CipherLab

User Guide

MIRROR Browser

for

Windows Mobile 6.5 CP30

Windows Embedded Handheld 6.5 CP50

CP60

9200

V2.20



Copyright © 2014 CIPHERLAB CO., LTD.
All rights reserved

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

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

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of CIPHERLAB CO., LTD.

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

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

All brand, product and service, and trademark names are the property of their registered owners.

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

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

RELEASE NOTES

Version	Date	Notes
2.20	Aug 14, 2014	▸ Supports CP60R ▸ Add SIP icon in 3.4.2 Full Screen Mode ▸ Remove Browser option from 3.2 Menu ▸ Revise description for test pages in 3.4.1 Test Page ▸ Add 3.5.1 Default Test Pages, 3.6 Use Hotkeys, 5.1.3 Network Settings, 5.2 Text Size Adjustment ▸ Update 5.1.2 Screen Settings, 6.2 JavaScript APIs ▸ Add 5.4 Using On-screen Keyboard ▸ Update reader types in Appendix I: Scan Engine Settings
2.14	Dec 12, 2013	▸ Add 6.2.3 RFID – ReadRfidData, WriteRfidData
2.13	Nov 4, 2013	▸ Add note to 3.5 Address Bar & Drop-down Arrow: Arrow becomes unavailable when home screen is locked ▸ Update 5.1.1 Browser Settings: Add “Submit when Enter is received”
2.12	Aug 16, 2013	▸ Supports 9200 ▸ Add notice: Screen size and layout may vary according to device model type ▸ Update 1.2 Install Folder ▸ Revise 5.1.1 Browser Settings ▸ Revise 5.1.4 Reader Settings for CP60/9200 ▸ Add 5.2 Import/Export Settings, 5.3 About, 5.4 Exit Application ▸ Update Appendix I~IV
2.10	Mar 1, 2013	▸ Supports CP60 ▸ Revise Feature: Cloning via INI file ▸ CP60 support: Add 5.4 Reader Settings for CP60, Appendix II/III Symbology Settings for CP60 ▸ 1.2 Install Folder: Add Cloning of Browser configurations ▸ 1.3 Launch Mirror Browser, 2.1 Activate: Remove activation message upon application launch ▸ 3.3 Toolbar: Revise full-screen mode description ▸ Add 3.4.2 Full Screen Mode ▸ 5.2 Screen Settings: Add Screen Settings under Browser Menu ▸ Update 6.2 JavaScript APIs ▸ Update Appendix I Symbologies Supported
2.01	Aug 21, 2012	Typo correction
2.00	Jul 6, 2012	Supports CP50
1.00	Aug 10, 2011	Initial release for CP30

CONTENTS

RELEASE NOTES III

INTRODUCTION 1

Features 2
Development Tools 2
Licensing 2

SETUP 3

1.1. Install 4
1.1.1. Install from PC 4
1.1.2. Install from Mobile Computer 8
1.2. Install Folder 10
1.3. Launch MIRROR Browser 11

ACTIVATE 13

2.1. Activate 13

USER INTERFACE 15

3.1. Screen Terminology 16
3.2. Menu 17
3.3. Toolbar 18
3.4. Browser View 19
3.4.1. Test Page 20
3.4.2. Full Screen Mode 21
3.5. Address Bar & Drop-down Arrow 24
3.5.1. Change Test Pages 25
3.6. Use Hotkeys 27

BROWSE AND COLLECT DATA 29

4.1. Open Web Pages 29
4.2. Collect Data 29
4.3. Noteworthy Browsing Facts 29

SETTINGS 31

5.1. Option Settings 32
5.1.1. Browser Settings 32
5.1.2. Screen Settings 35
5.1.3. Network Settings 38
5.1.4. Reader Settings for CP30/CP50 40
5.1.5. Reader Settings for CP60/9200 51
5.2. Text Size Adjustment 52
5.3. Import/Export Settings 53
5.4. Using On-screen Keyboard 55
5.5. About 56
5.6. Exit Application 57

DEVELOP YOUR HTML DOCUMENT 59
6.1. Data Collecting Script & Form 60
6.1.1. How Data Collection Works on HTML Doc 60
6.2. JavaScript APIs 61
6.2.1. System Information 61
6.2.2. Symbology 65
6.2.3. Reader 68
6.2.4. RFID Reader 85
SCAN ENGINE SETTINGS 105
Symbologies Supported 106
RFID Tags Supported 108
RFID Tags Supported on CP50 108
RFID Tags Supported on CP60 109
RFID Tags Supported on 9200 110
CCD (SM1) 111
Symbology Settings for 9200 111
LASER (SE955) 115
Symbology Settings for CP30/CP50/9200 115
Miscellaneous 119
Symbology Settings for CP60/9200 120
Miscellaneous 125
AIM Code ID – Code Characters 126
AIM Code ID – Modifier Characters 126
2D IMAGER (SE4500DL) 129
Symbology Settings for CP30/CP50 129
1D Symbologies 129
2D Symbologies 136
Miscellaneous 137
Symbology Settings for CP60/9200 138
1D Symbologies 138
2D Symbologies 145
Miscellaneous 146
HF RFID READER 147
RFID Tag Default Block 147
RESPONSE CODE INSTRUCTIONS 149

INTRODUCTION

As more and more data collection goes on web, the need arises for a web browser with extended ability to control readers. CipherLab heard your wishes and answered it by bringing MIRROR Browser, a compact web browser rooted in **Internet Explorer Mobile** with reduced but essential reader configuration thrown in. MIRROR Browser facilitates your web-based data collection and, at the same time, offers easy operation for launching and tuning the reader module.

MIRROR Browser makes it effortless to integrate a client device to a web server afar. MIRROR Browser enables a client mobile computer to pass collected data to a remote web server by simply navigating to the web page in mind, pressing the scan key and submitting data.

This manual explicates the use of MIRROR Browser, and also includes HTML scripts and a series of APIs for users to create their own dynamic HTML document that collects data. We recommend that you read this manual before use and keep it at hand for quick reference.

Thank you for choosing CipherLab products!

FEATURES

- Supports CipherLab handhelds powered by Windows Mobile 6.5 and Windows Embedded Handheld 6.5
- Browses web pages on a remote web server
- Updates real-time data to remote web server
- User-friendly with simplified user interface
- Auto-enters data upon reading and decoding
- Allows full-screen maximization with one click
- Provides screen lock against unauthorized users
- Imports and exports settings as profiles
- JavaScript APIs provided for device control
- Supports SSL communications protocol

DEVELOPMENT TOOLS

- A plain text editor such as Notepad (of Windows OS) to create an HTML document, either .htm or .html.
- JavaScript

LICENSING

MIRROR Browser requires a product key for activation; otherwise it allows only 20-minute trial-use. Contact your CipherLab sales representative for license terms and price.

SETUP

This chapter guides to the setup of MIRROR Browser, which is a simple task that takes just a few clicks to activate the installer and go through installation.

This chapter also explicates the 1st launch of the application and where the application is installed to.

IN THIS CHAPTER

1.1 Install 4
1.2 Install Folder 10
1.3 Launch MIRROR Browser 11

1.1. INSTALL

MIRROR Browser presents a compressed archive that yields two installation files, an .exe and .cab file. They allow installation through two approaches.

Each installation file is able to set up MIRROR Browser independently. Users can handle the setup either on the mobile computer or from the comfort of their Windows-based PC.

The two approaches are explicated in the following.

1.1.1. INSTALL FROM PC

Installation from PC relies hardware-wise on the USB connection and software-wise on the ActiveSync synchronization partnership between the PC and mobile computer.

