISSN EAN**

Enable/Disable Code 128

Parameter # 8

Enable or disable Code 128:

- ***1 - Enable Code 128**
- **0 - Disable Code 128**

Set Lengths for Code 128

Length1 Parameter #209 [Range: 0..55]

Length2 Parameter #210 [Range: 0..55]

Default: Length1=0, Length2=0

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for Code 128 to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of Code 128 to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14, Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of Code 128 to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode Code 128 codes of either 2 or 14 characters only, set **Length1 = 14, Length2 = 2**.
- **Length Within Range** - To decode only Code 128 codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode Code 128 codes of length 4 through 12 characters, set **Length1 = 4, Length2 = 12**.

Enable/Disable GS1-128 (formerly UCC/EAN-128)

Parameter # 14

Enable or disable GS1-128:

- ***1 - Enable GS1-128**
- **0 - Disable GS1-128**

Enable/Disable ISBT 128

Parameter # 84

ISBT 128 is a variant of Code 128 used in the blood bank industry. Enable or disable ISBT 128. If necessary, the host must perform concatenation of the ISBT data.

- ***1 - Enable ISBT 128**
- **0 - Disable ISBT 128**

ISBT Concatenation

Parameter # 577

Select an option for concatenating pairs of ISBT code types:

- ***0 - Disable ISBT Concatenation** - The device does not concatenate pairs of ISBT codes it encounters.
- **1 - Enable ISBT Concatenation** - There must be two ISBT codes in order for the device to decode and perform concatenation. The device does not decode single ISBT symbols.
- **2 - Autodiscriminate ISBT Concatenation** - The device decodes and concatenates pairs of ISBT codes immediately. If only a single ISBT symbol is present, the device must decode the symbol the number of times set via [ISBT Concatenation Redundancy](#) before transmitting its data to confirm that there is no additional ISBT symbol.

Check ISBT Table

Parameter # 578

The ISBT specification includes a table that lists several types of ISBT bar codes that are commonly used in pairs. If you enable **ISBT Concatenation**, enable **Check ISBT Table** to concatenate only those pairs found in this table. Other types of ISBT codes are not concatenated.

- ***1 - Enable Check ISBT Table**
- **0 - Disable Check ISBT Table**

ISBT Concatenation Redundancy

Parameter # 223

With **ISBT Concatenation** set to **Autodiscriminate**, this option sets the number of times the device must decode an ISBT symbol before determining that there is no additional symbol. The range is from two to 20 times. The default is 10.

Code 128 Reduced Quiet Zone

Parameter # 1208

Enable or disable decoding Code 128 bar codes with reduced quiet zones. If you select **Enable**, select a [1D Quiet Zone Level on page 5-36](#).

- ***0 - Disable Code 128 Reduced Quiet Zone**
- **1 - Enable Code 128 Reduced Quiet Zone**

Ignore Code 128 <FNC4>

Parameter # 1254

This feature applies to Code 128 bar codes with an embedded <FNC4> character.

- ***0 - Disable Ignore Code 128 <FNC4>** - The <FNC4> character is not transmitted but the following character has 128 added to it.
- **1 - Enable Ignore Code 128 <FNC4>** - This strips the <FNC4> character from the decode data. The remaining characters do not change.

Enable/Disable Code 39

Parameter # 0

Enable or disable Code 39:

- ***1 - Enable Code 39**
- **0 - Disable Code 39**

Enable/Disable Trioptic Code 39

Parameter # 13

Trioptic Code 39 is a variant of Code 39 used in the marking of computer tape cartridges. Trioptic Code 39 symbols always contain six characters. Enable or disable Trioptic Code 39.

- ***0 - Disable Code 39**
- **1 - Enable Code 39**

✓ **NOTE** Trioptic Code 39 and Code 39 Full ASCII cannot be enabled simultaneously.

Convert Code 39 to Code 32

Parameter # 86

Code 32 is a variant of Code 39 used by the Italian pharmaceutical industry. Enable or disable converting Code 39 to Code 32.

- ***0 - Disable Convert Code 39 to Code 32**
- **1 - Enable Convert Code 39 to Code 32**

✓ **NOTE** Code 39 must be enabled for this parameter to function.

Code 32 Prefix

Parameter # 231

Enable or disable adding the prefix character “A” to all Code 32 bar codes.

- *0 - Disable Code 32 Prefix
- 1 - Enable Code 32 Prefix

✓ **NOTE** Convert Code 39 to Code 32 must be enabled for this parameter to function.

Set Lengths for Code 39

Length1 Parameter #18 [Range: 0..55]

Length2 Parameter #19 [Range: 0..55]

Default: Length1=2, Length2=55

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for Code 39 to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of Code 39 to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14, Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of Code 39 to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode Code 39 codes of either 2 or 14 characters only, set **Length1 = 14, Length2 = 2**.
- **Length Within Range** - To decode only Code 39 codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode Code 39 codes of length 4 through 12 characters, set **Length1 = 4, Length2 = 12**.

Code 39 Check Digit Verification

Parameter # 48

Enable this to check the integrity of all Code 39 symbols to verify that the data complies with specified check digit algorithm. Only Code 39 symbols which include a modulo 43 check digit are decoded. Enable this feature if the Code 39 symbols contain a Modulo 43 check digit.

- *0 - Disable Code 39 Check Digit Verification
- 1 - Enable Code 39 Check Digit Verification

Transmit Code 39 Check Digit

Parameter # 43

Select whether to transmit Code 39 data with or without the check digit.

- *0 - Do Not Transmit Code 39 Check Digit (Disable)
- 1 - Transmit Code 39 Check Digit (Enable)

✓ **NOTE** Code 39 Check Digit Verification must be enabled for this parameter to function.

Code 39 Full ASCII Conversion

Parameter # 17

Code 39 Full ASCII is a variant of Code 39 which pairs characters to encode the full ASCII character set. Enable or disable Code 39 Full ASCII:

- *0 - Disable Code 39 Full ASCII
- 1 - Enable Code 39 Full ASCII



NOTE Trioptic Code 39 and Code 39 Full ASCII cannot be enabled simultaneously.

Code 39 Full ASCII to Full ASCII Correlation is host-dependent.

Code 39 Reduced Quiet Zone

Parameter # 1209

Enable or disable decoding Code 39 bar codes with reduced quiet zones. If you select **Enable**, select a [1D Quiet Zone Level on page 5-36](#).

- *0 - Disable Code 39 Reduced Quiet Zone
- 1 - Enable Code 39 Reduced Quiet Zone

Enable/Disable Code 93

Parameter # 9

Enable or disable Code 93:

- *0 - Disable Code 93
- 1 - Enable Code 93

Set Lengths for Code 93

Length1 Parameter #26 [Range: 0..55]

Length2 Parameter #27 [Range: 0..55]

Default: Length1=4, Length2=55

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for Code 93 to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of Code 93 to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14, Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of Code 93 to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode Code 93 codes of either 2 or 14 characters only, set **Length1 = 14, Length2 = 2**.
- **Length Within Range** - To decode only Code 93 codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode Code 93 codes of length 4 through 12 characters, set **Length1 = 4, Length2 = 12**.

Code 11

Parameter # 10

Enable or disable Code 11:

- *0 - Disable Code 11
- 1 - Enable Code 11

Set Lengths for Code 11

Length1 Parameter #28 [Range: 0..55]

Length2 Parameter #29 [Range: 0..55]

Default: Length1=4, Length2=55

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for Code 11 to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of Code 11 to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14, Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of Code 11 to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode Code 11 codes of either 2 or 14 characters only, set **Length1 = 14, Length2 = 2**.
- **Length Within Range** - To decode only Code 11 codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode Code 11 codes of length 4 through 12 characters, set **Length1 = 4, Length2 = 12**.

Code 11 Check Digit Verification

Parameter # 52

This feature allows the decoder to check the integrity of all Code 11 symbols to verify that the data complies with the specified check digit algorithm. This selects the check digit mechanism for the decoded Code 11 bar code. To enable this feature, set the number of check digits encoded in the Code 11 symbols:

- *0 - Disable Code 11 Check Digit Verification
- 1 - 1 Check Digit
- 2 - 2 Check Digits

Transmit Code 11 Check Digits

Parameter # 47

Select whether or not to transmit the Code 11 check digit(s).

- *0 - Do Not Transmit Code 11 Check Digit(s) (Disable)
- 1 - Transmit Code 11 Check Digit(s) (Enable)

✓ **NOTE** Code 11 Check Digit Verification must be enabled for this parameter to function.

Enable/Disable Interleaved 2 of 5

Parameter # 6

Enable or disable Interleaved 2 of 5:

- *1 - Enable Interleaved 2 of 5
- 0 - Disable Interleaved 2 of 5

Set Lengths for Interleaved 2 of 5

Length1 Parameter #22 [Range: 0..55]

Length2 Parameter #23 [Range: 0..55]

Default: Length1=14, Length2=0

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for I 2 of 5 to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of I 2 of 5 to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14, Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of I 2 of 5 to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode I 2 of 5 codes of either 2 or 14 characters only, set **Length1 = 14, Length2 = 2**.
- **Length Within Range** - To decode only I 2 of 5 codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode I 2 of 5 codes of length 4 through 12 characters, set **Length1 = 4, Length2 = 12**.
- **Any Length** - To decode I 2 of 5 codes of any length, set the values of **Length1** and **Length2** parameters to 0.

✓ **NOTE** Due to the construction of the I 2 of 5 symbology, it is possible for a scan line covering only a portion of the code to be interpreted as a complete scan, yielding less data than is encoded in the bar code. To prevent this, select specific lengths (one or two discrete lengths) for I 2 of 5 applications.

I 2 of 5 Check Digit Verification

Parameter # 49

Enable this feature to check the integrity of all I 2 of 5 symbols to verify the data complies with either the specified Uniform Symbology Specification (USS), or the Optical Product Code Council (OPCC) check digit algorithm.

- *0 - Disable
- 1 - USS Check Digit
- 2 - OPCC Check Digits

Transmit I 2 of 5 Check Digit

Parameter # 44

Select whether to transmit I 2 of 5 data with or without the check digit:

- *0 - Do Not Transmit I 2 of 5 Check Digit (Disable)
- 1 - Transmit I 2 of 5 Check Digit (Enable)

Convert I 2 of 5 to EAN-13

Parameter # 82

Enable this parameter to convert 14-character I 2 of 5 codes to EAN-13, and transmit to the host as EAN-13. To accomplish this, the I 2 of 5 code must be enabled, and the code must have a leading zero and a valid EAN-13 check digit.

- *0 - Do Not Convert I 2 of 5 to EAN-13 (Disable)
- 1 - Convert I 2 of 5 to EAN-13 (Enable)

I 2 of 5 Security Level

Parameter # 1121

Interleaved 2 of 5 bar codes are vulnerable to misdecodes, particularly when I 2 of 5 Lengths is set to **Any Length**. The scanner offers four levels of decode security for Interleaved 2 of 5 bar codes. There is an inverse relationship between security and scanner aggressiveness. Increasing the level of security can reduce scanning aggressiveness, so select only the level of security necessary.

