



LEXICON LE45
SCANWEDGE
USER GUIDE

TABLE OF CONTENTS



3 OVERVIEW

- FEATURE SUMMARY
- CUSTOMIZE OPERATION FEEDBACK
- FORMAT SCANNED DATA
- CUSTOMIZE OUTPUT BEHAVIOR
- SETUP SCAN ENGINE FEATURES

4 GETTING STARTED

- START SCANWEDGE
- ENABLE/DISABLE SCANWEDGE

5 CUSTOMIZE OPERATION FEEDBACK

- ENABLE/DISABLE AUDIO FEEDBACK
- ENABLE/DISABLE HAPTIC FEEDBACK

6 FORMAT SCANNED DATA

- SWITCH INPUT FROM PHYSICAL KEYPAD TO VIRTUAL KEYBOARD

7 SWITCH INPUT FROM VIRTUAL KEYBOARD TO PHYSICAL KEYPAD

8 ADD A PREFIX STRING

- ADD A SUFFIX STRING

9 INTERPRETING ESC CHARACTER IN PREFIX/SUFFIX

10 REPLACE CHARACTER IN THE SCANNED DATA BY A PREDEFINED CHARACTER (BASIC)

11 REPLACE STRING IN THE SCANNED DATA BY PREDEFINED STRINGS (ADVANCED)

13 CUSTOMIZE OUTPUT BEHAVIOR

- SEND SCANNED DATA TO KEYBOARD OR CLIPBOARD

- COPY SCANNED DATA TO CLIPBOARD

14 SEND KEY EVENT BEFORE PREFIX

15 SEND KEY EVENT AFTER SUFFIX

16 SEND ENTER KEY AFTER SUFFIX

- SEND TAB KEY AFTER SUFFIX

- SEND KEY CODE 0 AFTER SUFFIX

17 NOTIFY OTHER APPLICATIONS VIA INTENT BROADCAST

18 SEND BARCODE CODE ID ALONG WITH THE SCANNED DATA

19 SETUP SCAN ENGINE

- ENABLE / DISABLE MOBILE PHONESCANNING MODE

- SCAN ONLY BARCODE LOCATED AT THE CENTER OF DECODING AREA

20 SETUP SCAN ENGINE DECODERS

21 APPENDIX

- TABLE 1: SYMBOL CODE IDENTIFIERS

22 TABLE 1.1: AIM CODE IDENTIFIERS

- TABLE 2: CODE CHARACTERS

23 TABLE 2.1: MODIFIER CHARACTERS

27 TABLE 3: ASCII CODE

OVERVIEW



ScanWedge is a convenient and powerful data capture, processing, and output tool for end users or developers to handle captured data from barcodes. The tool is based on the concept of a zero-code approach. When an application screen is displayed in the foreground, ScanWedge automatically detects associated activity and applies the settings to the captured data.

FEATURE SUMMARY

The ScanWedge utility offers the following customizations to scan operation/scanned data:

- Customize operation feedback
- Format scanned data
- Customize output behavior
- Setup scan engine features

CUSTOMIZE OPERATION FEEDBACK

The utility allows users / developers to:

- Enable / disable audio (beeper) feedback after a successful barcode scan
- Enable / disable haptic (vibration) feedback after a successful barcode scan

FORMAT SCANNED DATA

ScanWedge provides the following facilities to format the scanned data:

- Add prefix string
- Add suffix string
- Interpret escape & control character within prefix / suffix string
- Replace character in the scanned data by a predefined character
- Replace characters / string in multiple rule base manner

CUSTOMIZE OUTPUT BEHAVIOR

Users can customize output behavior to scanned data:

- Send scanned data to keyboard or clipboard
- Copy scanned data to clipboard
- Send custom key event before / after the prefix / suffix
- Send ENTER key after suffix
- Send TAB key after suffix
- Send Key code 0 after suffix
- Notify other applications via Intent Broadcast
- Send barcode code ID, AIM and Symbol type, along with the scanned data

SETUP SCAN ENGINE FEATURES

Some scan engine scanning features can be customized in the utility:

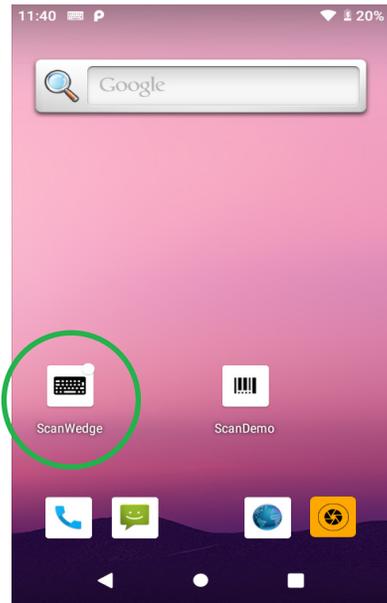
- Enable / disable mobile phone scanning mode
- Scan only barcodes located at the center of the decoding area
- Setup scan engine decoder

GETTING STARTED



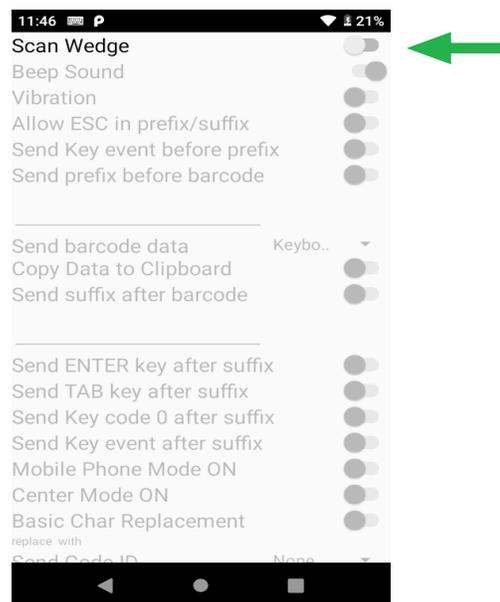
START SCANWEDGE

Launch ScanWedge from the Desktop



ENABLE/DISABLE SCANWEDGE

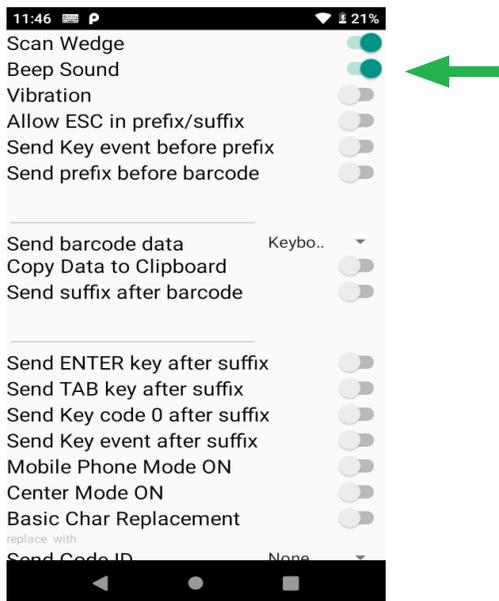
You can turn on the ScanWedge switch in the ScanWedge App



CUSTOMIZE OPERATION FEEDBACK

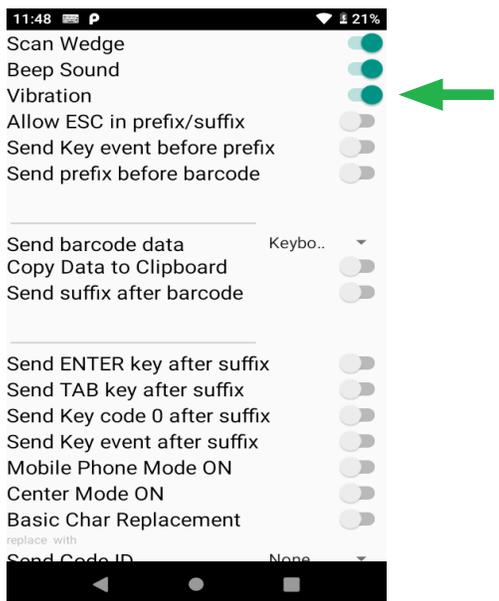
ENABLE/DISABLE AUDIO FEEDBACK

You can enable / disable beeper sound during the scan operations. To do this, turn on (green) or off (gray) the Beep Sound switch as shown below.