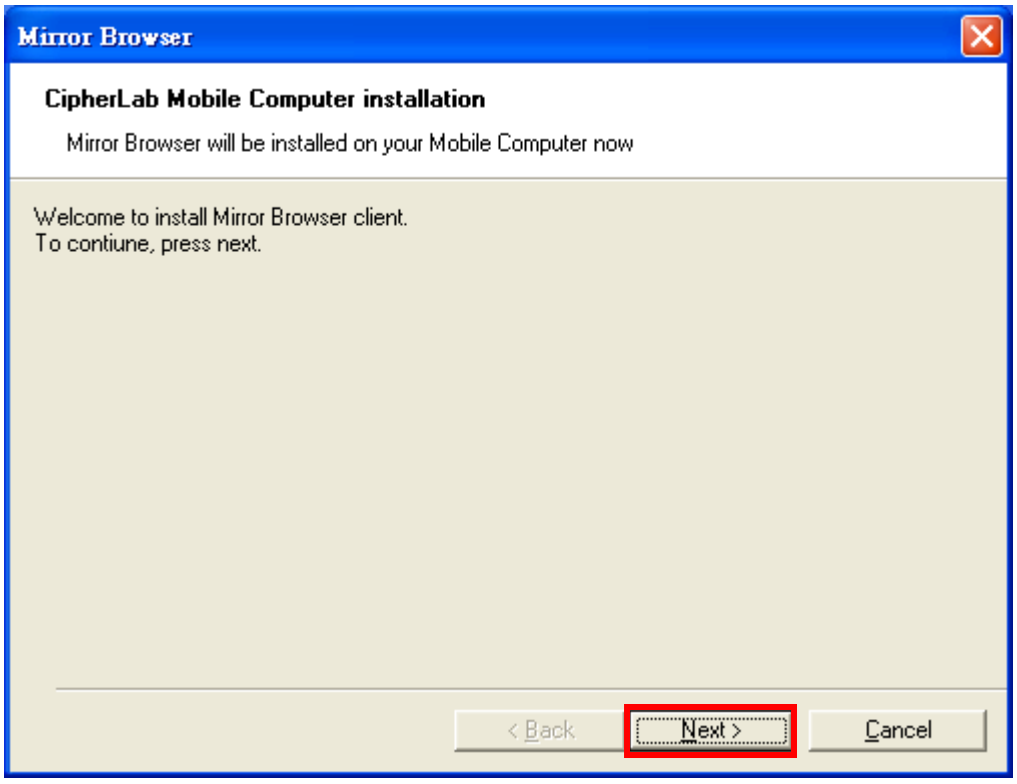
To install MIRROR Browser from PC:

1) Set up USB connection between the PC and mobile computer.
2) Run ActiveSync on PC.
3) Establish sync partnership between two computers.

   For details about sync partnership between two computers, see the reference manual of due mobile computer.

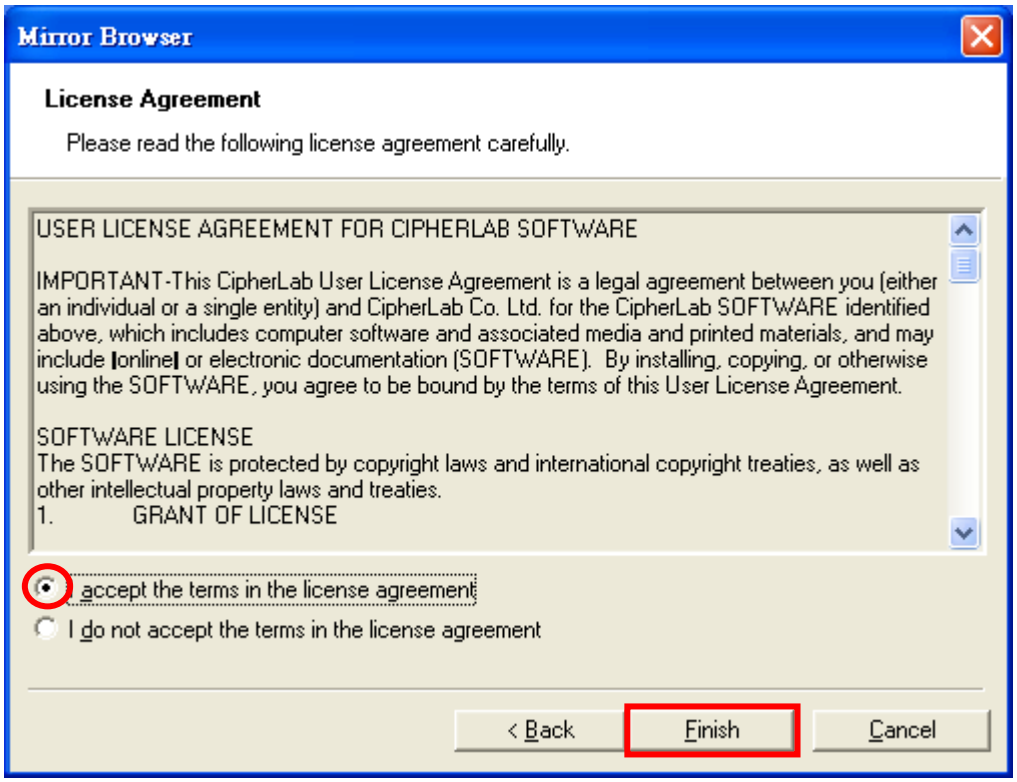
4) On PC, run the executable file MirrorBrowser_X.XXXX.exe. (X specifies the version).

   Installation wizard opens.



5) Click **Next** to proceed.

You are led to a **License Agreement** window.



6) Select **I accept the terms in the license agreement**. Click **Finish** to proceed.

The wizard then offloads Mirror Browser to the synced mobile computer.



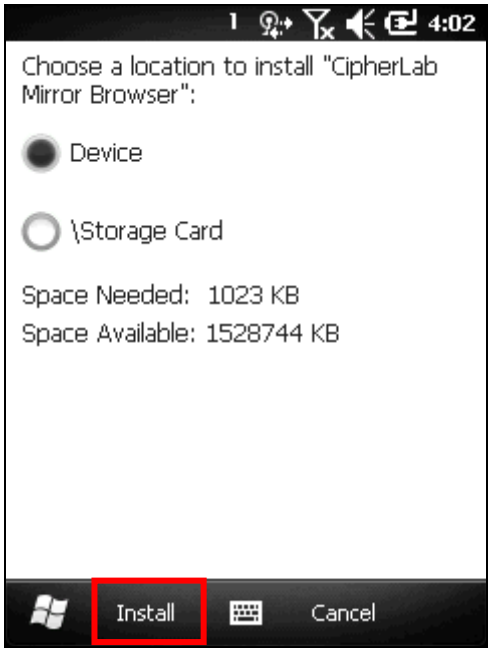
Once offload finishes, a dialog opens prompting you to check on the mobile computer.



7) Click **OK**.

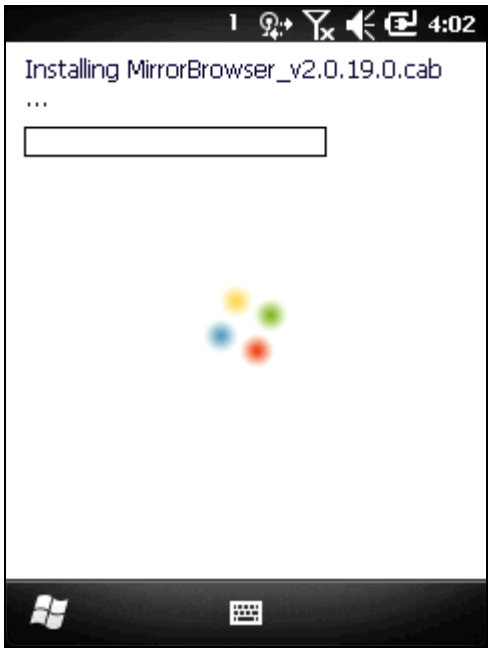
8) Check the mobile computer, which shows a screen that prompts you to choose the storage to install Mirror Browser, internal or external.

   If there isn't an SD card on board the mobile computer, the software will be installed to the internal storage.

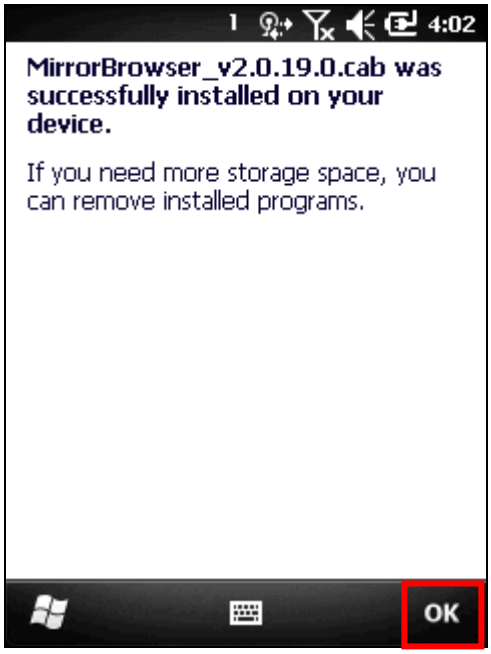


9) Select desired storage. Tap **Install** on softkey bar.