- **0 - I 2 of 5 Security Level 0:** This setting allows the scanner to operate in its most aggressive state, while providing sufficient security in decoding most in-spec bar codes.
- ***1 - I 2 of 5 Security Level 1:** A bar code must be successfully read twice, and satisfy certain safety requirements before being decoded. This default setting eliminates most misdecodes.
- **2 - I 2 of 5 Security Level 2:** Select this option with greater bar code security requirements if **Security Level 1** fails to eliminate misdecodes.
- **3 - I 2 of 5 Security Level 3:** If you selected **Security Level 2**, and misdecodes still occur, select this security level. The highest safety requirements are applied. A bar code must be successfully read three times before being decoded.



NOTE Selecting this option is an extreme measure against mis-decoding severely out-of-spec bar codes. Selecting this level of security significantly impairs the decoding ability of the scanner. If this level of security is required, try to improve the quality of the bar codes.

I 2 of 5 Reduced Quiet Zone

Parameter # 1210

Enable or disable decoding I 2 of 5 bar codes with reduced quiet zones. If you select **Enable**, select a [1D Quiet Zone Level on page 5-36](#).

- *0 - Disable I 2 of 5 Reduced Quiet Zone
- 1 - Enable I 2 of 5 Reduced Quiet Zone

Enable/Disable Discrete 2 of 5

Parameter # 5

Enable or disable Discrete 2 of 5:

- *0 - Disable Discrete 2 of 5
- 1 - Enable Discrete 2 of 5

Set Lengths for Discrete 2 of 5

Length1 Parameter #20 [Range: 0..55]

Length2 Parameter #21 [Range: 0..55]

Default: Length1=12, Length2=0

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for D 2 of 5 to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of D 2 of 5 to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14, Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of D 2 of 5 to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode D 2 of 5 codes of either 2 or 14 characters only, set **Length1 = 14, Length2 = 2**.
- **Length Within Range** - To decode only D 2 of 5 codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode D 2 of 5 codes of length 4 through 12 characters, set **Length1 = 4, Length2 = 12**.

✓ **NOTE** Due to the construction of the D 2 of 5 symbology, it is possible for a scan line covering only a portion of the code to be interpreted as a complete scan, yielding less data than is encoded in the bar code. To prevent this, select specific lengths (one or two discrete lengths) for D 2 of 5 applications.

Enable/Disable Codabar

Parameter # 7

Enable or disable Codabar:

- *0 - Disable Codabar
- 1 - Enable Codabar

Set Lengths for Codabar

Length1 Parameter #24 [Range: 0..55]

Length2 Parameter #25 [Range: 0..55]

Default: Length1=5, Length2=55

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for Codabar to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of Codabar to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14, Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of Codabar to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode Codabar codes of either 2 or 14 characters only, set **Length1 = 14, Length2 = 2**.
- **Length Within Range** - To decode only Codabar codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode Codabar codes of length 4 through 12 characters, set **Length1 = 4, Length2 = 12**.

CLSI Editing

Parameter # 54

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. Enable this if the host system requires this data format.

- *0 - Disable CLSI Editing
- 1 - Enable CLSI Editing



NOTE Symbol length does not include start and stop characters.

NOTIS Editing

Parameter # 55

Enable this parameter to strip the start and stop characters from a decoded Codabar symbol. Enable this if the host system requires this data format.

- *0 - Disable NOTIS Editing
- 1 - Enable NOTIS Editing

Enable/Disable MSI

Parameter # 11

Enable or disable MSI.

- *0 - Disable MSI
- 1 - Enable MSI

Set Lengths for MSI

Length1 Parameter #30 [Range: 0..55]

Length2 Parameter #31 [Range: 0..55]

Default: Length1=4, Length2=55

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for MSI to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of MSI to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14**, **Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of MSI to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode MSI codes of either 2 or 14 characters only, set **Length1 = 14**, **Length2 = 2**.
- **Length Within Range** - To decode only MSI codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode MSI codes of length 4 through 12 characters, set **Length1 = 4**, **Length2 = 12**.

✓ **NOTE** Due to the construction of the MSI symbology, it is possible for a scan line covering only a portion of the code to be interpreted as a complete scan, yielding less data than is encoded in the bar code. To prevent this, select specific lengths (one or two discrete lengths) for MSI applications.

MSI Check Digits

Parameter # 50

With MSI symbols, one check digit is mandatory and always verified by the reader. The second check digit is optional. If the MSI codes include two check digits, select **Two MSI Check Digits** to enable verification of the second check digit:

- *0 - One MSI Check Digit
- 1 - Two MSI Check Digits

See [MSI Check Digit Algorithm on page 5-25](#) to select second digit algorithms.

Transmit MSI Check Digit(s)

Parameter # 46

Select whether to transmit MSI data with or without the check digit.

- *0 - Do Not Transmit MSI Check Digit(s) (Disable)
- 1 - Transmit MSI Check Digit(s) (Enable)

MSI Check Digit Algorithm

Parameter # 51

Select one of two algorithms for the verification of the second MSI check digit:

- *1 - MOD 10/MOD 10
- 0 - MOD 10/MOD 11

Enable/Disable Chinese 2 of 5

Parameter # 408

Enable or disable Chinese 2 of 5:

- *0 - Disable Chinese 2 of 5
- 1 - Enable Chinese 2 of 5

Enable/Disable Korean 3 of 5

Parameter # 581

Enable or disable Korean 3 of 5:

- *0 - Disable Korean 3 of 5
- 1 - Enable Korean 3 of 5



NOTE The length for Korean 3 of 5 is fixed at 6.

Enable/Disable Matrix 2 of 5

Parameter # 618

Enable or disable Matrix 2 of 5.

- *0 - Disable Matrix 2 of 5
- 1 - Enable Matrix 2 of 5

Set Lengths for Matrix 2 of 5

Length1 Parameter #619 [Range: 0..55]

Length2 Parameter #620 [Range: 0..55]

Default: Length1=14, Length2=0

The length of a code refers to the number of characters (i.e., human readable characters), including check digit(s) the code contains. Assign lengths for Matrix 2 of 5 to decode either one or two discrete lengths, or a length within a specific range.

- **One Discrete Length** - To limit the decoding of Matrix 2 of 5 to one specific length, assign this length to the **Length1** parameter and 0 to the **Length2** parameter. For example, for fixed length 14, set **Length1 = 14, Length2 = 0**.
- **Two Discrete Lengths** - To limit the decoding of Matrix 2 of 5 to either of two specific lengths, assign the greater length to the **Length1** parameter and the lesser to **Length2**. For example, to decode Matrix 2 of 5 codes of either 2 or 14 characters only, set **Length1 = 14, Length2 = 2**.
- **Length Within Range** - To decode only Matrix 2 of 5 codes that fall within a specific length range, assign the lesser length to the **Length1** parameter and the greater to the **Length2** parameter. For example, to decode Matrix 2 of 5 codes of length 4 through 12 characters, set **Length1 = 4, Length2 = 12**.

Matrix 2 of 5 Redundancy

Parameter # 621

Enable or disable Matrix 2 of 5 redundancy:

- *0 - Disable Matrix 2 of 5 Redundancy
- 1 - Enable Matrix 2 of 5 Redundancy

Matrix 2 of 5 Check Digit

Parameter # 622

The check digit is the last character of the symbol used to verify the integrity of the data. Select whether to transmit the bar code data with or without the Matrix 2 of 5 check digit:

- *0 - Disable Matrix 2 of 5 Check Digit
- 1 - Enable Matrix 2 of 5 Check Digit

Transmit Matrix 2 of 5 Check Digit

Parameter # 623

Select whether to transmit Matrix 2 of 5 data with or without the check digit.

- *0 - Do Not Transmit Matrix 2 of 5 Check Digit
- 1 - Transmit Matrix 2 of 5 Check Digit

Inverse 1D

Parameter # 586

Set the 1D inverse decoder setting:

- *0 - Regular Only - the decoder decodes regular 1D bar codes only.
- 1 - Inverse Only - the decoder decodes inverse 1D bar codes only.
- 2 - Inverse Autodetect - the decoder decodes both regular and inverse 1D bar codes.

US Postnet

Parameter # 89

Enable or disable US Postnet:

- *1 - Enable US Postnet
- 0 - Disable US Postnet

US Planet

Parameter # 90

Enable or disable US Planet:

- *1 - Enable US Planet
- 0 - Disable US Planet

Transmit US Postal Check Digit

Parameter # 95

Select whether to transmit US Postal data, which includes both US Postnet and US Planet, with or without the check digit:

- *1 - Transmit US Postal Check Digit
- 0 - Do Not Transmit US Postal Check Digit

UK Postal

Parameter # 91

Enable or disable UK Postal:

- ***1 - Enable UK Postal**
- **0 - Disable UK Postal**

Transmit UK Postal Check Digit

Parameter # 96

Select whether to transmit UK Postal data with or without the check digit:

- ***1 - Transmit UK Postal Check Digit**
- **0 - Do Not Transmit UK Postal Check Digit**

Japan Postal

Parameter # 290

Enable or disable Japan Postal:

- ***1 - Enable Japan Postal**
- **0 - Disable Japan Postal**

Australia Post

Parameter # 291

Enable or disable Australia Post:

- ***1 - Enable Australia Post**
- **0 - Disable Australia Post**

Australia Post Format

Parameter # 718

Select one of the following formats for Australia Post:

- ***0 - Autodiscriminate** (or Smart mode) - Attempt to decode the Customer Information Field using the N and C Encoding Tables.
- ✓ **NOTE** This option increases the risk of misdecodes because the encoded data format does not specify the Encoding Table used for encoding.
- **1 - Raw Format** - Output raw bar patterns as a series of numbers 0 through 3.
- **2 - Alphanumeric Encoding** - Decode the Customer Information Field using the C Encoding Table.
- **3 - Numeric Encoding** - Decode the Customer Information Field using the N Encoding Table.

For more information on Australia Post Encoding Tables, refer to the *Australia Post Customer Barcoding Technical Specifications* available at <http://www.auspost.com.au>.

Netherlands KIX Code

Parameter # 326

Enable or disable Netherlands KIX Code:

- *1 - Enable Netherlands KIX Code
- 0 - Disable Netherlands KIX Code

USPS 4CB/One Code/Intelligent Mail

Parameter # 592

Enable or disable USPS 4CB/One Code/Intelligent Mail:

- *0 - Disable USPS 4CB/One Code/Intelligent Mail
- 1 - Enable USPS 4CB/One Code/Intelligent Mail

UPU FICS Postal

Parameter # 611

Enable or disable UPU FICS Postal:

- *0 - Disable UPU FICS Postal
- 1 - Enable UPU FICS Postal

GS1 DataBar-14

Parameter # 338

Enable or disable GS1 DataBar-14:

- *1 - Enable GS1 DataBar-14
- 0 - Disable GS1 DataBar-14

GS1 DataBar Limited

Parameter # 339

Enable or disable GS1 DataBar Limited:

- *0 - Disable GS1 DataBar Limited
- 1 - Enable GS1 DataBar Limited

GS1 DataBar Limited Security Level

Parameter # 728

There are four levels of decode security for GS1 DataBar Limited bar codes. There is an inverse relationship between security and scanner aggressiveness. Increasing the level of security may result in reduced aggressiveness in scanning, so only choose the level of security necessary.