ENABLE/DISABLE HAPTIC FEEDBACK

You can enable / disable the vibrator during the scan operations. To do this, turn on (green) or off (gray) the Vibration switch as shown below.



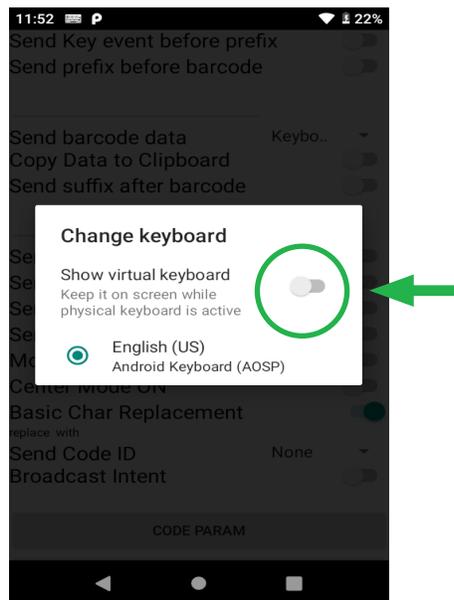
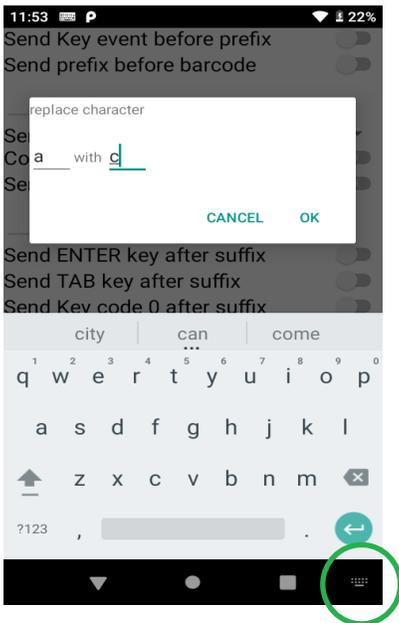
FORMAT SCANNED DATA

SWITCH INPUT FROM PHYSICAL KEYPAD TO VIRTUAL KEYBOARD

Users are allowed to switch the character / symbol input from physical keypad to virtual keyboard. The following steps apply to all input fields throughout the ScanWedge utility.

Use the character replacement dialog box as an example:

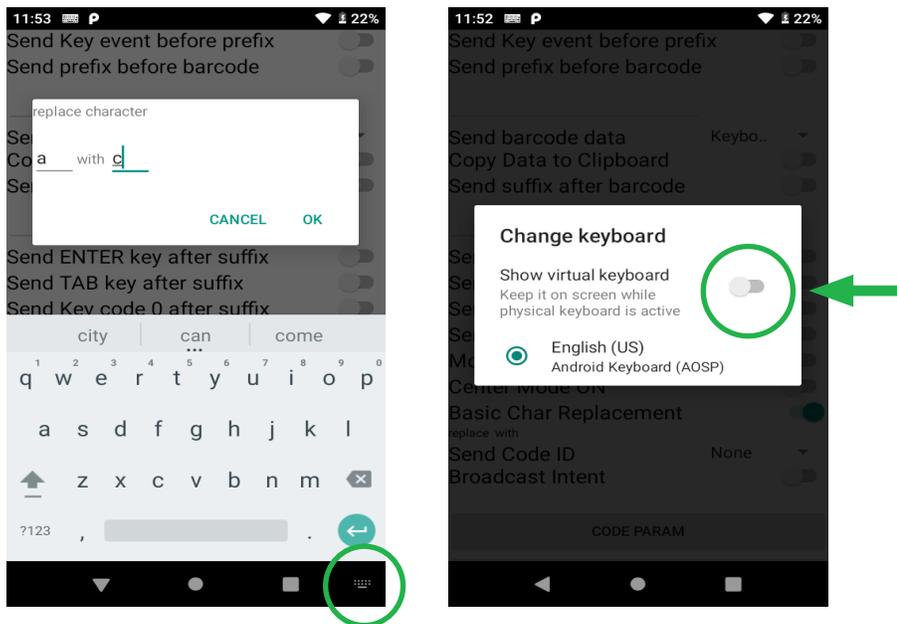
1. Tap on the input field where you want to switch the input from physical keypad to virtual keyboard.
2. A keyboard icon will be shown at the bottom right corner of the screen.
3. Tap on the keyboard icon.
4. A Change keyboard dialog box pops up.
5. Turn ON the SHOW VIRTUAL KEYBOARD switch.
6. Now you can input any key available in that virtual keyboard.



SWITCH INPUT FROM VIRTUAL KEYBOARD TO PHYSICAL KEYPAD

To switch the character / symbol input from virtual keyboard to physical keypad:

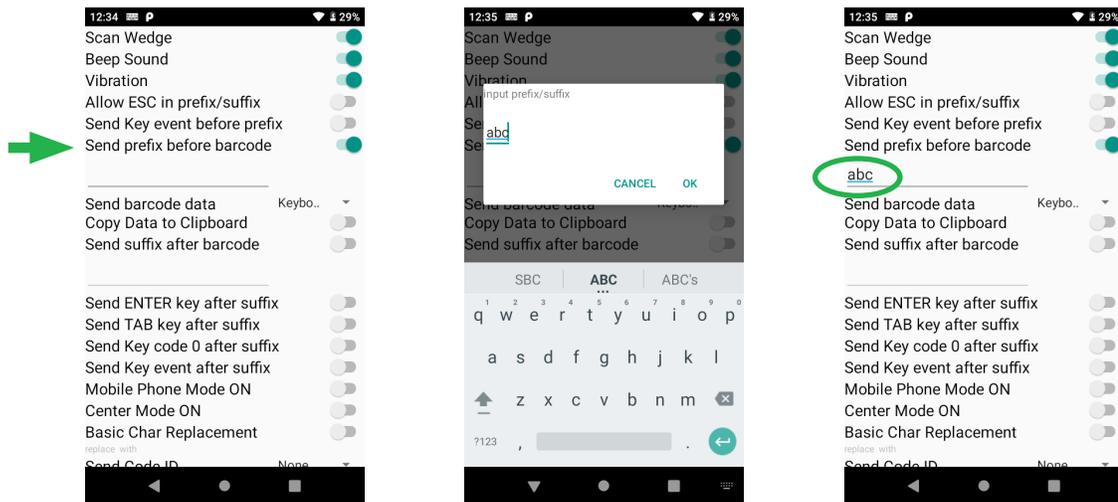
1. Tap on the input field where you want to switch the input from virtual keyboard to physical keypad.
2. A keyboard icon will be shown at the bottom right corner of the screen.
3. Tap on the keyboard icon.
4. A change keyboard dialog box pops up.
5. Turn OFF the SHOW VIRTUAL KEYBOARD switch.
6. You may now input any key through the physical keypad.