   Mirror Browser proceeds to install on the mobile computer.

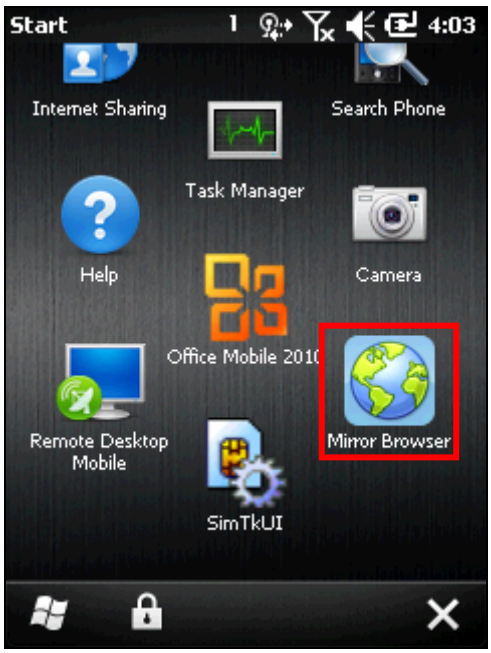


The installation is through in a few seconds. Once through, the screen shows a notice of successful installation.



10) Tap **OK** to finish installation.

MIRROR Browser icon comes up on Start screen.



11) Proceed to launch MIRROR Browser as described in Launch MIRROR Browser.

1.1.2. INSTALL FROM MOBILE COMPUTER

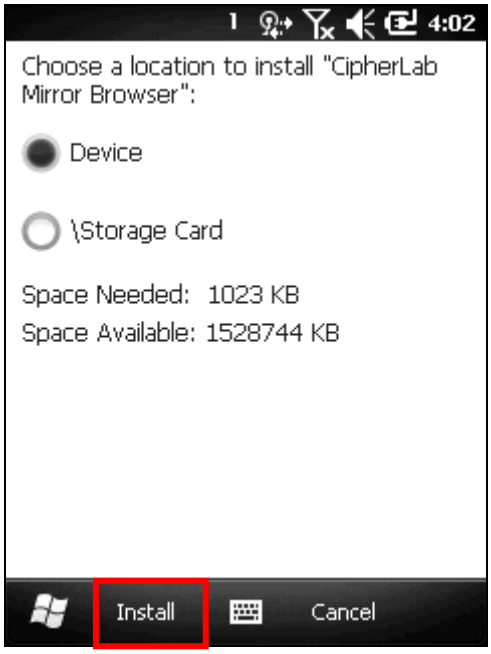
Installation from mobile computer relies on the .cab file extracted from the compressed archive. The mobile computer needs a copy of this .cab file in its storage, either internal or external storage will do.

To install from mobile computer:

1) Copy MirrorBrowser_X.XXXX.cab (X for the version) to the mobile computer, either internal or external storage.
2) On the mobile computer, open Start screen.
3) Tap File Explorer .
4) Browse for the .cab file.
5) Tap the .cab file.

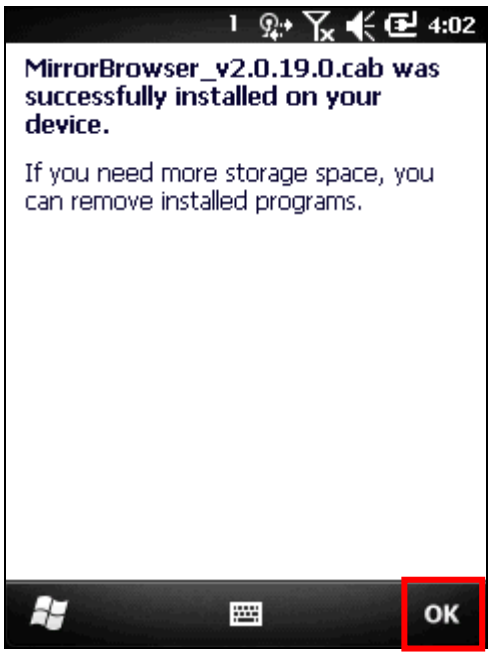
   Installer starts. A screen opens prompting you to choose the storage to install MIRROR Browser to, internal or external.

   If there isn't an SD card on board the mobile computer, the software will be installed to the internal storage.



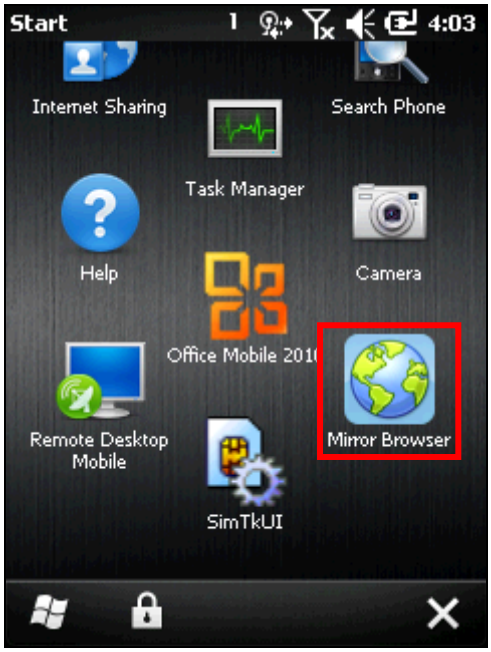
6) Follow on-screen instructions to proceed.

Once the installation is through, the screen shows a notice of successful installation.



7) Tap **OK** to finish and quit installation.

MIRROR Browser icon comes up on Start screen.



8) Proceed to launch MIRROR Browser as described in Launch MIRROR Browser.

1.2. INSTALL FOLDER

If you haven't selected an alternate, MIRROR Browser is installed to the internal storage under **My Device\Program Files\Mirror Browser**. Alongside the executable file that launches MIRROR Browser, there are some others as explicated below:

File	Description
MirrorBrowser.exe	The executable file that launches MIRROR Browser.
MirrorBrowser.png	MIRROR Browser's shortcut graphic, which users tap on-screen to launch MIRROR Browser.
test.htm	An HTML document that delivers users' test-scanned data. This is the page that loads and displays when MIRROR Browser launches (if no home page is assigned in Browser Settings).
browser_setting.ini	Stores browser settings. Available only after defaults are changed.
Develop_reader.ini (only on CP30/50)	Stores reader settings. Available only after defaults are changed.

1.3. LAUNCH MIRROR BROWSER

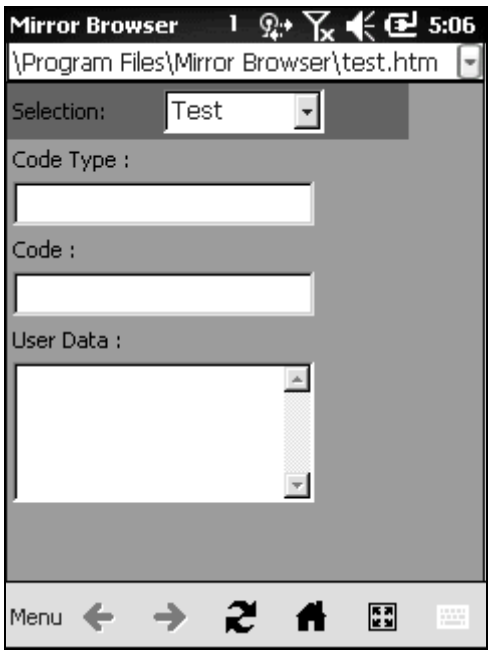
After installation is completed, proceed to launch MIRROR Browser.

To launch MIRROR Brower:

1) Open **Start** screen.