- **1 - Level 1** – No clear margin required. This complies with the original GS1 standard, yet might result in erroneous decoding of the DataBar Limited bar codes when scanning some UPC symbols that start with the digits “9” and “7”.
- **2 - Level 2** – Automatic risk detection. This level of security may result in erroneous decoding of DataBar Limited bar codes when scanning some UPC symbols. If a misdecode is detected, the scanner operates in Level 3 or Level 1.
- ***3 - Level 3** – Security level reflects newly proposed GS1 standard that requires a 5X trailing clear margin.
- **4 - Level 4** – Security level extends beyond the standard required by GS1. This level of security requires a 5X leading and trailing clear margin.

GS1 DataBar Expanded

Parameter # 340

Enable or disable GS1 DataBar Expanded:

- ***0 - Disable GS1 DataBar Expanded**
- **1 - Enable GS1 DataBar Expanded**

Convert GS1 DataBar to UPC/EAN

Parameter # 397

This parameter only applies to GS1 DataBar-14 and GS1 DataBar Limited symbols not decoded as part of a Composite symbol. Enable this to strip the leading 010 from GS1 DataBar-14 and GS1 DataBar Limited symbols encoding a single zero as the first digit, and report the bar code as EAN-13.

For bar codes beginning with two or more zeros but not six zeros, this parameter strips the leading '0100 and reports the bar code as UPC-A. The UPC-A Preamble parameter that transmits the system character and country code applies to converted bar codes. Note that neither the system character nor the check digit can be stripped.

- ***0 - Disable Convert GS1 DataBar to UPC/EAN**
- **1 - Enable Convert GS1 DataBar to UPC/EAN**

Composite CC-C

Parameter # 341

Enable or disable Composite bar codes of type CC-C:

- *0 - Disable CC-C
- 1 - Enable CC-C

Composite CC-A/B

Parameter # 342

Enable or disable Composite bar codes of type CC-A/B:

- *0 - Disable CC-A/B
- 1 - Enable CC-A/B

Composite TLC-39

Parameter # 371

Enable or disable Composite bar codes of type TLC-39:

- *0 - Disable TLC39
- 1 - Enable TLC39

UPC Composite Mode

Parameter # 344

Select an option for linking UPC symbols with a 2D symbol during transmission as if they were one symbol:

- **1 - UPC Always Linked** - transmit UPC bar codes and the 2D portion. If 2D is not present, the UPC bar code does not transmit.
- ***0 - UPC Never Linked** - transmit UPC bar codes regardless of whether a 2D symbol is detected.
- **2 - Autodiscriminate UPC Composites** - the imager engine determines if there is a 2D portion, then transmits the UPC, as well as the 2D portion if present.

GS1-128 Emulation Mode for UCC/EAN Composite Codes

Parameter # 427

Enable or disable this mode:

- *0 - Disable GS1-128 Emulation Mode for UCC/EAN Composite Codes
- 1 - Enable GS1-128 Emulation Mode for UCC/EAN Composite Codes

Enable/Disable PDF417

Parameter # 15

Enable or disable PDF417:

- ***1 - Enable PDF417**
- **0 - Disable PDF417**

Enable/Disable MicroPDF417

Parameter # 227

Enable or disable MicroPDF417.

- ***0 - Disable MicroPDF417**
- **1 - Enable MicroPDF417**

Code 128 Emulation

Parameter # 123

Enable this parameter to transmit data from certain MicroPDF417 symbols as if it was encoded in Code 128 symbols. Transmit AIM Symbology Identifiers must be enabled for this parameter to work.

- ***0 - Disable Code 128 Emulation** - transmits these MicroPDF417 symbols with one of the following prefixes:
 -]L3 if the first codeword is 903-905
 -]L4 if the first codeword is 908 or 909
 -]L5 if the first codeword is 910 or 911
- **1 - Enable Code 128 Emulation** - transmits these MicroPDF417 symbols with one of the following prefixes:
 -]C1 if the first codeword is 903-905
 -]C2 if the first codeword is 908 or 909
 -]C0 if the first codeword is 910 or 911



NOTE Linked MicroPDF codewords 906, 907, 912, 914, and 915 are not supported. Use GS1 Composites instead.

Data Matrix

Parameter # 292

Enable or disable Data Matrix:

- ***1 - Enable Data Matrix**
- **0 - Disable Data Matrix**

Data Matrix Inverse

Parameter # 588

Set the Data Matrix inverse decoder setting:

- ***0 - Regular Only** - the decoder decodes regular Data Matrix bar codes only.
- **1 - Inverse Only** - the decoder decodes inverse Data Matrix bar codes only.
- **2 - Inverse Autodetect** - the decoder decodes both regular and inverse Data Matrix bar codes.

Decode Mirror Images (Data Matrix Only)

Parameter # 537

Select an option for decoding mirror image Data Matrix bar codes:

- ***0 - Never** - do not decode Data Matrix bar codes that are mirror images
- **1 - Always** - decode only Data Matrix bar codes that are mirror images
- **2 - Auto** - decode both mirrored and unmirrored Data Matrix bar codes.

Maxicode

Parameter # 294

Enable or disable Maxicode:

- ***1 - Enable Maxicode**
- **0 - Disable Maxicode**

QR Code

Parameter # 293

Enable or disable QR Code:

- ***1 - Enable QR Code**
- **0 - Disable QR Code**

QR Inverse

Parameter # 587

Set the QR inverse decoder setting:

- ***0 - Regular Only** - the decoder decodes regular QR bar codes only.
- **1 - Inverse Only** - the decoder decodes inverse QR bar codes only.
- **2 - Inverse Autodetect** - the decoder decodes both regular and inverse QR bar codes.

MicroQR

Parameter # 573

Enable or disable MicroQR:

- ***1 - Enable MicroQR**
- **0 - Disable MicroQR**

Aztec

Parameter # 574

Enable or disable Aztec:

- ***1 - Enable Aztec**
- **0 - Disable Aztec**

Aztec Inverse

Parameter # 589

Set the Aztec inverse decoder setting:

- ***0 - Regular Only** - the decoder decodes regular Aztec bar codes only.
- **1 - Inverse Only** - the decoder decodes inverse Aztec bar codes only.
- **2 - Inverse Autodetect** - the decoder decodes both regular and inverse Aztec bar codes.

Han Xin

Parameter # 1167

Enable or disable Han Xin:

- *0 - Disable Han Xin
- 1 - Enable Han Xin

Han Xin Inverse

Parameter # 1168

Set the Han Xin inverse decoder setting:

- *0 - **Regular Only** - the decoder decodes Han Xin bar codes with normal reflectance only.
- 1 - **Inverse Only** - the decoder decodes Han Xin bar codes with inverse reflectance only.
- 2 - **Inverse Autodetect** - the decoder decodes both regular and inverse Han Xin bar codes.

Redundancy Level

Parameter # 78

The decoder offers four levels of decode redundancy. Select higher redundancy levels for decreasing levels of bar code quality. As redundancy levels increase, the decoder's aggressiveness decreases.

Select the redundancy level appropriate for the bar code quality:

- ***Redundancy Level 1** - The decoder must read the following code types twice before decoding:
 - Codabar (8 characters or less)
 - MSI (4 characters or less)
 - D 2 of 5 (8 characters or less)
 - I 2 of 5 (8 characters or less)
- **Redundancy Level 2** - The decoder must read all code types twice before decoding.
- **Redundancy Level 3** - The decoder must read code types other than the following twice before decoding, but must read the following codes three times:
 - Codabar (8 characters or less)
 - MSI (4 characters or less)
 - D 2 of 5 (8 characters or less)
 - I 2 of 5 (8 characters or less)
- **Redundancy Level 4** - The decoder must read all code types three times before decoding.

Security Level

Parameter # 77

The decoder offers four levels of decode security for delta bar codes, which include the Code 128 family, UPC/EAN, and Code 93. Select increasing levels of security for decreasing levels of bar code quality. There is an inverse relationship between security and decoder aggressiveness, so choose only that level of security necessary for any given application.

- **0 - Security Level 0:** This setting allows the decoder to operate in its most aggressive state, while providing sufficient security in decoding most “in-spec” bar codes.
- ***1 - Security Level 1:** Select this option if misdecodes occur. This default setting eliminates most misdecodes.
- **2 - Security Level 2:** Select this option if Security level 1 fails to eliminate misdecodes.
- **3 - Security Level 3:** If misdecodes still occur with Security Level 2, select this security level. Be advised, selecting this option is an extreme measure against mis-decoding severely out of spec bar codes. Selecting this level of security significantly impairs the decoding ability of the decoder. If this level of security is necessary, try to improve the quality of the bar codes.

1D Quiet Zone Level

Parameter # 1288

This feature sets the level of aggressiveness in decoding bar codes with a reduced quiet zone (the area in front of and at the end of a bar code), and applies to symbologies enabled by a Reduced Quiet Zone parameter. Because higher levels increase the decoding time and risk of misdecodes, we strongly recommends enabling only the symbologies which require higher quiet zone levels, and leaving Reduced Quiet Zone disabled for all other symbologies. Options are:

- **0 - 1D Quiet Zone Level 0:** The decoder performs normally in terms of quiet zone.
- ***1 - 1D Quiet Zone Level 1:** The decoder performs more aggressively in terms of quiet zone.
- **2 - 1D Quiet Zone Level 2:** The decoder only requires one side EB (end of bar code) for decoding.
- **3 - 1D Quiet Zone Level 3:** The decoder decodes anything in terms of quiet zone or end of bar code.

Intercharacter Gap Size

Parameter # 381

The Code 39 and Codabar symbologies have an intercharacter gap that is typically quite small. Due to various bar code printing technologies, this gap can grow larger than the maximum size allowed, preventing the decoder from decoding the symbol. If this problem occurs, select **Large Intercharacter Gaps** to tolerate these out-of-specification bar codes.

- ***6 - Normal Intercharacter Gaps**
- **10 - Large Intercharacter Gaps**

CHAPTER 6 OCR PROGRAMMING

Introduction

This chapter describes the programmable OCR programming parameters available via the *Software Decode SDK for Android*, and includes the parameter numbers and option values to use. The SDK supports font types OCR-A, OCR-B, MICR-E13B, and US Currency Serial Number.

OCR is not as secure as a bar code. To decrease OCR misdecodes and speed OCR reading, set an accurate OCR template and character subset, and use a check digit.

All OCR fonts are disabled by default. Enabling OCR can slow bar code decoding. Enabling more than one OCR font could also slow OCR decoding and impact OCR decoding accuracy.

✓ **NOTE** The SE2100 does not support OCR programming.

OCR Parameter Defaults

Table 6-1 lists the defaults for OCR parameters.

Table 6-1 *OCR Programming Default Table*

Parameter	Parameter Number (Decimal)	Default	Page Number
OCR-A	680	Disable	6-3
OCR-A Variant	684	Full ASCII	6-3
OCR-B	681	Disable	6-4
OCR-B Variant	685	Full ASCII	6-4
MICR E13B	682	Disable	6-5
US Currency	683	Disable	6-6
OCR Orientation	687	0°	6-6
OCR Lines	691	1	6-6
OCR Minimum Characters	689	3	6-6
OCR Maximum Characters	690	100	6-6
OCR Subset	686	Selected font variant	6-7
OCR Quiet Zone	695	50	6-7
OCR Template	547	54R	6-8
OCR Check Digit Modulus	688	1	6-13
OCR Check Digit Multiplier	700	1212121212	6-14
OCR Check Digit Validation	694	None	6-15
Inverse OCR	856	Regular	6-17

OCR Programming Parameters

In this section, * indicates the default option.