ADD A PREFIX STRING

To add a prefix to scanned data:

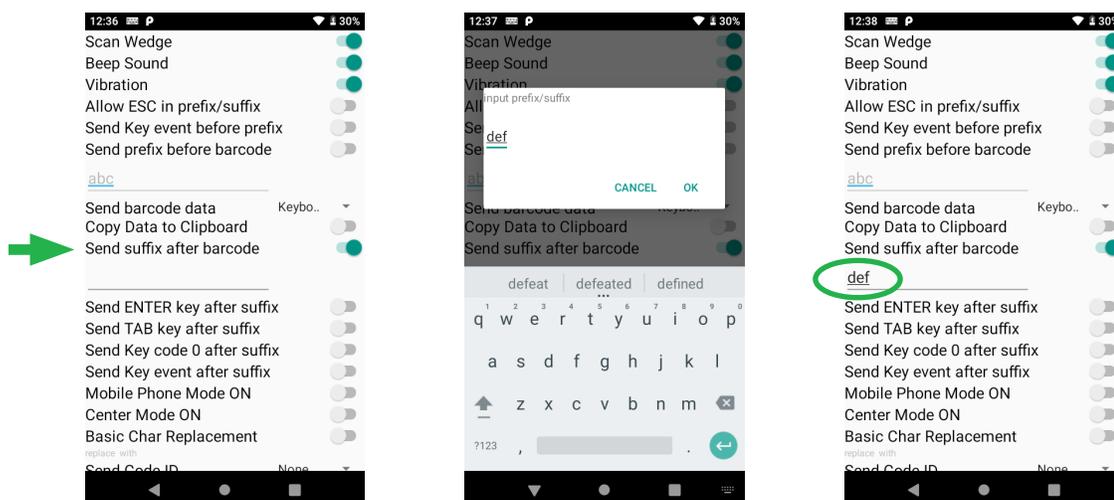
1. In the ScanWedge utility, turn on the SEND PREFIX BEFORE BARCODE switch.
2. A dialog box pops up.
3. Input the prefix string in the box and click OK.
4. Make sure the prefix string is shown in the utility.



ADD A SUFFIX STRING

To add a suffix to scanned data:

1. In the ScanWedge utility, turn on the SEND PREFIX BEFORE BARCODE switch.
2. A dialog box pops up.
3. Input the prefix string in the box and click OK.
4. Make sure the prefix string is shown in the utility.



INTERPRETING ESC CHARACTER IN PREFIX/SUFFIX

The ScanWedge utility can interpret escape characters and the following characters in the sequence and in the suffix & prefix string. It then invokes the corresponding operations when sending the prefix & suffix string to the designated output.

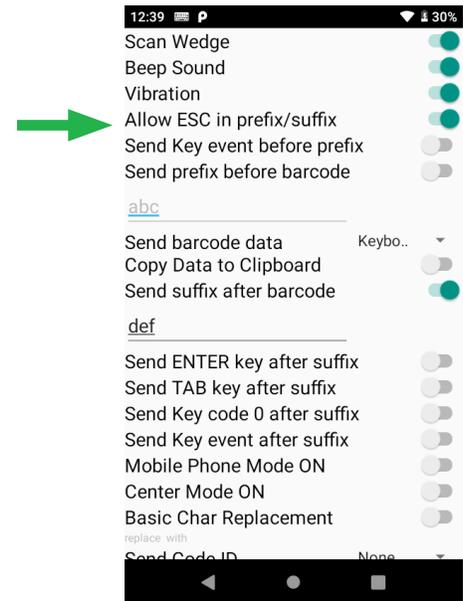
Typical example of escape character operations are:

\` single quote
\" double quote
\\ backslash
\n new line
\r carriage return
\t tab
\b backspace
\f form feed

To enable this feature:

1. In the ScanWedge utility, turn on (green) the ALLOW ESC in prefix / suffix switch.
2. Make sure there is ESC control code in prefix/suffix string.

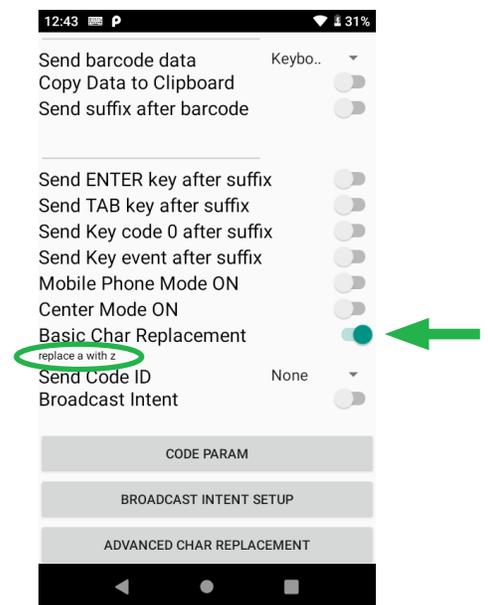
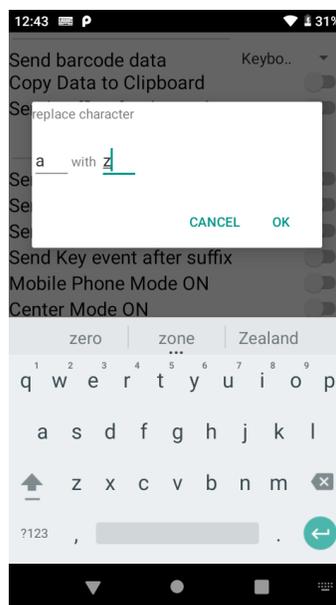
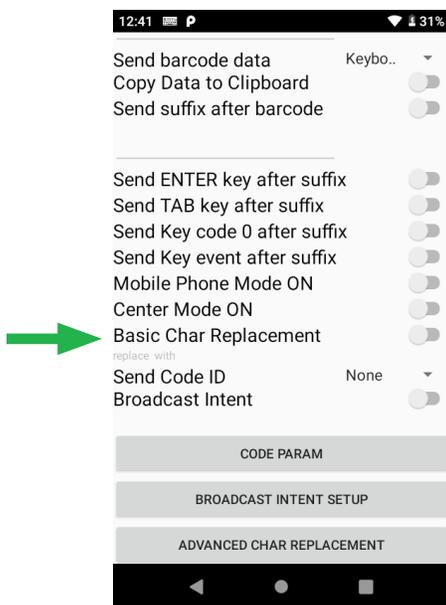
Note that this feature is enabled only when Prefix String/ Suffix String are present AND with the correct sequence control character.