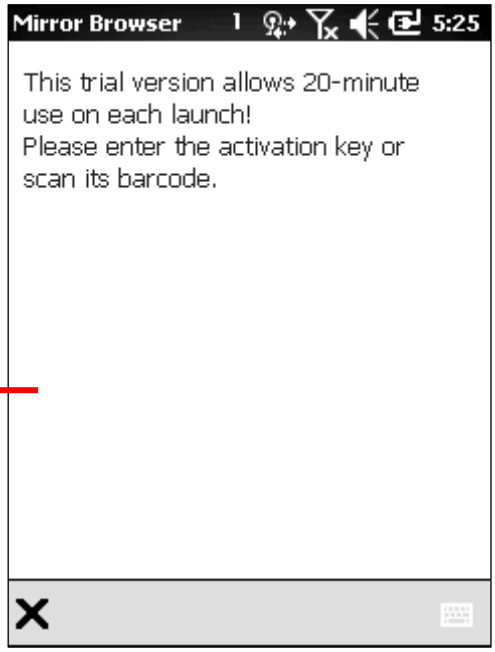
2) Tap MIRROR Browser icon .

MIRROR Browser launches. Trial period starts and the test.htm displays.



Without activation, MIRROR Browser allows a 20 minute trial-usage. Once the trial period is over, an Activation screen will appear on-screen notifying that “activation” is required for continuous usage. To use MIRROR Browser uninterruptedly, activate it. See Activate.

If no activation key is entered, an Activation screen appears after 20 minutes notifying that trial period is over



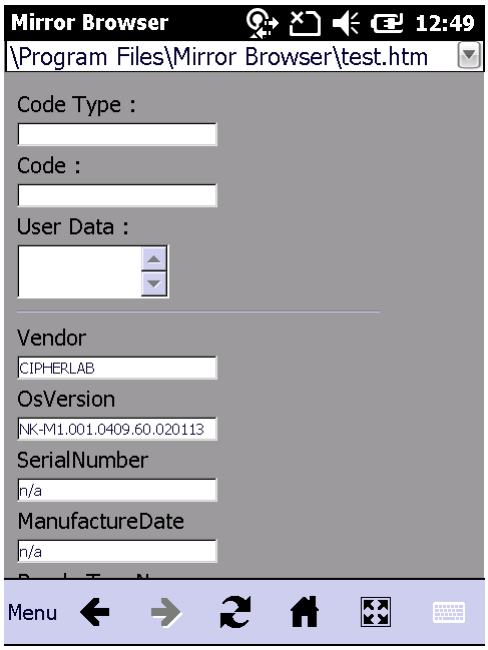
ACTIVATE

This chapter explicates activation of MIRROR Browser with an activation key.

2.1. ACTIVATE

If an activation key is available, activate MIRROR Browser following the steps below:

1) Launch MIRROR Browser as described in Launch MIRROR Browser.

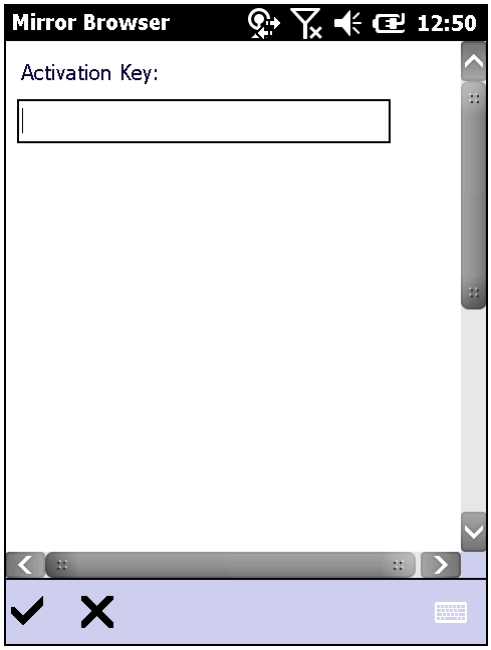


2) Tap the **Menu** label seated on the softkey bar.



3) Tap **Activation**.

"Activation key" screen opens presenting a single input field.



4) Enter an activation key by scanning the activation barcode.

OR

Manually type the activation key with the on-screen keyboard or physical keypad.

5) Press Enter key to submit the input.

A dialog opens notifying activation success.

6) Tap **ok** at the upper-right of the dialog.

The dialog closes and MIRROR Browser is activated.

USER INTERFACE

MIRROR Browser comes with an environment with graphics, bars and boxes for users to tap on-screen so its settings can be accessed and actions can be taken. This chapter gives an overview of MIRROR Browser's system appearances.

Note: For different Windows Mobile or Windows Embedded Handheld device models, screen sizes and layout will differ according to the LCD panel resolution of each device.

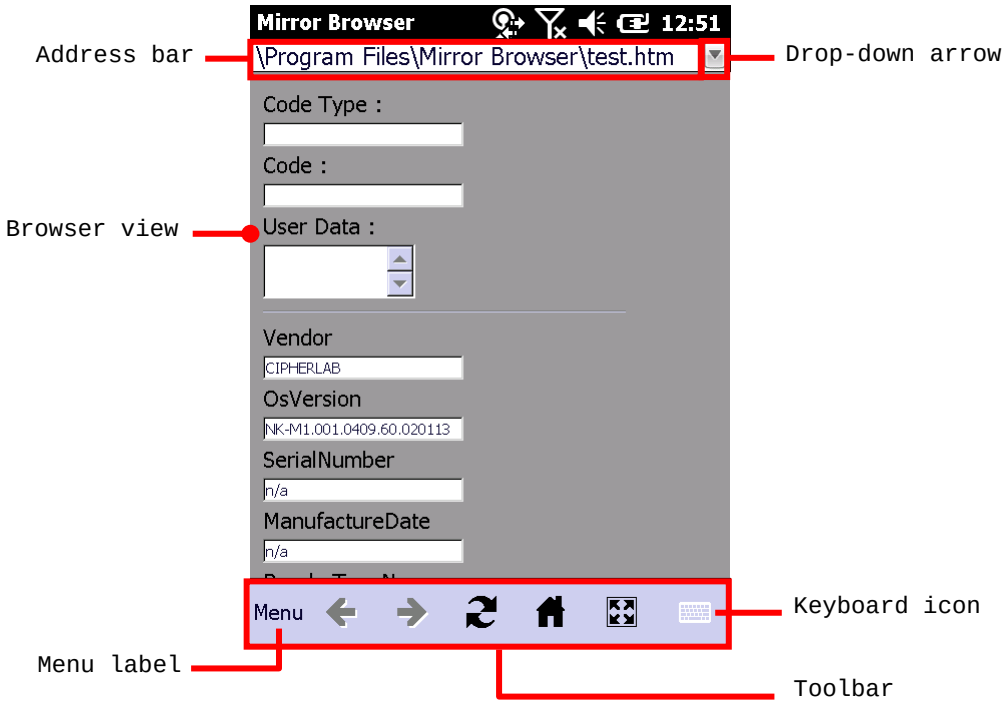
IN THIS CHAPTER

3.1 Screen Terminology 16
3.2 Menu 17
3.3 Toolbar 18
3.4 Browser View 19
3.5 Address Bar & Drop-down Arrow 24

3.1. SCREEN TERMINOLOGY

The screenshot below shows MIRROR Browser opened on the mobile computer. The software presents a test.htm in its window along with several GUI elements.

On-screen elements are illustrated as follows:



Term	Description
Browser view (or simply “Browser”)	The screen MIRROR Browser displays which users work on to collect data. See Browser View for details.
Address bar	A text box that enables users to enter a URL of the page they wish to display in MIRROR Browser. Address bar comes with a drop-down arrow at the right. See Address Bar & Drop-down Arrow for details.
Drop-down arrow	Shows recently visited pages. See Address Bar & Drop-down Arrow for details.
Menu label	Opens MIRROR Browser menu. See Menu for details.
Toolbar	Bears a few icons that quick launches some actions from MIRROR Browser. See Toolbar for details.