- ✓ **NOTE** OCR is not as secure as a bar code. To decrease OCR misdecodes and speed OCR reading, set an accurate OCR template and character subset, and use a check digit. See [OCR Subset on page 6-7](#) and [OCR Template on page 6-8](#).
- ✓ **NOTE** All OCR fonts are disabled by default. Enabling OCR can slow bar code decoding. Enabling more than one OCR font could also slow OCR decoding and impact OCR decoding accuracy.

Enable/Disable OCR-A

Parameter # 680

Enable or disable OCR-A:

- 1 - Enable OCR-A
- *0 - Disable OCR-A

OCR-A Variant

Parameter # 684

- ✓ **NOTE** Enable OCR-A before setting this parameter. If disabling OCR-A, set the variant to its default (OCR-A Full ASCII).

Font variant sets a processing algorithm and default character subset for the given font. Select one of the following supported variants. Selecting the most appropriate font variant optimizes performance and accuracy.

- *0 - OCR-A Full ASCII
!"#\$()*+,-./0123456789<>ABCDEFGHIJKLMNOPQRSTUVWXYZ^
- 1 - OCR-A Reserved 1
\$*+,-./0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
- 2 - OCR-A Reserved 2
\$*+,-./0123456789<>ABCDEFGHIJKLMNOPQRSTUVWXYZ
- 3 - OCR-A Banking
-0123456789<> ƒ ɹ ʞ

Special banking characters output as the following representative characters:

- ƒ outputs as f
- ɹ outputs as c
- ʞ outputs as h

- ✓ **NOTE** Enable OCR-A before setting this parameter. If disabling OCR-A, set the variant to its default (OCR-A Full ASCII).

Enable/Disable OCR-B

Parameter # 681

Enable or disable OCR-B:

- **1 - Enable OCR-B**
- ***0 - Disable OCR-B**

OCR-B Variant

Parameter # 685

Font variant sets a processing algorithm and default character subset for the given font. Selecting the most appropriate font variant affects performance and accuracy.

Selecting the following OCR-B variants automatically sets the appropriate [OCR Lines on page 6-6](#). These five variants invoke extensive special algorithms and checking for that particular document type:

Variant	OCR Lines Setting
Passport	2
TD1 ID Cards	3
TD2 ID Cards	2
Visa Type A	2
Visa Type B	2

Selecting one of the ISBN Book Numbers automatically applies the appropriate ISBN checksum, so you do not need to set this.

For the best performance in passport reading, fix the target passport and the decoder in place (6.5 - 7.5”).

✓ **NOTE** Enable OCR-B before setting this parameter. If disabling OCR-B, set the variant to its default (OCR-B Full ASCII).

Select one of the following supported OCR-B variants.

- ***0 - OCR-B Full ASCII**
!#\$%()*+,-./0123456789<>ABCDEFGHIJKLMNOPQRSTUVWXYZ^|ñ
- **1 - OCR-B Banking**
#+-0123456789<>JNP|
- **2 - OCR-B Limited**
+,-./0123456789<>ACENPSTVX
- **6 - OCR-B ISBN 10-Digit Book Numbers**
-0123456789>BCEINPSXz
- **7 - OCR-B ISBN 10 or 13-Digit Book Numbers**
-0123456789>BCEINPSXz
- **3 - OCR-B Travel Document Version 1 (TD1) 3-Line ID Cards**
-0123456789<ABCDEFGHIJKLMNOPQRSTUVWXYZ
- **8 - OCR-B Travel Document Version 2 (TD2) 2-Line ID Cards**
-0123456789<ABCDEFGHIJKLMNOPQRSTUVWXYZ

- **20 - OCR-B Travel Document 2 or 3-Line ID Cards Auto-Detect**

!#\$%()*+,-./0123456789<>ABCDEFGHIJKLMNOPQRSTUVWXYZ^|ñ

- **4 - OCR-B Passport**

-0123456789<ABCDEFGHIJKLMNOPQRSTUVWXYZñ

- **9 - OCR-B Visa Type A**

-0123456789<ABCDEFGHIJKLMNOPQRSTUVWXYZ

- **10 - OCR-B Visa Type B**

-0123456789<ABCDEFGHIJKLMNOPQRSTUVWXYZñ

- **11 - OCR-B ICAO Travel Documents**

This allows reading either TD1, TD2, Passport, Visa Type A, or Visa Type B without switching between these options. It automatically recognizes the travel document read.

Enable/Disable MICR E13B

Parameter # 682

MICR E 13B uses the following characters:

0 1 2 3 4 5 6 7 8 9 t a o d

TOAD characters (Transit, On Us, Amount, and Dash) output as the following representative characters:

t outputs as t

a outputs as a

o outputs as o

d outputs as d

Enable or disable MICR E13B.

- **1 - Enable MICR E13B**
- ***0 - Disable MICR E13B**

Enable/Disable US Currency Serial Number

Parameter # 683

Enable or disable US Currency Serial Number.

- **1 - Enable US Currency**
- ***0 - Disable US Currency**

OCR Orientation

Parameter # 687

Select one of five options to specify the orientation of an OCR string to be read. Setting an incorrect orientation can cause misdecodes.

- ***0 - 0° to the imaging engine (default)**
- **1 - 270° clockwise (or 90° counterclockwise) to the imaging engine**
- **2 - 180° (upside down) to the imaging engine**
- **3 - 90° clockwise to the imaging engine**
- **4 - Omnidirectional**

OCR Lines

Parameter # 691

Select the specific number of OCR lines to decode. Selecting Visas, TD1, or TD2 ID cards automatically sets the appropriate number of **OCR Lines**. Also see [OCR-B Variant on page 6-4](#).

✓ **NOTE** This parameter sets the exact number of lines to decode, not the minimum.

- ***1 - Decode OCR 1 Line**
- **2 - Decode OCR 2 Lines**
- **3 - Decode OCR 3 Lines**

OCR Minimum Characters

Parameter # 689

Set the minimum number of OCR characters (not including spaces) per line to decode from 3 and 100. Strings of OCR characters less than the minimum are ignored. The default is 3.

OCR Maximum Characters

Parameter # 690

Set the maximum number of OCR characters (including spaces) per line to decode from 3 and 100. Strings of OCR characters greater than the maximum are ignored. The default is 100.

OCR Subset

Parameter # 686

Set an OCR subset to define a custom group of characters in place of a preset font variant. For example, if scanning only numbers and the letters A, B, and C, create a subset of just these characters to speed decoding. This applies a designated OCR Subset across all enabled OCR fonts.

To set or modify the OCR font subset, first enable the appropriate OCR font(s). Next, for this parameter, set a string of numbers and letters in the application to form the desired OCR Subset.

To cancel an OCR subset, for OCR-A or OCR-B, set OCR-A variant **Full ASCII** or OCR-B variant **Full ASCII**, and clear any previously set subsets to a null string.

For MICR E13B or US Currency Serial Number, create a subset which includes all allowed characters in that character set.

OCR Quiet Zone

Parameter # 695

This option sets the OCR quiet zone. The decoder stops scanning a field when it detects a sufficiently wide blank space. The width of this space is defined by the End of Field option. Used with parsers that tolerate slanted characters, the End of Field count is roughly a count of 8 for a character width.

For example if set to 15, then two character widths are an end of line indicator for the parser. Larger end of field numbers require bigger quiet zones at each end of text line.

Set a quiet zone in the range of 20 - 99. The default is 50, indicating a six character width quiet zone.

OCR Template

Parameter # 547

This option creates a template for precisely matching scanned OCR characters to a desired input format. Carefully constructing an OCR template eliminates scanning errors.

To set or modify the OCR decode template, select the following numbers and letters to form the template expression. The default is **54R** which accepts any character OCR strings. See the remainder of this section for explanations of options.

- **9 - Required Digit**
- **A - Required Alpha**
- **1 - Optional Alphanumeric**
- **2 - Optional Alpha**
- **3 - Alpha or Digit**
- **4 - Any Including Space & Reject**
- **5 - Any Except Space & Reject**
- **7 - Optional Digit**
- **8 - Digit or Fill**
- **F - Alpha or Fill**
- **() [space] - Optional Space**

Other template operators:

- **" - Literal String**
- **+ - Literal String**
- **E - New Line**
- **C - String Extract**
- **D - End of Field**
- **P1 - Skip Until**
- **P0 - Skip Until Not**
- **R - Repeat Previous**
- **S - Scroll Until Match**

Required Digit (9)

Only a numeric character is allowed in this position.

Template	Valid data	Valid data	Invalid data
99999	12987	30517	123AB

Required Alpha (A)

Only an alpha character is allowed in this position.

Template	Valid data	Valid data	Invalid data
AAA	ABC	WXY	12F

Optional Alphanumeric (1)

When this option appears in the template string, the data validator accepts an alphanumeric character if present. Optional characters are not allowed as the first character(s) in a field of like characters.

Template	Valid data	Valid data	Invalid data
99991	1234A	12345	1234<

Optional Alpha (2)

When this option appears in the template string, the data validator accepts an alpha character if present. Optional characters are not allowed as the first character(s) in a field of like characters.

Template	Valid data	Valid data	Invalid data
AAAA2	ABCDE	WXYZ	ABCD6

Alpha or Digit (3)

The data validator requires an alphanumeric character in this position to validate the incoming data.

Template	Valid data	Valid data	Invalid data
33333	12ABC	WXY34	12AB<

Any Including Space & Reject (4)

The template accepts any character in this position, including space and reject. Rejects are represented as an underscore (_) in the output. This is a good selection for troubleshooting.

Template	Valid data	Valid data
99499	12\$34	34_98

Any except Space & Reject (5)

The template accepts any character in this position except a space or reject.

Template	Valid data	Valid data	Invalid data
55999	A.123	*Z456	A BCD

Optional Digit (7)

When this option appears in the template string, the template accepts a numeric character if present. Optional characters are not allowed as the first character(s) in a field of like characters.

Template	Valid data	Valid data	Invalid data
99977	12345	789	789AB

Digit or Fill (8)

The data validator accepts any numeric or fill character in this position.

Template	Valid data	Valid data	Valid data
88899	12345	>>789	<<789

Alpha or Fill (F)

The data validator accepts any alpha or fill character in this position.

Template	Valid data	Valid data	Valid data
AAAFF	ABCXY	LMN>>	ABC<5

Optional Space ()

When this option appears in the template string, the template accepts a space if present. Optional characters are not allowed as the first character(s) in a field of like characters.

Template	Valid data	Valid data	Invalid data
99 99	12 34	1234	67891

Optional Small Special (.)

When this option appears in the template string, the data validator accepts a special character if present. Optional characters are not allowed as the first character(s) in a field of like characters. Small special characters are - , and .

Template	Valid data	Valid data	Invalid data
AA.99	MN.35	XY98	XYZ12

Other Template Operators

These template operators assist in capturing, delimiting, and formatting scanned OCR data.

Literal String (" and +)

Use either of these delimiting characters to surround characters to define a literal string within a template that must be present in scanned OCR data. There are two characters used to delimit required literal strings; if one of the delimiter characters is present in the desired literal string, use the other delimiter.

Template	Valid data	Invalid data
"35+BC"	35+BC	AB+22

New Line (E)

To create a template of multiple lines, add **E** between the template of each single line.