REPLACE CHARACTER IN THE SCANNED DATA BY A PREDEFINED CHARACTER (BASIC)

The ScanWedge utility allows an application to replace a character with a specified character within the scanned data. To do this:

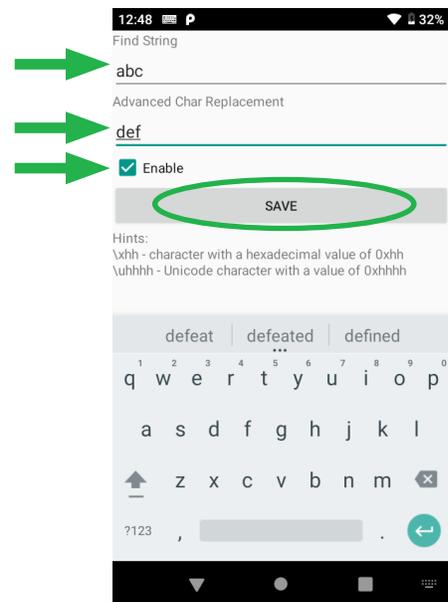
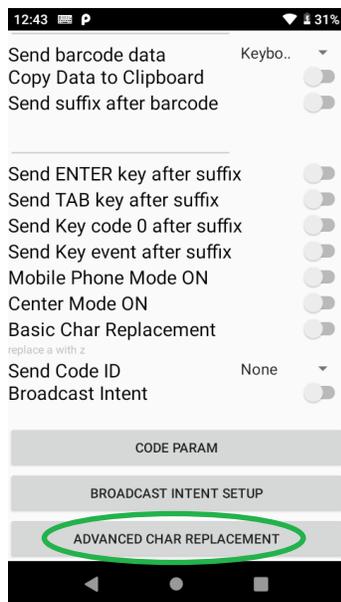
1. In the ScanWedge utility, tap on the BASIC CHAR REPLACEMENT switch.
2. A dialog box pops up.
3. Input the character to be replaced and the substitute character in the box, click OK.
4. Make sure the switch is ON (green).
5. The character replacement rule is shown below BASIC CHAR REPLACEMENT. Note that this feature is enabled only when Prefix String/Suffix String are present AND with the correct sequence control character.



REPLACE STRING IN THE SCANNED DATA BY PREDEFINED STRINGS (ADVANCED)

The ScanWedge utility allows an application to replace a string with specified strings within the scanned data. Currently, the utility supports up to 10 replacement rules. These rules will be applied to the decoded scanned data unless the rules are EMPTY. To do this:

1. In the ScanWedge utility, tap on the ADVANCED CHAR REPLACEMENT button located at the bottom of the page.
2. You will be taken to the rules definition screen.
3. Tap on the 1st [EMPTY] button. For example, REPLACE STRING RULE 1 button in the image below.
4. You will be taken to the string replacement rule screen.
5. Input the string to be replaced in FIND STRING.
6. Input the replacement string in ADVANCED CHARACTER REPLACEMENT
7. Make sure the ENABLE checkbox is checked.
8. Tap on the SAVE button to save the rule.



NOTE: Below is the Find String rules supported by the utility:

- X – checks for the character “x”
- \xhh – checks for the character with a hexadecimal value of 0xhh
- \uhhhh – checks for the Unicode character with a value 0xhhhh

For example, if a user wants to replace Group Separator (i.e. 0x29) to the character “[”, the

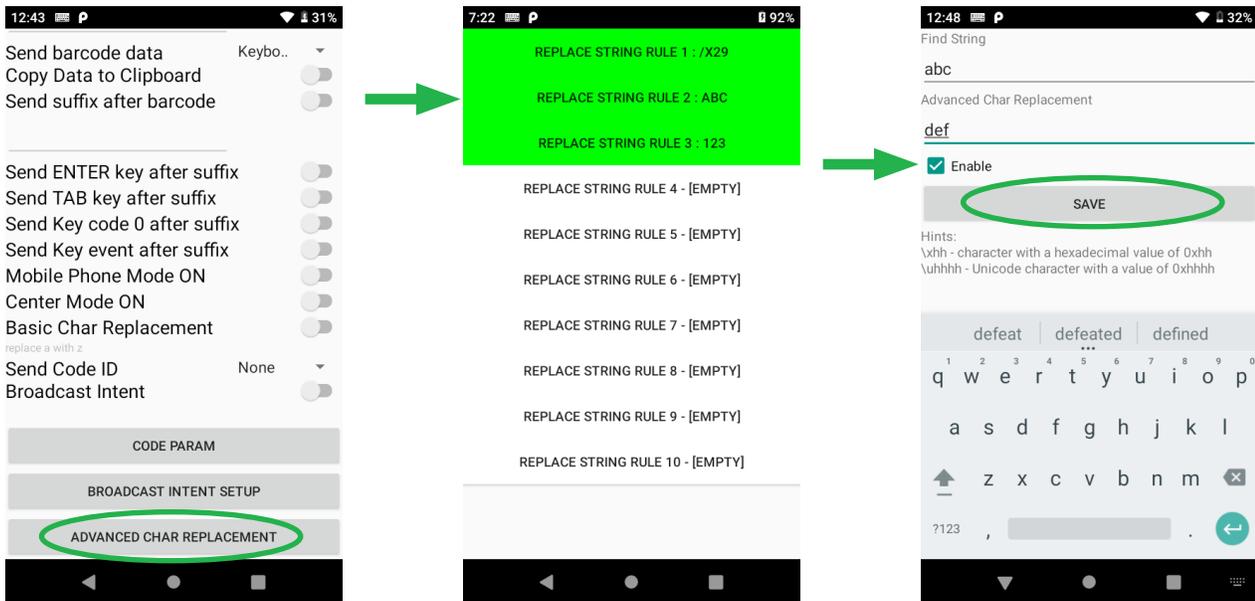
definition of rule will look like the above screen capture.

To make the deployment more flexible according to the needs of each individual site, the character replacement rules can be enabled or disabled individually. Enabled, disabled, and non-defined rules can easily be identified individually in the rules definition screen:

- All active rules are highlighted in light green
- All inactive rules are highlighted in red
- All non-defined rules are in white

For example, to disable Rule 2:

- In the ScanWedge utility, tap on the ADVANCED CHAR REPLACEMENT button located at the page bottom
- It jumps to the rules definition screen. Current active rules are highlighted in light green.
- Tap on REPLACE STRING RULES 2.
- Uncheck the ENABLE checkbox.
- Tap on the SAVE button.
- In the definition screen, the inactive rule is highlighted in red.

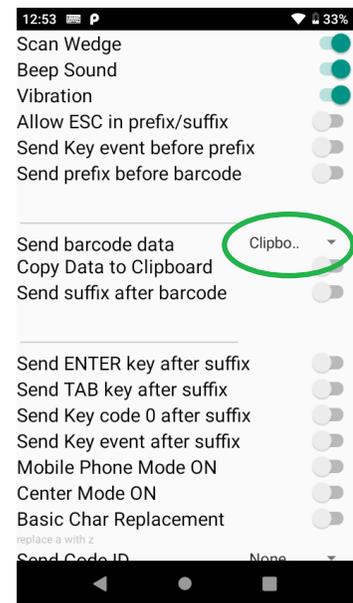
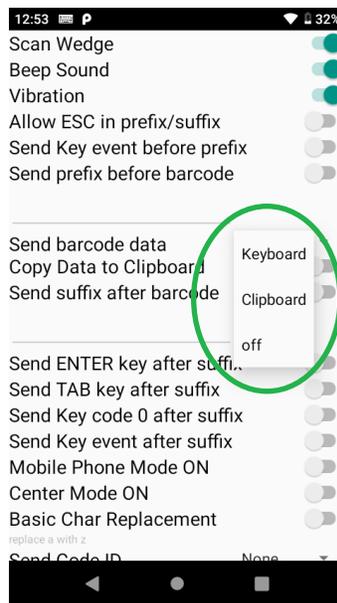
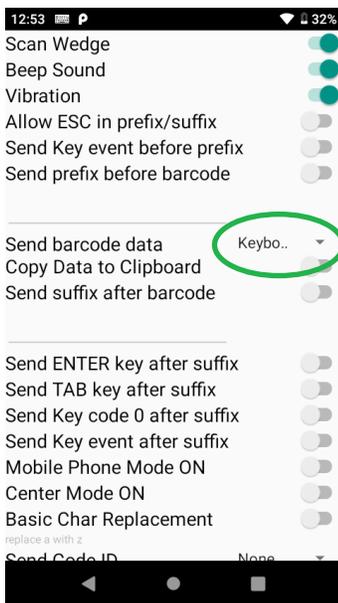