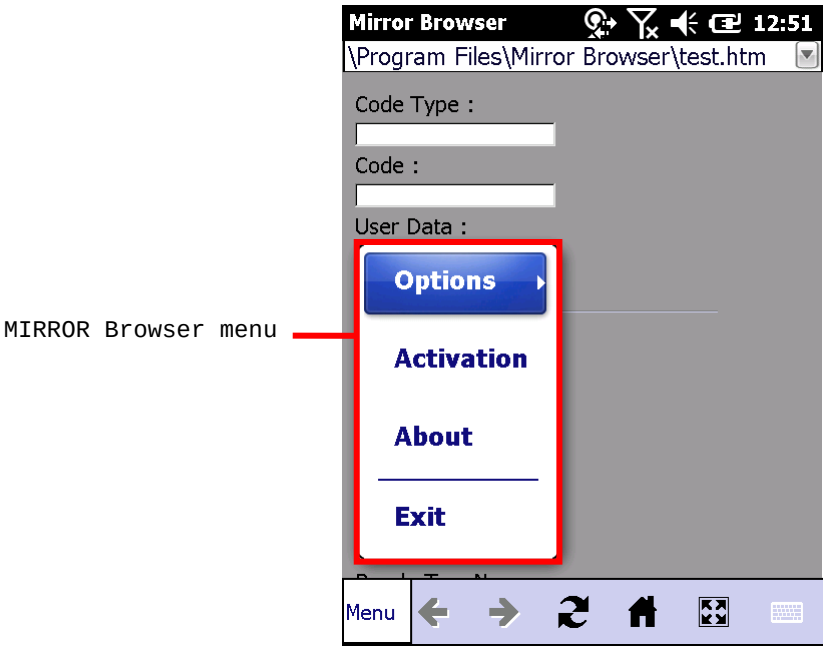
3.2. MENU

MIRROR Browser features a menu label, which provides a list of options for users to make selections and access the settings of MIRROR Browser.

Note: General browser settings should be made in **Internet Explorer Mobile** as described in General Browser Settings.

To open MIRROR Browser menu:

1) Launch MIRROR Browser as described in Launch MIRROR Browser.
2) Tap the bottom-left **Menu** label.



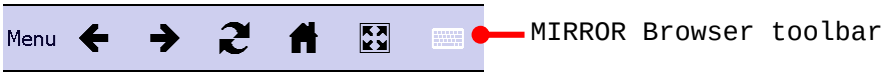
MIRROR Browser menu delivers the following functions:

Item	Description
Option	Accesses the settings of Browser View, readers and symbologies.
Activation	Activates MIRROR Browser for uninterrupted use.
About	Delivers software developer and version info.
Exit	Quits MIRROR Browser.

3.3. TOOLBAR

Different from menu, the toolbar bears on-screen buttons. Each button directly launches an action from MIRROR Browser.

Toolbar is presented along the bottom of Browser View as long as Browser View isn't maximized to full-screen.



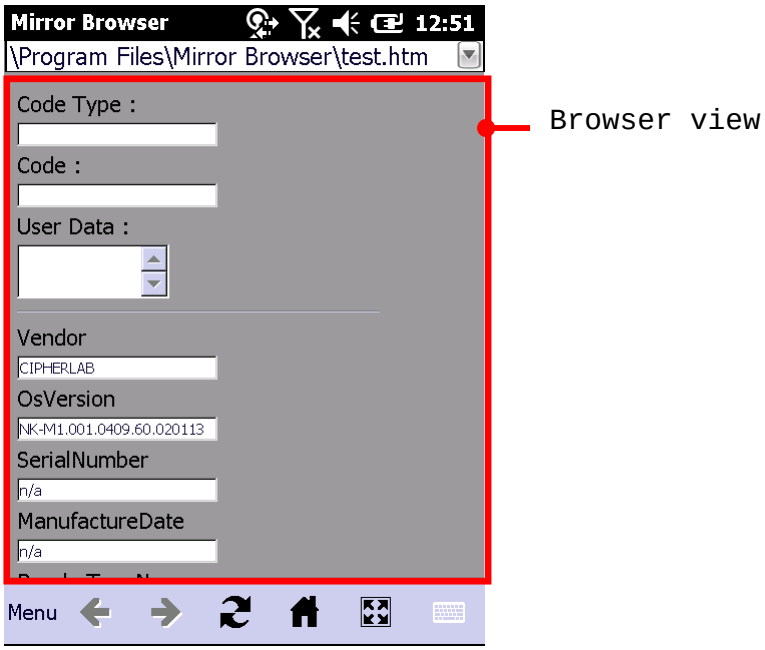
Toolbar delivers the following functions:

Icon	Description
Menu	Opens MIRROR Browser menu. See Menu for details.
	Opens the previous page worked on.
	Moves forward to the next page (which was previously opened).
	Reloads the current page or loads a page again if it doesn't display properly.
	Opens MIRROR Browser's home page, which is definable at **Menu** | **Option** | **Settings** | **Browser** tab page.
	Maximizes Browser View to full screen whereby the **Menu** label, toolbar and address bar will all be hidden. Certain operations can be performed under Full Screen Mode, including relocation of icons, checking actual percentage for battery and Wi-Fi status, and exiting full screen. Users can select whether to display the OS taskbar or specific status icons under full screen mode. See Screen Settings for details.
	Opens on-screen keyboard to input text.

3.4. BROWSER VIEW

"Browser view", or simply "Browser", is the web-browsing area that collects data.

The "test.htm" page loads in Browser view when MIRROR Browser launches if there is no home page defined yet. The test.htm is a file that comes with MIRROR Browser and shares the same whereabouts with MIRROR Browser's installation files at **My Device\Program Files\MIRROR Browser**.



3.4.1. TEST PAGE

Mirror Browser provides a few test pages that can serve as templates for your HTML document. These test pages carry data collecting scripts that retrieve data using JavaScript APIs. The default HTML page opened in Browser View when Mirror Browser launches is "test.htm". See Change Test Pages for more details on other test pages.

To use the default test page:

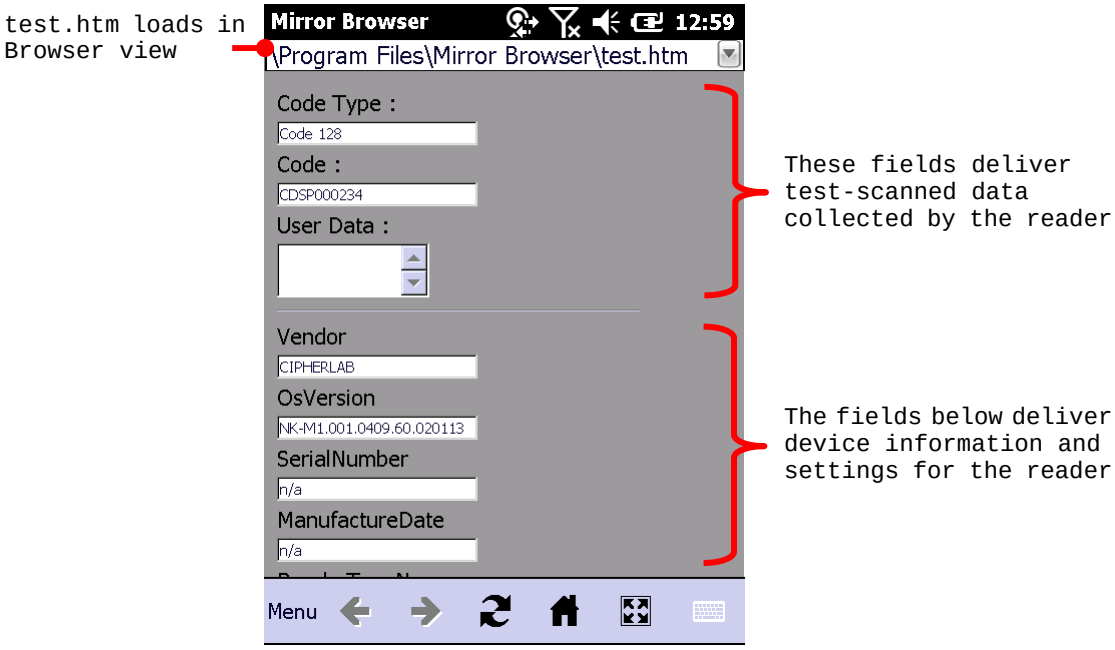
1) Launch MIRROR Browser as described in Launch MIRROR Browser.