Template	Valid data	Valid data	Invalid data
999EAAAA	321	987	XYZW
	BCAD	ZXYW	12

String Extract (C)

This operator combined with others defines a string of characters to extract from the scanned data. The string extract is structured as follows:

CbPe

Where:

- **c** is the string extract operator
- **b** is the string begin delimiter
- **P** is the category (one or more numeric or alpha characters) describing the string representation
- **e** is the string end delimiter

Values for *b* and *e* can be any scannable character. They are included in the output stream.

Template	Incoming data	Output
C>A>	XQ3>ABCDE>	>ABCDE>
	->ATHRUZ>123	>ATHRUZ>
	1ABCZXYZ	No Output

Ignore to End of Field (D)

This operator causes all characters after a template to be ignored. Use this as the last character in a template expression. Examples for the template 999D:

Template	Incoming data	Output
999D	123-PED	123
	357298	357
	193	193

Skip Until (P1)

This operator allows skipping over characters until a specific character type or a literal string is detected. It can be used in two ways:

P1ct

Where:

- P1 is the Skip Until operator
- c is the type of character that triggers the start of output
- t is one or more template characters

P1"s"t

Where:

- P1 is the Skip Until operator
- "s" is one or more literal string characters (see [Literal String \(" and +\) on page 6-10](#)) that trigger the start of output
- t is one or more template characters

The trigger character or literal string is included in output from a Skip Until operator, and the first character in the template should accommodate this trigger.

Template	Incoming data	Output
P1"PN"AA9999	123PN9876	PN9876
	PN1234	PN1234
	X-PN3592	PN3592

Skip Until Not (P0)

This operator allows skipping over characters until a specific character type or a literal string is not matched in the output stream. It can be used in two ways:

```
P0ct
```

Where:

- P0 is the Skip Until Not operator
- c is the type of character that triggers the start of output
- t is one or more template characters

```
P0"s"t
```

Where:

- P0 is the Skip Until Not operator
- "s" is one or more literal string characters (see [Literal String \(" and +\) on page 6-10](#)) that trigger the start of output
- t is one or more template characters

The trigger character or literal string is not included in output from a Skip Until Not operator.

Template	Incoming data	Output
P0A9999	BP3456	3456
	PN1234	1234
	5341	No output

Template	Incoming data	Output
P0"PN"9999	PN3456	3456
	5341	No output
	PNPN7654	7654

Repeat Previous (R)

This operator allows a template character to repeat one or more times, allowing the capture of variable-length scanned data. The following examples capture two required alpha characters followed by one or more required digits:

Template	Incoming data	Output
AA9R	AB34	AB34
	PN12345	PN12345
	32RM52700	No output

Scroll Until Match (S)

This operator steps through scanned data one character at a time until the data matches the template.

Template	Incoming data	Output
S99999	AB3	No Output
	PN12345	12345
	32RM52700	52700

Multiple Templates

This feature sets up multiple templates for OCR decoding. To do this, follow the procedure described in [OCR Template on page 6-8](#) for each template in the multiple template string, using a capital letter **X** as a separator between the templates.

For example, set the [OCR Template](#) as **99999XAAAAA** to decode OCR strings of either **12345** or **ABCDE**. Up to 99 templates are permitted.

Template Examples

Following are sample templates with descriptions of valid data for each definition.

Field Definition	Description
"M"99977	M followed by three digits and two optional digits.
"X"997777"X"	X followed by two digits, four optional digits, and an X .
9959775599	Two digits followed by any character, a digit, two optional digits, any two characters, and two digits.
A55"- "999"- "99	A letter followed by two characters, a dash, three digits, a dash, and two digits.
33A" . "99	Two alphanumeric characters followed by a letter, a period, and two digits.
999992991	Five digits followed by an optional alpha, two digits, and an optional alphanumeric.
"PN98"	Literal field - PN98

OCR Check Digit Modulus

Parameter # 688

This option sets OCR module check digit calculation. The check digit is the last digit (in the right most position) in an OCR string and improves the accuracy of the collected data. The check digit is the end product of a calculation made on the incoming data. For check digit calculation, for example Modulus 10, alpha and numeric characters are assigned numeric weights (see [OCR Check Digit Multiplier on page 6-14](#)). The calculation is applied to the character weights and the resulting check digit is added to the end of the data. If the incoming data does not match the check digit, the data is considered corrupt.

The selected check digit option does not take effect until you set [OCR Check Digit Validation on page 6-15](#).

To select the Check Digit Modulus, such as 10 for modulo 10, set this parameter to a three-digit number from 001 to 099 representing the check digit. The default is 1.

OCR Check Digit Multiplier

Parameter # 700

This option sets OCR check digit multipliers for the character positions. For check digit validation, each character in scanned data has an equivalent weight used in the check digit calculation. Zebra device OCR ships with the following weight equivalents:

0 = 0	A = 10	K = 20	U = 30
1 = 1	B = 11	L = 21	V = 31
2 = 2	C = 12	M = 22	W = 32
3 = 3	D = 13	N = 23	X = 33
4 = 4	E = 14	O = 24	Y = 34
5 = 5	F = 15	P = 25	Z = 35
6 = 6	G = 16	Q = 26	Space = 0
7 = 7	H = 17	R = 27	
8 = 8	I = 18	S = 28	
9 = 9	J = 19	T = 29	

All other characters are equivalent to one (1).

You can define the multiplier string if it is different from the default.

121212121212 (default)

123456789A (for ISBN, Product Add Right to Left. See [OCR Check Digit Validation on page 6-15](#))

For example:

ISBN	0	2	0	1	1	8	3	9	9	4
Multiplier	10	9	8	7	6	5	4	3	2	1
Product	0	18	0	7	6	40	12	27	18	4
Product add	0+	18+	0+	7+	6+	40+	12+	27+	18+	4= 132

ISBN uses modulo 11 for its check digit. In this case, 132 is divisible by 11, so it passes the check digit.

To set the check digit multiplier, set numbers and letters to form the multiplier string.

OCR Check Digit Validation

Parameter # 694

Use **OCR Check Digit Validation** to protect against scanning errors by applying a check digit validation scheme. Select one of the following options. See the remainder of this section for explanations of options.

- ***0 - No Check Digit**
- **3 - Product Add Left to Right**
- **1 - Product Add Right to Left**
- **4 - Digit Add Left to Right**
- **2 - Digit Add Right to Left**
- **5 - Product Add Right to Left Simple Remainder**
- **6 - Digit Add Right to Left Simple Remainder**
- **9 - Health Industry - HIBCC43**

No Check Digit (0)

No check digit validation, indicating no check digit is applied. This is the default.

Product Add Left to Right (3)

Each character in the scanned data is assigned a numeric value (see [OCR Check Digit Multiplier on page 6-14](#)). Each digit representing a character in the scanned data is multiplied by its corresponding digit in the multiplier, and the sum of these products is computed. The check digit passes if this sum modulo Check Digit Modulus is zero.

Example:

Scanned data numeric value is 132456 (check digit is 6)

Check digit multiplier string is 123456

Digit	1	3	2	4	5	6	
Multiplier	1	2	3	4	5	6	
Product	1	6	6	16	25	36	
Product add	1+	6+	6+	16+	25+	36=	90

The Check Digit Modulus is 10. It passes because 90 is divisible by 10 (the remainder is zero).

Product Add Right to Left (1)

Each character in the scanned data is assigned a numeric value (see [OCR Check Digit Multiplier on page 6-14](#)). The check digit multiplier is reversed in order. Each value representing a character in the scanned data is multiplied by its corresponding digit in the reversed multiplier, resulting in a product for each character in the scanned data. The sum of these products is computed. The check digit passes if this sum modulo Check Digit Modulus is zero.

Example:

Scanned data numeric value is 132459 (check digit is 9)

Check digit multiplier string is 123456

Digit	1	3	2	4	5	9
Multiplier	6	5	4	3	2	1
Product	6	15	8	12	10	9
Product add	6+	15+	8+	12+	10+	9= 60

The Check Digit Modulus is 10. It passes because 60 is divisible by 10 (the remainder is 0).

Digit Add Left to Right (4)

Each character in the scanned data is assigned a numeric value (see [OCR Check Digit Multiplier on page 6-14](#)). Each value representing a character in the scanned data is multiplied by its corresponding digit in the multiplier, resulting in a product for each character in the scanned data. The sum of each individual digit in all of the products is then calculated. The check digit passes if this sum modulo Check Digit Modulus is zero.

Example:

Scanned data numeric value is 132456 (check digit is 6)

Check digit multiplier string is 123456

Digit	1	3	2	4	5	6
Multiplier	1	2	3	4	5	6
Product	1	6	6	16	25	36
Digit add	1+	6+	6+	1+6+	2+5+	3+6= 36

The Check Digit Modulus is 12. It passes because 36 is divisible by 12 (the remainder is 0).

Digit Add Right to Left (2)

Each character in the scanned data is assigned a numeric value (see [OCR Check Digit Multiplier on page 6-14](#)). The check digit multiplier is reversed in order. Each value representing a character in the scanned data is multiplied by its corresponding digit in the reversed multiplier, resulting in a product for each character in the scanned data. The sum of each individual digit in all of the products is then calculated. The check digit passes if this sum modulo Check Digit Modulus is zero.

Example:

Scanned data numeric value is 132456 (check digit is 6)

Check digit multiplier string is 123456

Digit	1	3	2	4	5	6
Multiplier	6	5	4	3	2	1
Product	6	15	8	12	10	6
Digit add	6+	1+5+	8+	1+2+	1+0+	6= 30

The Check Digit Modulus is 10. It passes because 30 is divisible by 10 (the remainder is 0).

Product Add Right to Left Simple Remainder (5)

Each character in the scanned data is assigned a numeric value (see [OCR Check Digit Multiplier on page 6-14](#)). The check digit multiplier is reversed in order. Each value representing a character in the scanned data is multiplied by its corresponding digit in the reversed multiplier, resulting in a product for each character in the scanned data. The sum of these products **except for the check digit's product** is computed. The check digit passes if this sum modulo Check Digit Modulus is equal to the check digit's product.

Example:

Scanned data numeric value is 122456 (check digit is 6)

Check digit multiplier string is 123456

Digit	1	2	2	4	5	6
Multiplier	6	5	4	3	2	1
Product	6	10	8	12	10	6
Product add	6+	10+	8+	12+	10=	46 6

The Check Digit Modulus is 10. It passes because 46 divided by 10 leaves a remainder of 6.

Digit Add Right To Left Simple Remainder (6)

Each character in the scanned data is assigned a numeric value (see [OCR Check Digit Multiplier on page 6-14](#)). The check digit multiplier is reversed in order. Each value representing a character in the scanned data is multiplied by its corresponding digit in the reversed multiplier, resulting in a product for each character in the scanned data. The sum of each individual digit in all of the products **except for the check digit's product** is then calculated. The check digit passes if this sum modulo Check Digit Modulus is equal to the check digit's product.

Example:

Scanned data numeric value is 122459 (check digit is 6)

Check digit multiplier string is 123456

Digit	1	2	2	4	5	9
Multiplier	6	5	4	3	2	1
Product	6	10	8	12	10	9
Digit add	6+	1+0+	8+	1+2+	1+0=	19 9

The Check Digit Modulus is 10. It passes because 19 divided by 10 leaves a remainder of 9.

Health Industry - HIBCC43 (9)

This is the health industry module 43 check digit standard.