CUSTOMIZE OUTPUT BEHAVIOR

SEND SCANNED DATA TO KEYBOARD OR CLIPBOARD

The ScanWedge utility allows users/developers to designate scanned data to either Keyboard or Clipboard, according to the application need. To do this:

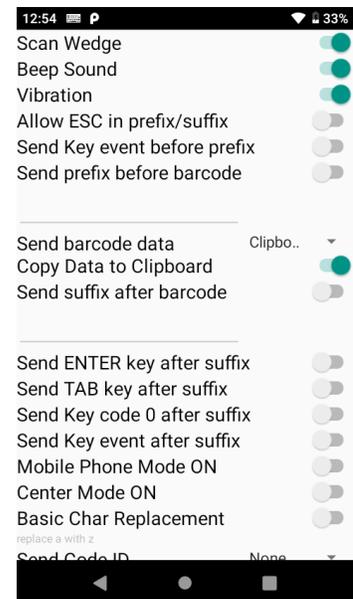
1. In the ScanWedge utility, the default output is KEYBOARD.
2. Tap on the KEYBOARD drop-down box.
3. A list box pops up.
4. Touch to select CLIPBOARD or OFF to turn off SEND BARCODE DATA



COPY SCANNED DATA TO CLIPBOARD

The ScanWedge utility allows users/developers to copy scanned data to Clipboard according to the application's need. Unlike Send barcode data to Clipboard (above section), this feature does not send data stored in the clipboard directly to the input field. The data just stores in the clipboard and waits for further processing.

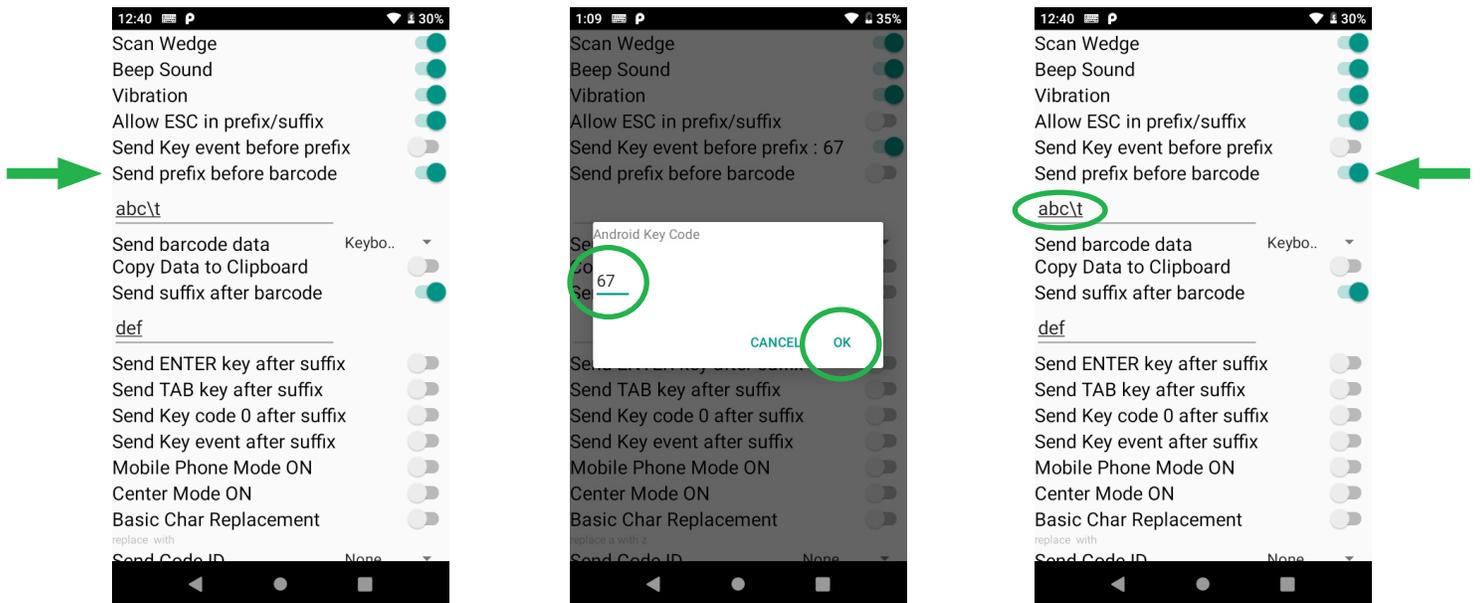
You can enable / disable the COPY DATA TO CLIPBOARD as shown



SEND KEY EVENT BEFORE PREFIX

The ScanWedge utility allows users/developers to send a defined key event (Android key event) before sending prefix. This feature is particularly useful to some terminal emulators where a backspace key must be input before the scan data is fed. To do this:

1. In the ScanWedge utility, tap on the SEND KEY EVENT BEFORE PREFIX switch.
2. A dialog box pops up.
3. Input the Android key event code (in numeric, for example, 67 means backspace)
4. Click OK.



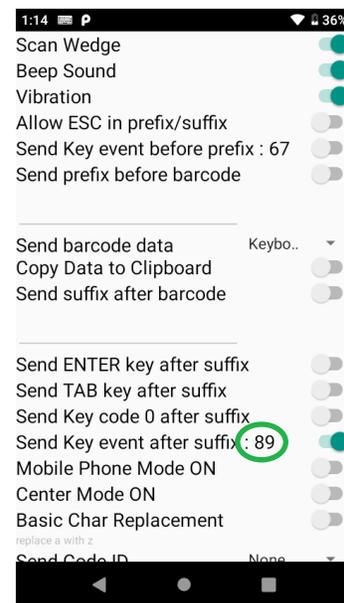
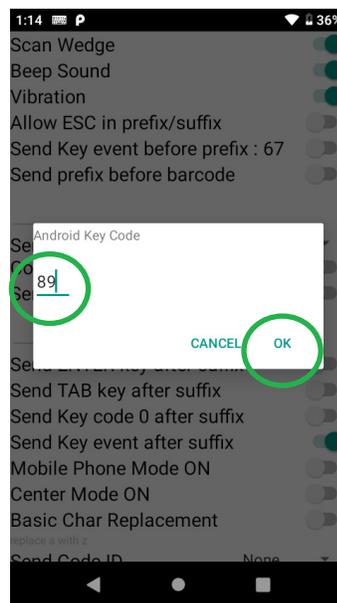
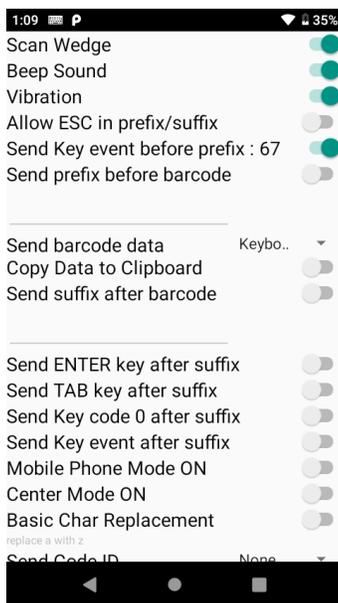
Note: the Android key event code can be found:

<https://developer.android.com/reference/android/view/KeyEvent>

SEND KEY EVENT AFTER SUFFIX

The ScanWedge utility allows users/developers to send a defined key event (Android key event) after sending a suffix. This feature is particularly useful to input a row of data where TAB is treated as a delimiter. To do this:

1. In the ScanWedge utility, tap on the SEND KEY EVENT AFTER SUFFIX switch.
2. A dialog box pops up.
3. Input the Android key event code (in numeric, for example, 61 means TAB).
4. Click OK



Make sure the switch turns green and the Android key event code is shown as above

Note: the Android key event code can be found: <https://developer.android.com/reference/android/view/KeyEvent>

SEND ENTER KEY AFTER SUFFIX

The ScanWedge utility can send the ENTER key right after the suffix. To do this:

1. In the ScanWedge utility, tap on the SEND ENTER KEY AFTER SUFFIX switch.
2. Make sure the switch is now green.

SEND TAB KEY AFTER SUFFIX

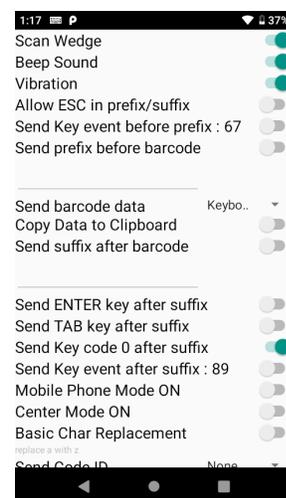
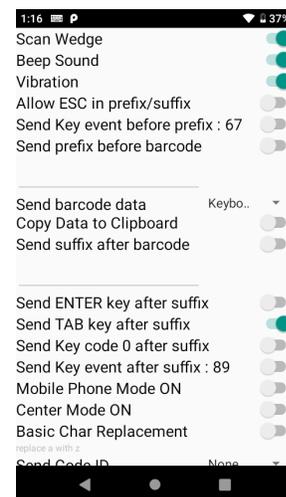
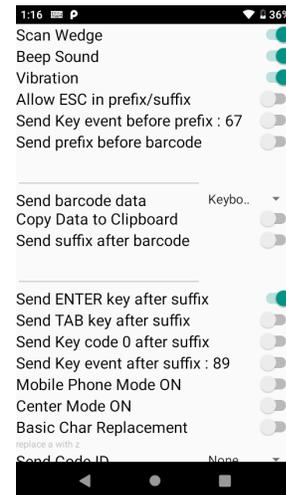
The ScanWedge utility can send the TAB key right after the suffix. To do this:

1. In the ScanWedge utility, turn on the SEND TAB KEY AFTER SUFFIX switch.
2. Make sure the switch is now green.

SEND KEY CODE 0 AFTER SUFFIX

To add suffix string to scanned data:

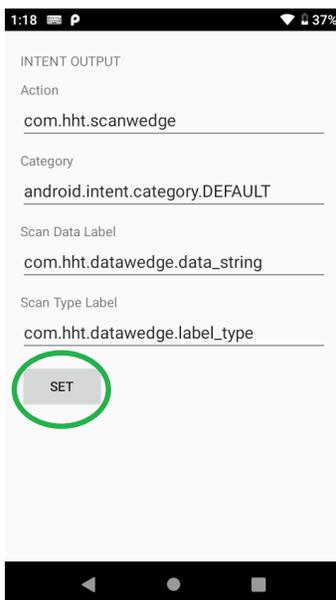
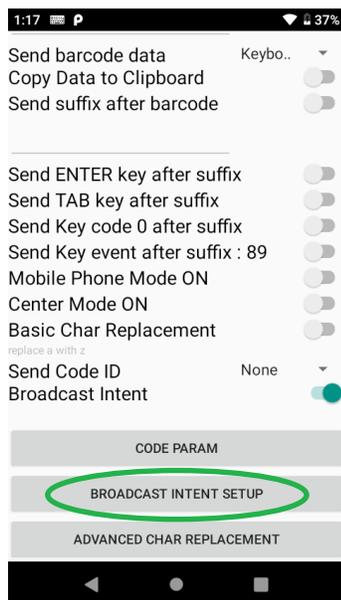
- In the ScanWedge utility, turn on the SEND KEY CODE 0 AFTER SUFFIX switch.
- Make sure the switch is now green.



NOTIFY OTHER APPLICATIONS VIA INTENT BROADCAST

You can customize the broadcast intent after scanned data is interpreted.

1. In the ScanWedge utility, turn on the Broadcast Intent switch (green).
2. To customize the Action / Category / Scan Data Label / Scan Type Label, click on the BROADCAST INTENT SETUP button.
3. An INTENT OUTPUT customization screen pops up.
4. A user/developer may customize corresponding items in this screen.
5. Click Set.

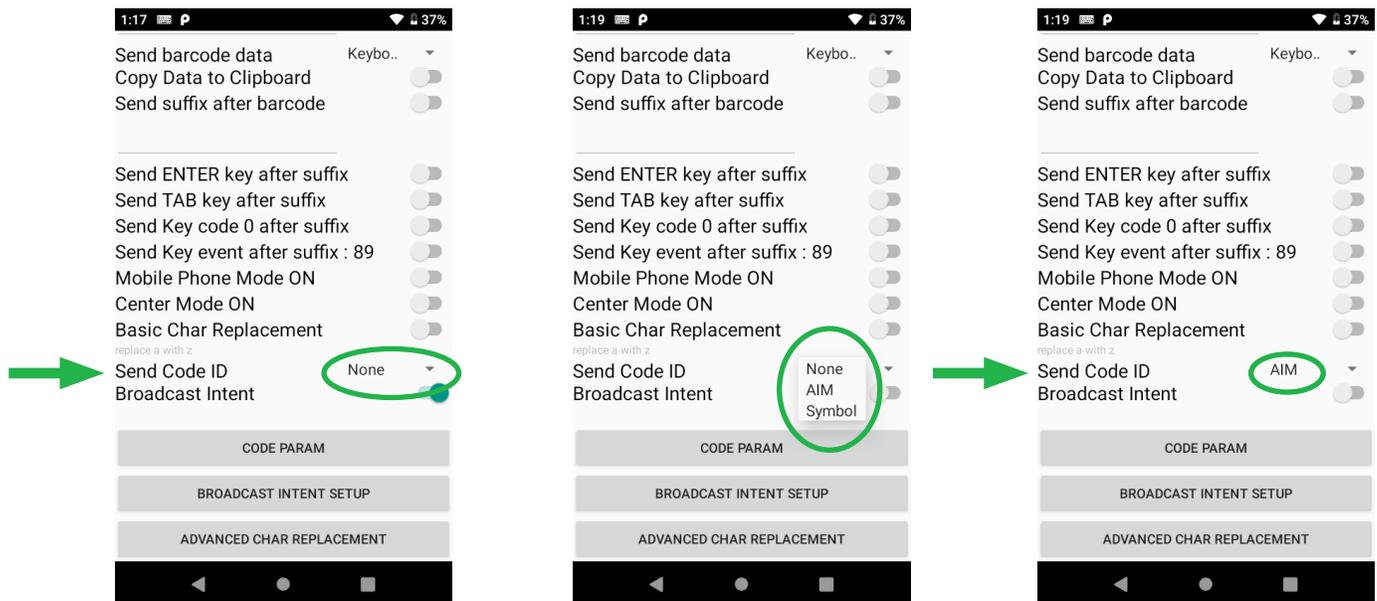


SEND BARCODE CODE ID ALONG WITH THE SCANNED DATA

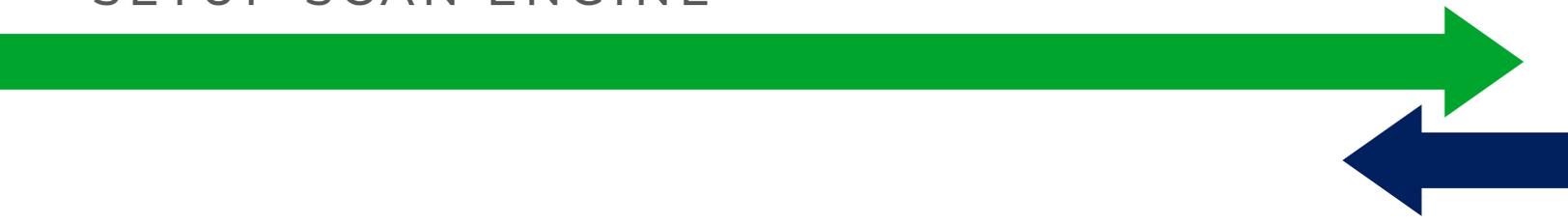
A Code ID character identifies the code type of a scanned barcode. This is useful when decoding more than one code type. In addition to any single character prefix already selected, the Code ID character is inserted between the prefix and the decoded symbol.