   Test.htm opens (on condition that no home page has been defined under Browser Settings).

2) Scan a barcode or an RFID tag by pressing the physical SCAN key or either side trigger.

   The scanned data is entered into the blank areas of the form.

   The fields below deliver device information as well as notification and data output settings for the reader.



For more applications, see Develop Your HTML Document for how to collect data using self-defined HTML documents.

3.4.2. FULL SCREEN MODE

Browser View can be maximized to full screen by tapping the icon  on the toolbar.

When full screen mode is activated, a resize icon  shows by default to quit maximization and bring Browser view back to its original scale.

In full screen mode, the Browser toolbar, Menu label and the address bar are all stripped from the screen. As for the OS taskbar located at the top of the screen, users can select whether to have this displayed when the Browser is maximized to full screen. Alternatively, users can select whether to display icons to indicate battery level and Wi-Fi signal strength, and an icon to open the on-screen keyboard. See Screen Settings for whether to display the taskbar or the two said icons.

Resize icon: 

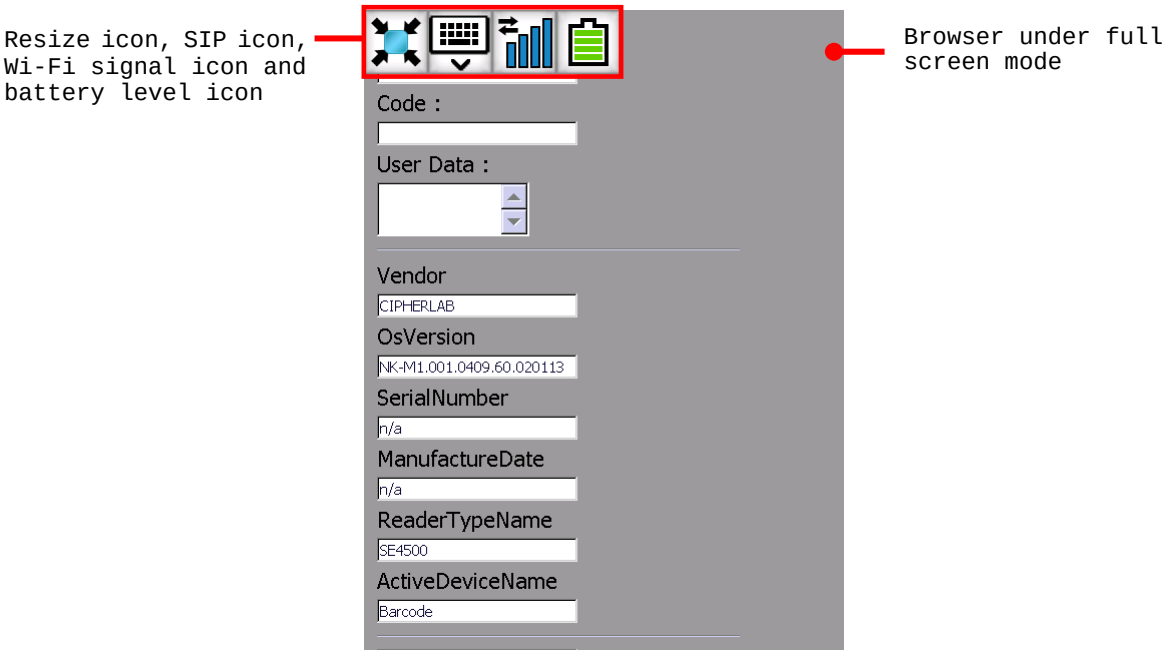
SIP icon: 

Battery level icons: 

Wi-Fi signal strength icons: 

ACTIONS TO TAKE ON FULL SCREEN

Several actions can be taken on the maximized full screen to improve performance.

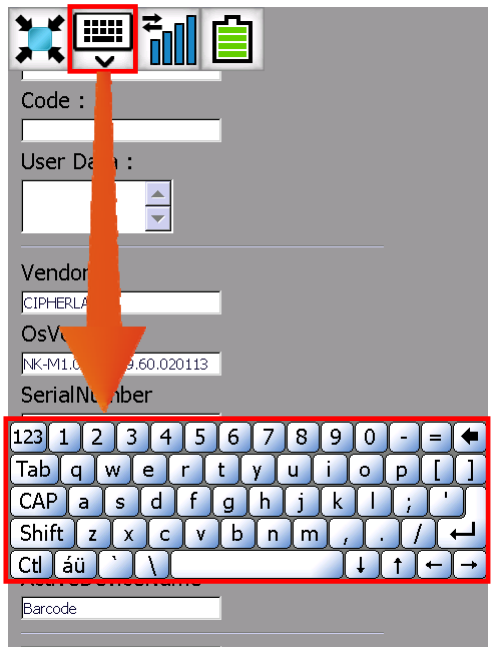


QUIT FULL SCREEN MODE

To quit full screen mode, double-tap the resize icon . Browser View will open on the screen.

OPEN ON-SCREEN KEYBOARD

Double-tap the SIP icon to open the on-screen keyboard for text input. Double-tap the icon once more to close the on-screen keyboard.



RELOCATE ICONS

By default, the resize icon, battery level icon and Wi-Fi signal strength icon will appear in the upper-left corner of the screen. These icons are movable and can be individually relocated to a better position on-screen. Simply drag and drop the icon to move it to the desired position.

VIEW BATTERY LEVEL & WI-FI SIGNAL

The graphical icons for battery and Wi-Fi only roughly sketch the battery level and Wi-Fi signal strength. To view the precise percentage of the battery level or Wi-Fi signal strength, double-tap the icon in order to pop up a dialog which delivers the precise percentage.

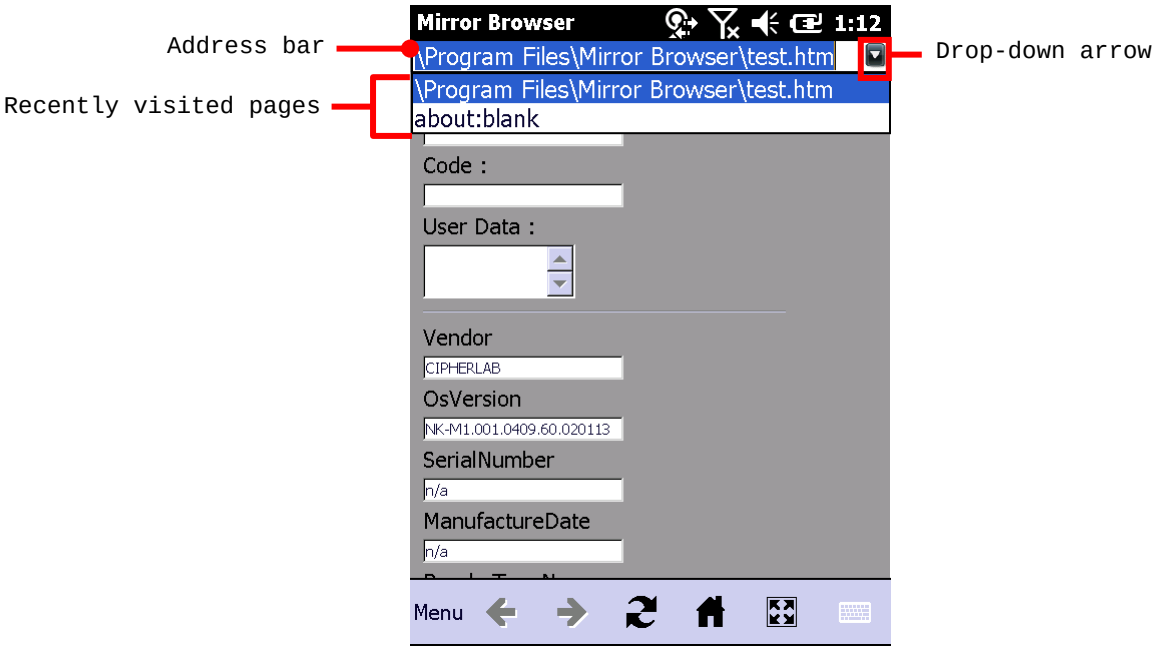


3.5. ADDRESS BAR & DROP-DOWN ARROW

The address bar is a text box atop Browser View for users to enter web page URL. It can be concealed by maximizing Browser View using the maximize icon  on toolbar. This bar is also subject to inactivation by one of the Browser Settings.