Inverse OCR

Parameter # 856

Inverse OCR is white or light words on a black or dark background. Set an option for decoding inverse OCR.

- ***0 - Regular Only** - decode regular OCR (black on white) strings only.
- **1 - Inverse Only** - decode inverse OCR (white on black) strings only.
- **2 - Autodiscriminate** - decodes both regular and inverse OCR strings.

APPENDIX A SDK DEMONSTRATION PROGRAM

Software Decode SDK Demonstration Programs

The *Software Decode SDK for Android* includes a sample application that demonstrates how to interact with the Software Decode Library.

Sample Source Code

This application includes source code for use in designing systems. This source code is included for demonstration purposes only and is not intended as a commercially viable solution.

- ✓ **NOTE** This software is provided as is without any express or implied warranty of any kind including warranties of merchantability, non-infringement, or fitness for a particular purpose. In no event shall Zebra be liable for any special, indirect, or consequential damages whatsoever, resulting from the use or inability to use the software, even if Zebra has been advised of the possibility of such damages. Some jurisdictions prohibit the exclusion or limitation of liability for consequential or incidental damages; therefore this limitation may not apply to you.

Sample Application

The sample application **SDLgui** demonstrates the following common interactions with a Software Decode Library device:

- Manipulating the trigger to start/stop sessions
- Obtaining and interpreting decoded data
- Reading and writing parameters
- Obtaining and displaying snapshot images from the device
- Displaying video images from the device

Install the sample application on the device and open it. Use this application to perform the following functions:

- To decode a bar code, select the **Decode** button and present the bar code to the imager engine. A successful decode displays the decoded bar code in the **data:** text area and the device beeps if the **beep** checkbox is checked.
- For a hands-free decode, select the **Hands Free** button and present the bar code to the imager engine.
- To capture an image (take a picture), point the imager engine at the intended target and select the **Snapshot** button. A image screen appears with the acquired image. Click the image to close the window.
- To display a video image select the **Video** button. A video screen displays the live video. Click the screen to close the window.
- To enable all the symbologies, select **EnableAll**.
- To disable all the symbologies, select **DisableAll**.
- To scan a bar code from digital media, check the **Reading Pane** box.
- To get the last decoded image, set [Retrieve Last Decoded Image on page 4-12](#) (parameter #905) to 1, decode, and then select **GetDecodedImage** to decode the last image.
- To display the imager engine properties select the **Properties** button.
- Use the parameter **Get**, **Set**, and **Defaults** buttons to manipulate the Software Decode Library parameters:
 - To query the current parameter setting, enter the decimal parameter identification number in the **Number** text box and select **Get**. For example, to read the current JPEG compression quality setting, enter **305** (from [JPEG Quality Value on page 4-9](#)) in the **Number** text box. By default, **65** appears in the **Value** field after selecting **Get**.
 - To alter a parameter, enter the parameter identification number in **Number** and the value to assign in **Value** and select **Set**. For example, to increase the JPEG compression quality, enter **305** in the **Number** text box and **90** in the **Value** field and select **Set**.

If you specify an illegal (out of range) value for a parameter, an error is reported in the **status** field.

Sample Application Source Code

Sample applications reside in the directory **Applications**, within the directory selected during SDK installation. This directory includes the complete source in addition to an Eclipse project you can use to create the executable. Sample applications use Java and assume a basic understanding of this environment.

APPENDIX B PROGRAMMING REFERENCE

Symbol Code Identifiers

Table B-1 *Symbol Code Characters*

Code Character	Code Type
A	UPC-A, UPC-E, UPC-E1, EAN-8, EAN-13
B	Code 39, Code 32
C	Codabar
D	Code 128, ISBT 128, ISBT 128 Concatenated
E	Code 93
F	Interleaved 2 of 5
G	Discrete 2 of 5, or Discrete 2 of 5 IATA
H	Code 11
J	MSI
K	GS1-128
L	Bookland EAN
M	Trioptic Code 39
N	Coupon Code
R	GS1 DataBar Family
S	Matrix 2 of 5
T	UCC Composite, TLC 39
U	Chinese 2 of 5

Table B-1 *Symbol Code Characters (Continued)*

Code Character	Code Type
V	Korean 3 of 5
X	ISSN EAN, PDF417, Macro PDF417, Micro PDF417
z	Aztec, Aztec Rune
P00	Data Matrix
P01	QR Code, MicroQR
P02	Maxicode
P03	US Postnet
P04	US Planet
P05	Japan Postal
P06	UK Postal
P08	Netherlands KIX Code
P09	Australia Post
P0A	USPS 4CB/One Code/Intelligent Mail
P0B	UPU FICS Postal
P0H	Han Xin
P0X	Signature Capture

AIM Code Identifiers

Each AIM Code Identifier contains the three-character string **Jcm** where:

- J = Flag Character (ASCII 93)
- c = Code Character (see [Table B-2](#))
- m = Modifier Character (see [Table B-3](#))

Table B-2 *Aim Code Characters*

Code Character	Code Type
A	Code 39, Code 39 Full ASCII, Code 32
C	Code 128, ISBT 128, ISBT 128 Concatenated, GS1-128, Coupon (Code 128 portion)
d	Data Matrix
E	UPC/EAN, Coupon (UPC portion)
e	GS1 DataBar Family
F	Codabar
G	Code 93
H	Code 11
h	Han Xin
I	Interleaved 2 of 5
L	PDF417, Macro PDF417, Micro PDF417
L2	TLC 39
M	MSI
Q	QR Code, MicroQR
S	Discrete 2 of 5, IATA 2 of 5
U	Maxicode
z	Aztec, Aztec Rune
X	Bookland EAN, ISSN EAN, Trioptic Code 39, Chinese 2 of 5, Matrix 2 of 5, Korean 3 of 5, US Postnet, US Planet, UK Postal, Japan Postal, Australia Post, Netherlands KIX Code, USPS 4CB/One Code/Intelligent Mail, UPU FICS Postal, Signature Capture

The modifier character is the sum of the applicable option values based on [Table B-3](#).

Table B-3 *Modifier Characters*

Code Type	Option Value	Option
Code 39	0	No check character or Full ASCII processing.
	1	Reader has checked one check character.
	3	Reader has checked and stripped check character.
	4	Reader has performed Full ASCII character conversion.
	5	Reader has performed Full ASCII character conversion and checked one check character.
	7	Reader has performed Full ASCII character conversion and checked and stripped check character.
	Example: A Full ASCII bar code with check character W, A+I+MI+DW , is transmitted as J A7 AIMID where 7 = (3+4).	
Trioptic Code 39	0	No option specified at this time. Always transmit 0.
	Example: A Trioptic bar code 412356 is transmitted as J X0 412356	
Code 128	0	Standard data packet, no Function code 1 in first symbol position.
	1	Function code 1 in first symbol character position.
	2	Function code 1 in second symbol character position.
	Example: A Code (EAN) 128 bar code with Function 1 character ^{FNC1} in the first position, AIMID is transmitted as J C1 AIMID	
I 2 of 5	0	No check digit processing.
	1	Reader has validated check digit.
	3	Reader has validated and stripped check digit.
	Example: An I 2 of 5 bar code without check digit, 4123, is transmitted as J I0 4123	
Codabar	0	No check digit processing.
	1	Reader has checked check digit.
	3	Reader has stripped check digit before transmission.
	Example: A Codabar bar code without check digit, 4123, is transmitted as J F0 4123	
Code 93	0	No options specified at this time. Always transmit 0.
	Example: A Code 93 bar code 012345678905 is transmitted as J G00 12345678905	
MSI	0	Check digits are sent.
	1	No check digit is sent.
	Example: An MSI bar code 4123, with a single check digit checked, is transmitted as J M1 4123	

Table B-3 Modifier Characters (Continued)

Code Type	Option Value	Option
D 2 of 5	0	No options specified at this time. Always transmit 0.
	Example: A D 2 of 5 bar code 4123, is transmitted as JS04123	
UPC/EAN	0	Standard data packet in full EAN format, i.e. 13 digits for UPC-A, UPC-E, and EAN-13 (not including supplemental data).
	1	Two digit supplemental data only.
	2	Five digit supplemental data only.
	3	Combined data packet comprising 13 digits from EAN-13, UPC-A or UPC-E symbol and 2 or 5 digits from supplemental symbol.
	4	EAN-8 data packet.
	Example: A UPC-A bar code 012345678905 is transmitted as JE00012345678905	
Bookland EAN	0	No options specified at this time. Always transmit 0.
	Example: A Bookland EAN bar code 123456789X is transmitted as JX0123456789X	
ISSN EAN	0	No options specified at this time. Always transmit 0.
	Example: An ISSN EAN bar code 123456789X is transmitted as JX0123456789X	
Code 11	0	Single check digit
	1	Two check digits
	3	Check characters validated but not transmitted.
GS1 DataBar Family		No option specified at this time. Always transmit 0. GS1 DataBar and GS1 DataBar Limited transmit with an Application Identifier "01". Note: In GS1-128 emulation mode, GS1 DataBar is transmitted using Code 128 rules (i.e., JC1).
	Example: A GS1 DataBar bar code 0110012345678902 is transmitted as Je00110012345678902 .	
EAN.UCC Composites (GS1 DataBar, GS1-128, 2D portion of UPC composite)		Native mode transmission. Note: UPC portion of composite is transmitted using UPC rules.
	0	Standard data packet.
	1	Data packet containing the data following an encoded symbol separator character.
	2	Data packet containing the data following an escape mechanism character. The data packet does not support the ECI protocol.
	3	Data packet containing the data following an escape mechanism character. The data packet supports the ECI protocol.
		GS1-128 emulation Note: UPC portion of composite is transmitted using UPC rules.
1	Data packet is a GS1-128 symbol (i.e., data is preceded with JC1).	

Table B-3 *Modifier Characters (Continued)*

Code Type	Option Value	Option
PDF417, Micro PDF417	0	Reader set to conform to protocol defined in 1994 PDF417 symbology specifications. Note: When this option is transmitted, the receiver cannot reliably determine whether ECIs have been invoked or whether data byte 92 _{DEC} has been doubled in transmission.
	1	Reader set to follow the ECI protocol (Extended Channel Interpretation). All data characters 92 _{DEC} are doubled.
	2	Reader set for Basic Channel operation (no escape character transmission protocol). Data characters 92 _{DEC} are not doubled. Note: When decoders are set to this mode, unbuffered Macro symbols and symbols requiring the decoder to convey ECI escape sequences cannot be transmitted.
	3	The bar code contains a GS1-128 symbol, and the first codeword is 903-907, 912, 914, 915.
	4	The bar code contains a GS1-128 symbol, and the first codeword is in the range 908-909.
	5	The bar code contains a GS1-128 symbol, and the first codeword is in the range 910-911.
	Example: A PDF417 bar code ABCD, with no transmission protocol enabled, is transmitted as]L2ABCD.	
Data Matrix	0	ECC 000-140, not supported.
	1	ECC 200.
	2	ECC 200, FNC1 in first or fifth position.
	3	ECC 200, FNC1 in second or sixth position.
	4	ECC 200, ECI protocol implemented.
	5	ECC 200, FNC1 in first or fifth position, ECI protocol implemented.
	6	ECC 200, FNC1 in second or sixth position, ECI protocol implemented.
MaxiCode	0	Symbol in Mode 4 or 5.
	1	Symbol in Mode 2 or 3.
	2	Symbol in Mode 4 or 5, ECI protocol implemented.
	3	Symbol in Mode 2 or 3, ECI protocol implemented in secondary message.