Currently there are two (2) ways to present the Code ID: Symbol and AIM. Application developers may choose either presentation. To do this:

1. In the ScanWedge utility, the default SEND CODE ID is None.
2. Tap on the NONE drop down box.
3. A list box pops up.
4. If you want to change to AIM, click OK.
5. If you want to change to Symbol, click OK.
6. Make sure your selected Code ID method is shown in the screen.



SETUP SCAN ENGINE

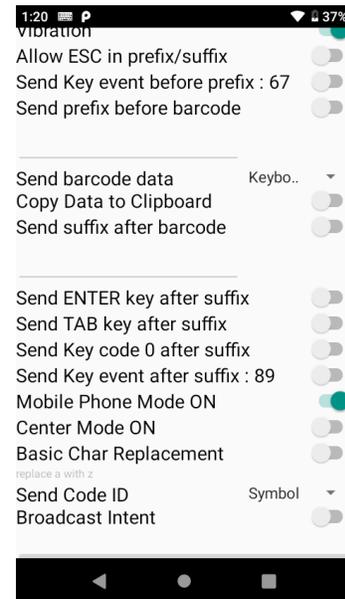


ENABLE / DISABLE MOBILE PHONE SCANNING MODE

Mobile Phone Scanning Mode allows users to optimize the barcode 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.

To enable Mobile Phone Scanning Mode:

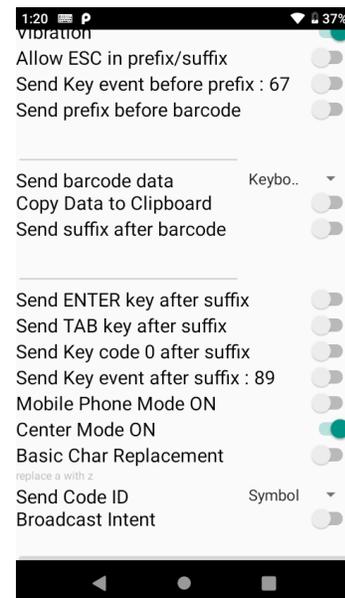
1. In the ScanWedge utility, turn the MOBILE PHONE MODE ON switch.
2. Make sure the switch now changes to green.



SCAN ONLY BARCODE LOCATED AT THE CENTER OF DECODING AREA

This feature enables the decoder to decode only barcodes aligned under the center of the laser aiming pattern. 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.

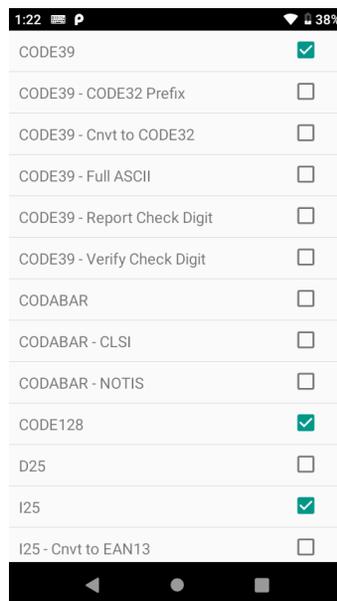
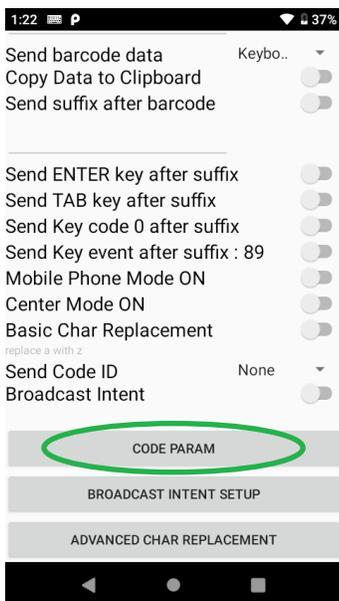
1. In the ScanWedge utility, turn on the CENTER MODE ON switch.
2. Make sure the switch now changes to green.



SETUP SCAN ENGINE DECODERS

The ScanWedge utility allows developers to select those decoders frequently used in the application in order to optimize the decoder performance. To do this:

1. In the ScanWedge utility, click on the CODE PARAM button.
2. A list of available barcode decoders pops up.
3. User / developer can check/uncheck decoder(s) to be available at run time on this screen



APPENDIX



TABLE 1: SYMBOL CODE IDENTIFIERS

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	GSI-128
L	Bookland EAN
M	Trioptic Code 39
N	Coupon Code
R	GSI DataBar Family
S	Matrix 2 of 5
T	UCC Composite, TLC 39
U	Chinese 2 of 5
V	Korean 3 of 5
X	SSN EAN, PDF417, Macro PDF417, Micro PDF417




TABLE 1.1 AIM CODE IDENTIFIERS

The ScanWedge utility allows developers to select those decoders frequently used in the application in order to optimize the decoder performance. To do this:

Each AIM Code Identifier contains the three-character string "]cm" where:

] = Flag Character (ASCII 93)

c = Code Character

m = Modifier Character

TABLE 2: 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

TABLE 2.1: 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 barcode with check character W, A+I+MI+DW, is transmitted as]A7AIMID where 7 = (3+4).	
Trioptic Code 39	0	No option specified at this time. Always transmit 0.
	Example: A Trioptic barcode 412356 is transmitted as]X0412356	
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]C1AIMID	
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 barcode without check digit, 4123, is transmitted as]I04123	
Codabar	0	No check digit processing.
	1	Reader has checked check digit.
	3	Reader has stripped check digit before transmission.
	Example: A Codabar barcode without check digit, 4123, is transmitted as]F04123	
Code 93	0	No options specified at this time. Always transmit 0.
	Example: A Code 93 barcode 012345678905 is transmitted as]G0012345678905	
MSI	0	Check digits are sent.
	1	No check digit is sent.
	Example: An MSI barcode 4123, with a single check digit checked, is transmitted as]M14123	
D 2 of 5 0	0	No options specified at this time. Always transmit 0.
	Example: A D 2 of 5 barcode 4123, is transmitted as]S04123	

TABLE 2.1: MODIFIER CHARACTERS, CONTINUED

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 barcode 012345678905 is transmitted as]E00012345678905	
Bookland EAN	0	No options specified at this time. Always transmit 0.
	Example: A Bookland EAN barcode 123456789X is transmitted as]X0123456789X	
ISSN EAN 0	0	No options specified at this time. Always transmit 0.
	Example: An ISSN EAN barcode 123456789X is transmitted as]X0123456789X	
Code 11	0	Single check digit
	1	Two check digits
	3	Check characters validated but not transmitted.
GSI DataBar Family		No option specified at this time. Always transmit 0. GSI DataBar and GSI DataBar Limited transmit with an Application Identifier "01". Note: In GSI-128 emulation mode, GSI DataBar is transmitted using Code 128 rules (i.e.,]C1).
	Example: A GSI DataBar barcode 0110012345678902 is transmitted as]e00110012345678902.	
EAN.UCC Composites (GSI DataBar, GSI-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.
		GSI-128 emulation Note: UPC portion of composite is transmitted using UPC rules.
	1	Data packet is a GSI-128 symbol (i.e., data is preceded with]JC1).