The address bar comes with a drop-down arrow to its right side, which can be used to open a drop-down list of up to 20 recently visited web pages.

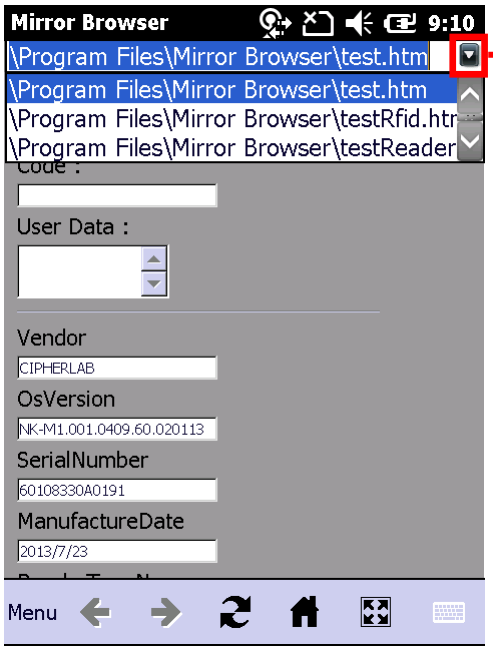
Note: When the home screen is locked as in Lock the Home page at the next start, the drop-down arrow will become inaccessible.



3.5.1. CHANGE TEST PAGES

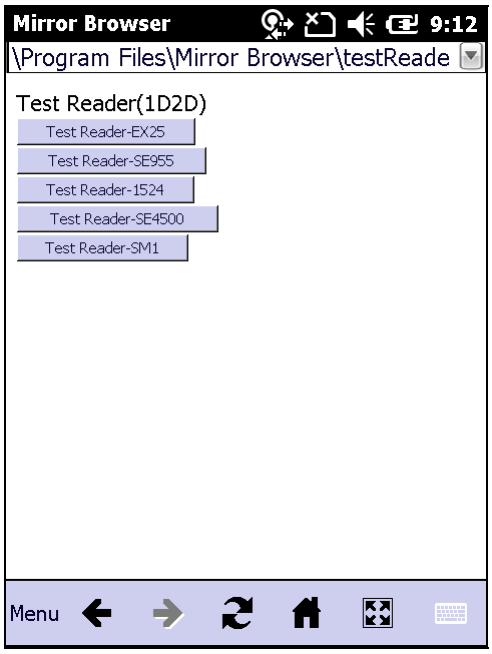
By default, the Browser provides three test pages which present different system and reader information retrieved through JavaScript APIs.

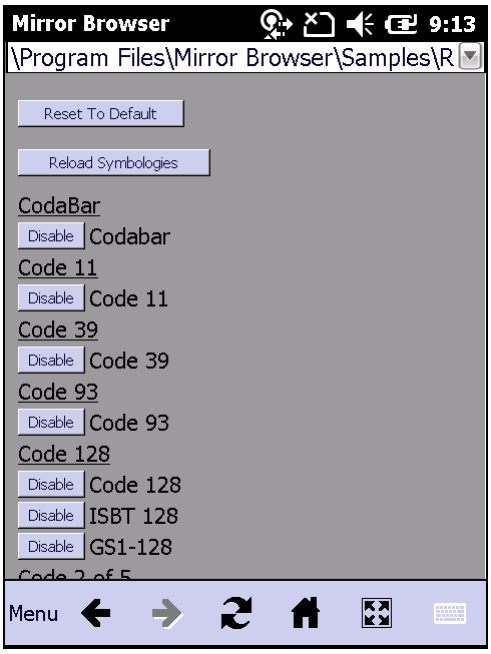
To change the test page opened in the Browser View, tap the drop-down arrow to show the list of default test pages.



Tap to open list for default test pages

Tap the test page to open, or press the Down button on the physical keypad to select the test page to open. The new default page provides items to adjust reader settings.





The three available default pages are:

Test Page	Description
test.htm	This is the default test page opened in Browser view when Mirror Browser is first launched. This page provides blank fields which are used to accept decoded barcode data, display information about the device and reader type, and also set notification for good reading and configure data output format.
testReader.htm	Use this test page to enable/disable the symbologies decoded by the reader. Select the reader type to configure. ▸ For settings to be effective, the selected reader type must match the Reader Type Name shown on the default test page (test.htm).
testRFID.htm	Use this test page to open or close the specific card, read its UID, read/write RFID data, and also set the starting block to read/write and input the data write. Select the RFID card type to work on. ▸ Effective on RFID devices only.

3.6. USE HOTKEYS

Use the following hotkeys to operate on the Browser View.

Hotkey	Description
F1	Opens Home page. Same as tapping the  icon on the toolbar.
F5	Refreshes the page opened in Browser view. Same as tapping the  icon on the toolbar.
F10	Opens the settings window. Same as tapping **Menu** | **Options** | **Settings**.
F11	Maximizes the screen, or quits full screen mode. Same as tapping the  icon on the toolbar, or tapping the  icon on the maximized screen.
Shift + Up	Increases the text size shown in Browser view. Same as selecting a larger text size in Text Size Adjustment.
Shift + Down	Decreases the text size shown in Browser view. Same as selecting a smaller text size in Text Size Adjustment.
Shift + Esc	Shows or hides the on-screen keyboard. Same as tapping the  icon on the maximized screen. ▸ To enabled this hotkey, select **Set HW key (SHIFT+ESC) to control SIP** under **Menu** | **Options** | **Settings** | **Screen**. See Screen Settings.

BROWSE AND COLLECT DATA

This chapter gives a brief sketch of browsing web pages and collecting data using MIRROR Browser.

4.1. OPEN WEB PAGES

To open a web page in Browser view:

1) Launch MIRROR Browser as described in Launch MIRROR Browser.

   MIRROR Browser opens in Browser view.

2) Enter the URL of the web page to visit in the address bar. Press/tap **Enter** key.

   OR

   Tap the drop-down arrow  at the right of the address bar to view the recently visited web pages. Tap the URL of the web page you would like to open.

   MIRROR Browser loads and displays the page to visit.

3) You can now proceed to collect data.

4.2. COLLECT DATA

After a web page with a data collecting form opens in Browser view, proceed to collect data:

1) Aim the scanning window at the printed barcode to read.

   OR

   Tap (slightly contact) the scanning window with the RFID tag to read.

2) Press the physical SCAN key (or either one of the side triggers).

   Data is auto-entered.

3) Submit data. Data is then passed to server.

4.3. NOTEWORTHY BROWSING FACTS

On the mobile computer, the COM ports assigned to stream decoded data can only be occupied by one application at a time. Avoid running two applications that access reader(s) at the same time, such as running MIRROR Browser and Reader Configuration concurrently.

SETTINGS

As MIRROR Browser aims at web-based data collection, it is equipped with the ability to launch and configure reader module(s) just as Reader Configuration does.

MIRROR Browser only works without another reader-controlling application running concurrently.

MIRROR Browser provides all its settings under **Menu** | **Option** | **Settings**, grouped on several tab pages. These settings adjust selected browser actions, screen presentation, configure reader(s), and set the symbologies to read.

Note: For different Windows Mobile or Windows Embedded Handheld device models, screen sizes and layout will differ according to the LCD panel resolution of each device.