Table B-3 *Modifier Characters (Continued)*

Code Type	Option Value	Option
QR Code	0	Model 1 symbol.
	1	Model 2 / MicroQR symbol, ECI protocol not implemented.
	2	Model 2 symbol, ECI protocol implemented.
	3	Model 2 symbol, ECI protocol not implemented, FNC1 implied in first position.
	4	Model 2 symbol, ECI protocol implemented, FNC1 implied in first position.
	5	Model 2 symbol, ECI protocol not implemented, FNC1 implied in second position.
	6	Model 2 symbol, ECI protocol implemented, FNC1 implied in second position.
Aztec	0	Aztec symbol.
	C	Aztec Rune symbol.

SDL Bar Code Type Identifiers

Table B-4 *SDL Bar Code Type Identifiers*

SDL Code Type ID	Bar Code Type
1	Code 39
2	Codabar
3	Code 128
4	Discrete (Standard) 2 of 5
5	IATA
6	Interleaved 2 of 5
7	Code 93
8	UPC-A
9	UPC-E0
10	EAN-8
11	EAN-13
12	Code 11
13	Code 49
14	MSI
15	EAN-128
16	UPC-E1
17	PDF-417
18	Code 16K
19	Code 39 Full ASCII
20	UPC-D
21	Code 39 Trioptic
22	Bookland
23	Coupon Code
24	NW-7
25	ISBT-128
26	Micro PDF
27	DataMatrix
28	QR Code

Table B-4 *SDL Bar Code Type Identifiers (Continued)*

SDL Code Type ID	Bar Code Type
29	Micro PDF CCA
30	PostNet US
31	Planet Code
32	Code 32
33	ISBT-128 Con
34	Japan Postal
35	Australian Postal
36	Dutch Postal
37	MaxiCode
38	Canadian Postal
39	UK Postal
40	Macro PDF
41	Macro QR
44	Micro QR
45	Aztec
46	Aztec Rune
48	GS1 DataBar-14
49	GS1 DataBar Limited
50	GS1 DataBar Expanded
52	USPS 4CB
53	UPU 4State
54	ISSN
55	Scanlet
56	CueCode
57	Matrix 2 of 5
72	UPC-A + 2 Supplemental
73	UPC-E0 + 2 Supplemental
74	EAN-8 + 2 Supplemental
75	EAN-13 + 2 Supplemental
80	UPC-E1 + 2 Supplemental
81	CCA EAN-128

Table B-4 *SDL Bar Code Type Identifiers (Continued)*

SDL Code Type ID	Bar Code Type
82	CCA EAN-13
83	CCA EAN-8
84	CCA GS1 DataBar Expanded
85	CCA GS1 DataBar Limited
86	CCA GS1 DataBar-14
87	CCA UPC-A
88	CCA UPC-E
89	CCC EAN-128
90	TLC-39
97	CCB EAN-128
98	CCB EAN-13
99	CCB EAN-8
100	CCB GS1 DataBar Expanded
101	CCB GS1 DataBar Limited
102	CCB GS1 DataBar-14
103	CCB UPC-A
104	CCB UPC-E
105	Signature Capture
114	Chinese 2 of 5
115	Korean 3 of 5
136	UPC-A + 5 supplemental
137	UPC-E0 + 5 supplemental
138	EAN-8 + 5 supplemental
139	EAN-13 + 5 supplemental
144	UPC-E1 + 5 Supplemental
154	Macro Micro PDF
180	GS1 Databar Coupon
183	Han Xin

Decode Data

If the decoded data contains more structure than can be presented in the standard format, the **Bar Code Type** field is set to 0x99 and the decode data is formatted into packets. The first byte of the **Decode Data** field contains the actual bar code type, the second byte contains the number of packets, and the remaining data is the packeted form of decode data. For example, a packeted Decode Data message for Micro PDF417 would have the following format, where the **Decode Data** field is broken out as follows:

Bar Code Type	Decode Data Field							
	Actual Bar code Type	# of Packets	Spare Byte	Byte Length of Packet #1	Data	Spare Byte	Byte Length of Packet #2	Data
0x99	1A	2	0	00 03	ABC	0	00 04	DEFG

Note that the **Packet Length** subfields consist of two bytes, where the first byte represents the high value of length x 256.

APPENDIX C SIGNATURE CAPTURE CODE

Introduction

CapCode, a signature capture code, is a special pattern that encloses a signature area on a document and allows a scanner to capture a signature.

There are several accepted patterns that allow automatic identification of different signatures on the same form. For example, on the federal tax return 1040 form there are three signature areas, one each for two joint filers, and one for a professional preparer. By using different patterns, a program can correctly identify all three, so they can be captured in any sequence and still be identified correctly.

Code Structure

Signature Capture Area

A CapCode is printed as two identical patterns on either side of a signature capture box, as shown in [Figure C-1](#). Each pattern extends the full height of the signature capture box.

The box is optional, so you can omit it, replace it with a single baseline, or print a baseline with an "X" on top of it towards the left, as is customarily done in the US to indicate a request for signature. However, if an "X" or other markings are added in the signature box area, these are captured with the signature.



Figure C-1 CapCode

CapCode Pattern Structure

A CapCode pattern structure consists of a start pattern followed by a separator space, a signature capture box, a second separator space, and then a stop pattern. Assuming that X is the dimension of the thinnest element, the start and stop patterns each contains 9X total width in 4 bars and 3 spaces. A 7X quiet zone is required to the left and to the right of the CapCode pattern.

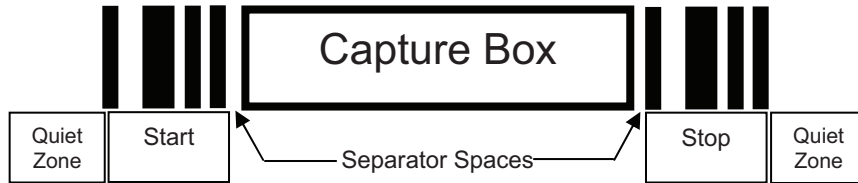


Figure C-2 CapCode Structure

The separator spaces on either side of the signature capture box can be between 1X and 3X wide.

Start / Stop Patterns

[Table C-1](#) lists the accepted start / stop patterns. The bar and space widths are expressed as multiples of X. You must use the same pattern on either side of a signature capture box. The type value is reported with the captured signature to indicate the purpose of the signature captured.

Table C-1 Start / Stop Pattern Definitions

Bar/Space Patterns							Type
B	S	B	S	B	S	B	
1	1	2	2	1	1	1	2
1	2	2	1	1	1	1	5
2	1	1	2	1	1	1	7
2	2	1	1	1	1	1	8
3	1	1	1	1	1	1	9

[Table C-2](#) lists selectable parameters used to generate the image of the captured signature.

Table C-2 *User Defined CapCode Parameters*

Parameter	Defined
Width	Number of pixels
Height	Number of pixels
Format	JPEG, BMP, TIFF
JPEG quality	1 (most compression) to 100 (best quality)
Bits Per Pixel (not applicable to JPEG format)	1 (2 levels)
	4 (16 levels)
	8 (256 levels)

BMP format does not use compression, JPEG and TIFF formats do.

Dimensions

The size of the signature capture box is determined by the height and separation of the start and stop patterns. The line width of the signature capture box is insignificant.

The thinnest element width, referred to here as X, is nominally 10 mils (1 mil = 0.0254 mm). Select this as an exact multiple of the pixel pitch of the printer used. For example, when using a 203 DPI (dots-per-inch) printer and printing 2 dots per module, the resulting X dimension is 9.85 mils.

Data Format

The scanner output is formatted according to [Table C-3](#). Zebra scanners allow different user options to output or inhibit bar code type. Selecting "Symbol ID" as the bar code type for output identifies the CapCode with letter "i".

Table C-3 *Data Format*

File Format (1 byte)	Type (1 byte)	Image Size (4 bytes, BIG Endian)	Image Data
JPEG - 1 BMP - 3 TIFF - 4	See Table C-1 , last column		(Same bytes as in a data file)

Additional Capabilities

Regardless of how the signature is captured, the output signature image is de-skewed and right-side up.

A scanner that captures signatures automatically determines whether it is scanning a signature or a bar code. You can disable the signature capturing capability in a scanner.

Signature Boxes

Figure C-3 illustrates the five acceptable signature boxes:

Type 2:



Type 5:



Type 7:



Type 8:



Type 9:



Figure C-3 *Acceptable Signature Boxes*

INDEX

Numerics

2D parameters	
aztec	5-34
aztec inverse	5-34
code 128 emulation	5-32
data matrix	5-33
data matrix inverse	5-33
decode mirror images	5-33
han xin	5-35
han xin inverse	5-35
maxicode	5-33
microPDF417	5-32
microQR	5-34
PDF417	5-32
QR code	5-34
QR inverse	5-34

A

AIM code identifiers	B-3
aiming options	
decode aiming pattern	4-6
hands-free decode aiming pattern	4-6
snapshot aiming pattern	4-7
snapshot mode timeout	4-7
video scaling	4-12
video view finder	4-12
aiming pattern	4-7
enabling	4-6
enabling hands-free	4-6
API	
configuration	3-1
sample file	3-2
aztec	5-34

B

bar codes	
decoding	2-4
block diagram	1-2
build environment	1-10

C

chapter descriptions	xi
Chinese 2 of 5	5-25
codabar parameters	5-23
CLSI editing	5-23
lengths	5-23
NOTIS editing	5-23
code 11 parameters	5-19
check digit verification	5-19
lengths	5-19
transmit check digits	5-19
code 128 parameters	5-13
check ISBT table	5-15
emulation	5-32
GS1-128	5-14
ignore fnc4	5-16
ISBT 128	5-14
ISBT concatenation	5-15
ISBT concatenation redundancy	5-15
lengths	5-14
reduced quiet zone	5-15
code 39 parameters	5-16
check digit verification	5-17
code 32 prefix	5-17
convert to code 32	5-16
full ASCII	5-18
lengths	5-17
reduced quiet zone	5-18
transmit check digit	5-17

- trioptic 5-16
- code 93 parameters 5-18
 - lengths 5-18
- code ID character
 - AIM B-3
 - modifiers B-4
 - Symbol B-1
 - transmitting 4-4
- code identifiers
 - AIM B-3
 - modifier characters B-4
 - SDL B-8
 - Symbol B-1
- composite parameters
 - composite CC-A/B 5-31
 - composite CC-C 5-31
 - composite TLC-39 5-31
 - UPC composite mode 5-31
- conventions
 - notational xii
- cropping 4-7, 4-8

D

- data matrix 5-33
- decode mode 1-3
- decoding bar codes 2-4
- default parameters 4-1, 5-1
 - OCR 6-2
- demonstration programs A-1
 - sample application A-2
 - sample source code A-1, A-2
- directory structure 1-5
- disabling symbologies 2-5
- discrete 2 of 5 parameters 5-22
 - lengths 5-22

E

- enabling symbologies 2-5
- engines supported 1-4
- exposure options
 - decoding illumination 4-6
 - illumination power level 4-6
 - image capture illumination 4-7
 - motion illumination 4-7