TABLE 2.1: MODIFIER CHARACTERS, CONTINUED

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 92DEC has been doubled in transmission.
	1	Reader set to follow the ECI protocol (Extended Channel Interpretation). All data characters 92DEC are doubled.
	2	Reader set for Basic Channel operation (no escape character transmission protocol). Data characters 92DEC 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 GSI-128 symbol, and the first codeword is 903-907, 912, 914, 915.
	4	The bar code contains a GSI-128 symbol, and the first codeword is in the range 908-909.
	5	The bar code contains a GSI-128 symbol, and the first codeword is in the range 910-911.
	Example: A PDF417 barcode ABCD, with no transmission protocol enabled, is transmitted as J 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 messages.
QR Code	0	Model 1 symbol.
	1	Model 2 / MicroQR symbol, ECI protocol not implemented.
	2	Model 2 symbol, ECI protocol implemented.

TABLE 2.1: MODIFIER CHARACTERS, CONTINUED

	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.

TABLE 3: ASCII CODE

DECIMAL	HEX	BINARY	HTML	CHARACTER	DESCRIPTION
0	00	00000000	�	NUL	Null
1	01	00000001		SOH	Start of Header
2	02	00000010		STX	Start of Text
3	03	00000011		ETX	End of Text
4	04	00000100		EOT	End of Transmission
5	05	00000101		ENQ	Enquiry
6	06	00000110		ACK	Acknowledge
7	07	00000111		BEL	Bell
8	08	00001000		BS	Backspace
9	09	00001001			HT	Horizontal Tab
10	0A	00001010	
	LF	Newline / Line Feed
11	0B	00001011		VT	Vertical Tab
12	0C	00001100		FF	Form Feed
13	0D	00001101		CR	Carriage Return
14	0E	00001110		SO	Shift Out
15	0F	00001111		SI	Shift In
16	10	00010000		DLE	Data Link Escape
17	11	00010001		DC1	Device Control 1
18	12	00010010		DC2	Device Control 2
19	13	00010011		DC3	Device Control 3
20	14	00010100		DC4	Device Control 4
21	15	00010101		NAK	Negative Acknowledge
22	16	00010110		SYN	Synchronize
23	17	00010111		ETB	End of Transmission Block
24	18	00011000		CAN	Cancel
25	19	00011001		EM	End of Medium
26	1A	00011010		SUB	Substitute
27	1B	00011011		ESC	Escape
28	1C	00011100		FS	File Separator
29	1D	00011101		GS	Group Separator
30	1E	00011110		RS	Record Separator
31	1F	00011111		US	Unit Separator

TABLE 3: ASCII CODE, CONTINUED

32	20	00100000	 	SP	Space
33	21	00100001	!	!	Exclamation mark
34	22	00100010	"	quot;	Double quote
35	23	00100011	#	#	Number
36	24	00100100	$	\$	Dollar
37	25	00100101	%	%	Percent
38	26	00100110	&	&	Ampersand
39	27	00100111	'	'	Single quote
40	28	00101000	((Left parenthesis
41	29	00101001))	Right parenthesis
42	2A	00101010	*	*	Asterisk
43	2B	00101011	+	+	Plus
44	2C	00101100	,	,	Comma
45	2D	00101101	-	-	Minus
46	2E	00101110	.	.	Period
47	2F	00101111	/	/	Slash
48	30	00110000	0	0	Zero
49	31	00110001	1	1	One
50	32	00110010	2	2	Two
51	33	00110011	3	3	Three
52	34	00110100	4	4	Four
53	35	00110101	5	5	Five
54	36	00110110	6	6	Six
55	37	00110111	7	7	Seven
56	38	00111000	8	8	Eight
57	39	00111001	9	9	Nine
58	3A	00111010	:	:	Colon
59	3B	00111011	;	;	Semicolon
60	3C	00111100	<	<	< Less than
61	3D	00111101	=	=	= Equal sign
62	3E	00111110	>	>	> Greater than
63	3F	00111111	?		
64	40	01000000	@	@	@ At sign

TABLE 3: ASCII CODE, CONTINUED

65	41	01000001	A	A	Uppercase A
66	42	01000010	B	B	Uppercase B
67	43	01000011	C	C	Uppercase C
68	44	01000100	D	D	Uppercase D
69	45	01000101	E	E	Uppercase E
70	46	01000110	F	F	Uppercase F
71	47	01000111	G	G	Uppercase G
72	48	01001000	H	H	Uppercase H
73	49	01001001	I	I	Uppercase I
74	4A	01001010	J	J	Uppercase J
75	4B	01001011	K	K	Uppercase K
76	4C	01001100	L	L	Uppercase L
77	4D	01001101	M	M	Uppercase M
78	4E	01001110	N	N	Uppercase N
79	4F	01001111	O	O	Uppercase O
80	50	01010000	P	P	Uppercase P
81	51	01010001	Q	Q	Uppercase Q
82	52	01010010	R	R	Uppercase R
83	53	01010011	S	S	Uppercase S
84	54	01010100	T	T	Uppercase T
85	55	01010101	U	U	Uppercase U
86	56	01010110	V	V	Uppercase V
87	57	01010111	W	W	Uppercase W
88	58	01011000	X	X	Uppercase X
89	59	01011001	Y	Y	Uppercase Y
90	5A	01011010	Z	Z	Uppercase Z
91	5B	01011011	[[Left square bracket
92	5C	01011100	\	\	backslash
93	5D	01011101]]	Right square bracket
94	5E	01011110	^	^	Caret / circumflex
95	5F	01011111	_	_	
96	60	01100000	`	`	Grave / accent
97	61	01100001	a	a	Lowercase a

TABLE 3: ASCII CODE, CONTINUED

98	62	01100010	b	b	Lowercase b
99	63	01100011	c	c	Lowercase c
100	64	01100100	d	d	Lowercase d
101	65	01100101	e	e	Lowercase e
102	66	01100110	f	f	Lowercase
103	67	01100111	g	g	Lowercase g
104	68	01101000	h	h	Lowercase h
105	69	01101001	i	i	Lowercase i
106	6A	01101010	j	j	Lowercase j
107	6B	01101011	k	k	Lowercase k
108	6C	01101100	l	l	Lowercase l
109	6D	01101101	m	m	Lowercase m
110	6E	01101110	n	n	Lowercase n
111	6F	01101111		o	o
112	70	01110000	p	p	Lowercase p
113	71	01110001	q	q	Lowercase q
114	72	01110010	r	r	Lowercase r
115	73	01110011	s	s	Lowercase s
116	74	01110100	t	t	Lowercase t
117	75	01110101	u	u	Lowercase u
118	76	01110110	v	v	Lowercase v
119	77	01110111		w	w
120	78	01111000	x	x	Lowercase x
121	79	01111001	y	y	Lowercase y
122	7A	01111010	z	z	Lowercase z
123	7B	01111011		{	{
124	7C	01111100	|		Vertical bar
125	7D	01111101		}	}
126	7E	01111110		~	~
127	7F	01111111			DEL