F

- file structure 1-5

G

- get decode count 2-5

- get last decoded image 2-5
- GS1 databar
 - convert GS1 databar to UPC/EAN 5-30
 - GS1 databar expanded 5-30
 - GS1 databar limited 5-29
 - GS1 databar limited security level 5-30
 - GS1 databar-14 5-29

H

- han xin 5-35

I

- illumination
 - decoding 4-6
 - image capture 4-7
 - motion 4-7
 - power level 4-6
- image capture 1-2
- image cropping 4-7, 4-8
- image enhancement 4-9
- image frames 2-3
- image options
 - bits per pixel 4-10, 4-11
 - cropping 4-7, 4-8
 - file formats 4-9, 4-11
 - image enhancement 4-9
 - image resolution 4-9
 - JPEG size/quality 4-9
- image resolution 4-9
- imager engine 1-2
- imager engines supported 1-4
- information, service xiii
- initialization 2-1
- installation 1-5
 - initialization 2-1
 - scanner connection 2-1
 - setting up build environment 1-10
- interleaved 2 of 5 parameters 5-20
 - check digit verification 5-20
 - convert to EAN-13 5-21
 - lengths 5-20
 - reduced quiet zone 5-21
 - security level 5-21
 - transmit check digit 5-21

J

- JPEG image options
 - size/quality 4-9

K

Korean 3 of 5 5-25

M

matrix 2 of 5 parameters 5-26
 check digit 5-26
 lengths 5-26
 redundancy 5-26
 transmit check digit 5-27
 maxicode 5-33
 memory 1-4
 modes of operation 1-3
 MSI parameters 5-24
 check digit algorithm 5-25
 check digits 5-24
 lengths 5-24
 transmit check digit 5-25

N

no read message 4-4
 notational conventions xii

O

OCR
 default parameters 6-2
 parameters 6-3
 operating modes 1-3
 operating system 1-4
 overview 1-1

P

parameters 2-2, 4-1, 5-1
 Australia post 5-28
 Australia post format 5-28
 aztec 5-34
 aztec inverse 5-34
 bar codes to read 4-5
 bits per pixel 4-10, 4-11
 bookland EAN 5-8
 bookland ISBN 5-12
 check ISBT table 5-15
 Chinese 2 of 5 5-25
 codabar 5-23
 codabar CLSI editing 5-23
 codabar lengths 5-23
 codabar NOTIS editing 5-23
 code 11 5-19
 code 11 check digit verification 5-19
 code 11 check digits 5-19

code 11 lengths 5-19
 code 128 5-13
 code 128 emulation 5-32
 code 128 lengths 5-14
 code 128 reduced quiet zone 5-15
 code 32 prefix 5-17
 code 39 5-16
 code 39 check digit verification 5-17
 code 39 full ASCII 5-18
 code 39 lengths 5-17
 code 39 reduced quiet zone 5-18
 code 39 transmit check digit 5-17
 code 93 5-18
 code 93 lengths 5-18
 composite CC-A/B 5-31
 composite CC-C 5-31
 composite TLC-39 5-31
 convert code 39 to code 32 5-16
 convert GS1 databar to UPC/EAN 5-30
 convert I 2 of 5 to EAN-13 5-21
 convert UPC-E to UPC-A 5-11
 convert UPC-E1 to UPC-A 5-12
 coupon report 5-13
 crop to address 4-8
 data matrix 5-33
 data matrix inverse 5-33
 decode aiming pattern 4-6
 decode mirror images 5-33
 decode session timeout 4-3
 decoding illumination 4-6
 defaults 4-1, 5-1
 discrete 2 of 5 5-22
 discrete 2 of 5 lengths 5-22
 EAN zero extend 5-12
 EAN-13/JAN-13 5-7
 EAN-8/JAN-8 5-7
 full read mode 4-5
 fuzzy 1D processing 4-4
 GS1 databar expanded 5-30
 GS1 databar limited 5-29
 GS1 databar limited security 5-30
 GS1 databar-14 5-29
 GS1-128 5-14
 han xin 5-35
 han xin inverse 5-35
 hands-free decode aiming pattern 4-6
 I 2 of 5 check digit verification 5-20
 I 2 of 5 convert to EAN-13 5-21
 I 2 of 5 reduced quiet zone 5-21
 I 2 of 5 security level 5-21
 I 2 of 5 transmit check digit 5-21
 ignore code 128 fnc4 5-16
 illumination power level 4-6
 image capture illumination 4-7

- image cropping 4-7
- image enhancement 4-9
- image file format 4-9, 4-11
- image resolution 4-9
- intercharacter gap size 5-36
- interleaved 2 of 5 5-20
- interleaved 2 of 5 lengths 5-20
- inverse 1D 5-27
- inverse OCR 6-17
- ISBT 128 5-14
- ISBT concatenation 5-15
- ISBT concatenation redundancy 5-15
- ISSN EAN 5-13
- Japan postal 5-28
- JPEG quality and size 4-9
- Korean 3 of 5 5-25
- matrix 2 of 5 5-26
- matrix 2 of 5 check digit 5-26
- matrix 2 of 5 lengths 5-26
- matrix 2 of 5 redundancy 5-26
- matrix 2 of 5 transmit check digit 5-27
- maxicode 5-33
- MICR E13B 6-5
- microPDF417 5-32
- microQR 5-34
- mobile phone/display mode 4-5
- motion illumination 4-7
- MSI 5-24
- MSI check digit algorithm 5-25
- MSI check digits 5-24
- MSI lengths 5-24
- MSI transmit check digit 5-25
- multi decode mode 4-5
- Netherlands KIX code 5-29
- OCR 6-3
- OCR check digit modulus 6-13
- OCR check digit multiplier 6-14
- OCR check digit validation 6-15
- OCR lines 6-6
- OCR maximum characters 6-6
- OCR minimum character 6-6
- OCR quiet zone 6-7
- OCR subset 6-7
- OCR template 6-8
- OCR-A 6-3
- OCR-A variant 6-3
- OCR-B 6-4
- OCR-B variant 6-4
- orientation 6-6
- PDF417 5-32
- picklist mode 4-3
- QR code 5-34
- QR inverse 5-34
- quiet zone level 5-36
- redundancy level 5-35
- retrieve last decoded image 4-12
- security level 5-36
- signature capture 4-10
- signature capture height 4-11
- signature capture JPEG quality 4-12
- signature capture width 4-11
- snapshot aiming pattern 4-7
- snapshot mode timeout 4-7
- supplementals 5-8
- symbology 5-1
- symbology defaults 5-1
- timeout between decodes 4-4
- transmit code ID character 4-4
- transmit no read message 4-4
- transmit UK postal check digit 5-28
- transmit US postal check digit 5-27
- trigger mode 4-3
- trioptic code 39 5-16
- UCC coupon extended code 5-12
- UK postal 5-28
- UPC composite mode 5-31
- UPC reduced quiet zone 5-10
- UPC/EAN/JAN supplemental AIM ID format 5-9
- UPC/EAN/JAN supplemental redundancy 5-9
- UPC-A 5-7
- UPC-A check digit 5-10
- UPC-A preamble 5-11
- UPC-E 5-7
- UPC-E check digit 5-10
- UPC-E preamble 5-11
- UPC-E1 5-7
- UPC-E1 check digit 5-10
- UPC-E1 preamble 5-11
- UPU FICS postal 5-29
- US currency serial number 6-6
- US planet 5-27
- US postnet 5-27
- user-programmable supplementals 5-9
- USPS 4CB/One Code/Intelligent Mail 5-29
- video scaling 4-12
- video view finder 4-12
- PDF417 5-32
- picklist mode 4-3
- postal codes
 - Australia post 5-28
 - Australia post format 5-28
 - Japan postal 5-28
 - Netherlands KIX code 5-29
 - transmit UK postal check digit 5-28
 - transmit US postal check digit 5-27
 - UK postal 5-28
 - UPU FICS postal 5-29
 - US planet 5-27

- US postnet 5-27
 - USPS 4CB/One Code/Intelligent Mail 5-29
 - processor 1-4
 - properties 2-2
- Q**
- QR code 5-34
- R**
- RAM 1-4
 - related documents xii
- S**
- sample application A-2
 - sample source code A-1, A-2
 - scanner
 - configuration 2-2
 - connecting 2-1
 - decoding bar codes 2-4
 - parameters 2-2
 - properties 2-2
 - SDL bar code identifiers B-8
 - security
 - intercharacter gap size 5-36
 - quiet zone level 5-36
 - redundancy level 5-35
 - security level 5-36
 - service information xiii
 - shutdown 2-5
 - signature capture 4-10
 - area C-1
 - bits per pixel 4-11
 - boxes C-4
 - dimensions C-3
 - file format selector 4-11
 - height 4-11
 - JPEG quality 4-12
 - parameters 4-10
 - width 4-11
 - signature capture code C-1
 - boxes C-4
 - capabilities C-4
 - code structure C-1
 - data format C-3
 - dimensions C-3
 - pattern structure C-2
 - start and stop patterns C-2
 - snapshot mode 1-3, 2-5
 - aiming pattern 4-7
 - timeout 4-7
 - solution diagram 1-2
 - supportxiii
 - Symbol code identifiers B-1
 - sympologies
 - defaults 5-1
 - enabling and disabling 2-5
 - sympologies supported 1-3
 - system configuration 3-1
 - sample file 3-2
 - system requirements 1-4
- T**
- theory of operation 1-2
 - trigger modes
 - selecting 4-3
- U**
- UPC/EAN parameters
 - bookland EAN 5-8
 - bookland ISBN 5-12
 - check digit 5-10
 - convert UPC-E to UPC-A 5-11
 - convert UPC-E1 to UPC-A 5-12
 - coupon report 5-13
 - EAN zero extend 5-12
 - EAN-13/JAN-13 5-7
 - EAN-8/JAN-8 5-7
 - ISSN EAN 5-13
 - reduced quiet zone 5-10
 - supplemental AIM ID format 5-9
 - supplemental redundancy 5-9
 - supplementals 5-8
 - UCC coupon extended code 5-12
 - UPC-A 5-7
 - UPC-A preamble 5-11
 - UPC-E 5-7
 - UPC-E preamble 5-11
 - UPC-E1 5-7
 - UPC-E1 preamble 5-11
 - user-programmable supplementals 5-9
- V**
- video capture 2-5
 - video mode 1-3, 2-5
 - scaling 4-12
 - view finder 4-12

Tell Us What You Think...

We'd like to know what you think about this Manual. Please take a moment to fill out this questionnaire and fax this form to: (631) 627-7184, or mail to:

Zebra Technologies Corporation
Lincolnshire, IL U.S.A.
Attention: Technical Publications Manager
Data Capture Solutions

IMPORTANT: If you need product support, please call the appropriate customer support number provided. Unfortunately, we cannot provide customer support at the fax number above.

Manual Title: _____
(please include revision level)

How familiar were you with this product before using this manual?

- Very familiar Slightly familiar Not at all familiar

Did this manual meet your needs? If not, please explain.

What topics need to be added to the index, if applicable?

What topics do you feel need to be better discussed? Please be specific.

What can we do to further improve our manuals?



Zebra Technologies Corporation
Lincolnshire, IL U.S.A.
<http://www.zebra.com>

Zebra and the stylized Zebra head are trademarks of ZIH Corp., registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners.

©2016 Symbol Technologies LLC, a subsidiary of Zebra Technologies Corporation. All rights reserved.