IN THIS CHAPTER

5.1 Option Settings 32
5.2 Text Size Adjustment 52
5.3 Import/Export Settings 53
5.4 Using On-screen Keyboard 55
5.3 Import/Export Settings 53
5.6 Exit Application 57

5.1. OPTION SETTINGS

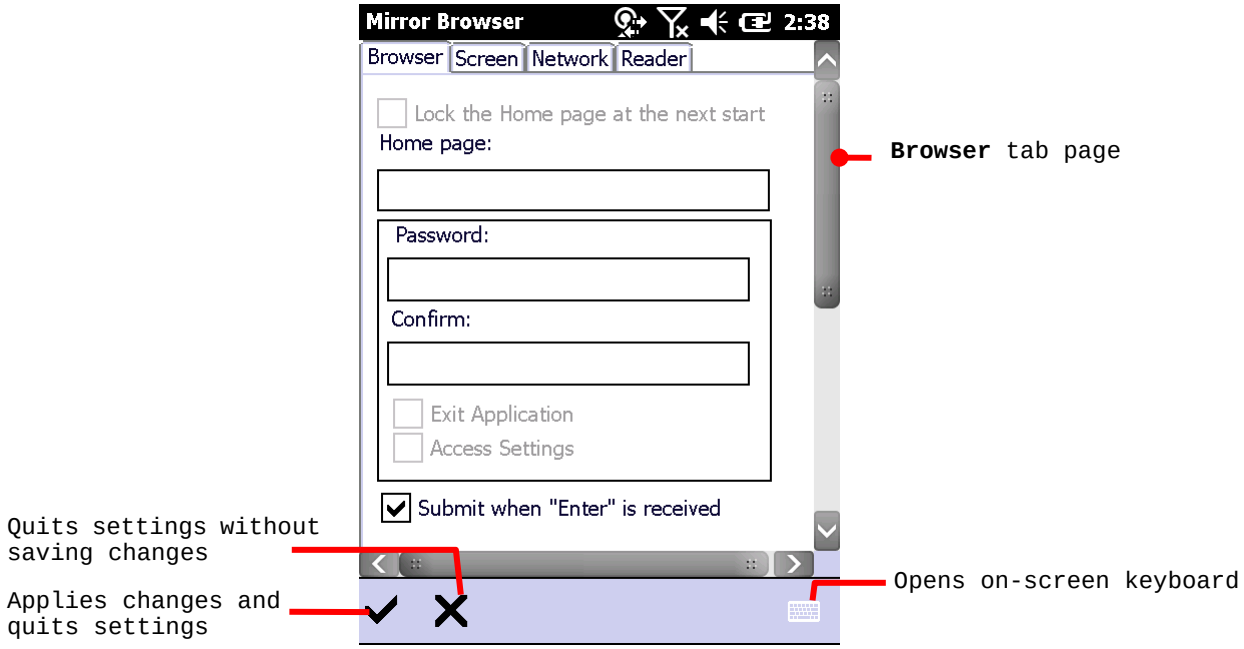
5.1.1. BROWSER SETTINGS

Browser settings adjust browsing preferences. They modulate how the Browser acts each time it is used.

To access Browser settings:

1) Open MIRROR Browser as described in Launch MIRROR Browser.
2) Tap **Menu** | **Option** | **Settings**.

   "Settings" screen opens showing **Browser** tab page.



Featured settings are:

Setting	Description
Lock the Home page at the next start	Makes address bar unavailable upon relaunch of MIRROR Browser. When set, browser will load the assigned home page when MIRROR Browser launches. ▸ This checkbox becomes available when a URL is entered in the "Home page" field. Note: To prevent unexpected user interference with the application, it is suggested to lock down the screen and apply password-protection.
Home page	Sets a home page to load in Browser view the next time MIRROR Browser opens. Enter the URL here.
Password	Puts up password protection against unauthorized access to the settings in **Menu** | **Option** | **Settings**. Enter a password here.
Confirm	Enter the password once more for confirmation.

Exit application	Select whether to put up password protection for exiting the application. If this checkbox is selected and a password is set in the text fields above, then you will need to enter the password in order to exit MIRROR Browser.
Submit when "Enter" is received	When selected, data on a web page form is submitted to the server once "Enter" is pressed. ▸ This is selected by default.

When finished with the settings, tap ✔ to apply changes and quit settings. Tap ✖ to quit settings without saving changes.

If a password is set up for accessing settings, when you tap **Menu** | **Option** | **Settings**, a password confirmation page will open with a blank field for you to enter the password. Tap ✔ after you have entered the password to exit the application, or tap ✖ to cancel.

GENERAL BROWSER SETTINGS

General browser settings are the settings available in the OS' inherent **Internet Explorer Mobile** only. As MIRROR Browser is based on the structure of **Internet Explorer Mobile**, some settings (the "general browser settings") such as Internet options need to be adjusted in **Internet Explorer Mobile** (hereinafter "IE Mobile").

To access general browser settings:

1) Open Start screen.
2) Tap IE Mobile icon .
3) Tap the bottom-right icon to display a row of tool icons along the bottom of the screen (if they do not appear in the first place).
4) Tap menu icon .

   IE Mobile's menu opens.



5) Tap **View**.

Featured settings in **View** submenu are:

Setting	Description
Text Size	Sets the size of displayed text in the Browser.
Enable ActiveX Controls	Enhances Browser with more interactivity and functionality such as animation or popup menus. ▸ ActiveX control is a reusable software component to add animation, multimedia effects, interactive objects and some other sophisticated applications to a web page.
Mobile	Views web pages designed for mobile devices.
Desktop	Views web pages designed for desktop computers.
Automatic	Automatically switches between viewing web pages designed for either desktop computers or mobile devices.

6) Adjust settings to best suit you needs.

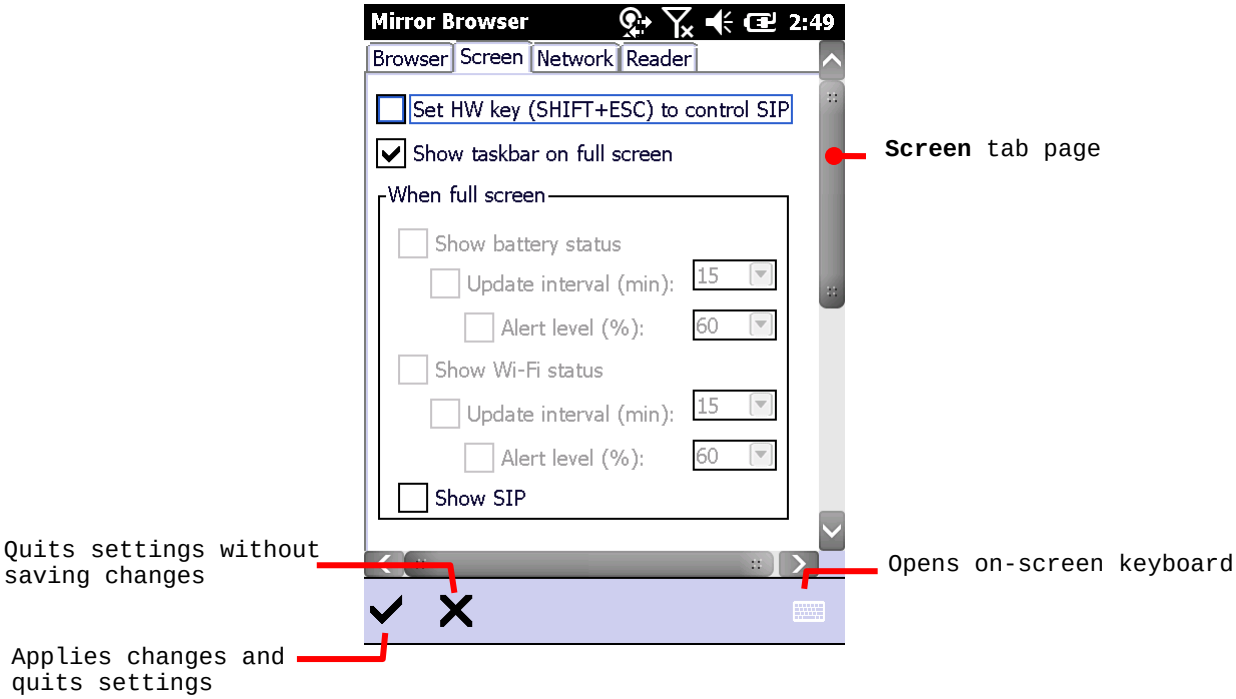
5.1.2. SCREEN SETTINGS

Screen settings select the way status information is displayed when Browser full screen mode is activated. Such information can be presented in the form of the OS taskbar, or as individual status icons for battery level and Wi-Fi signal strength.

To access **Screen** settings:

1) Open MIRROR Browser as described in Launch MIRROR Browser.
2) Tap **Menu** | **Option** | **Settings** | **Screen** tab page.

Settings page opens showing **Screen** tab page.

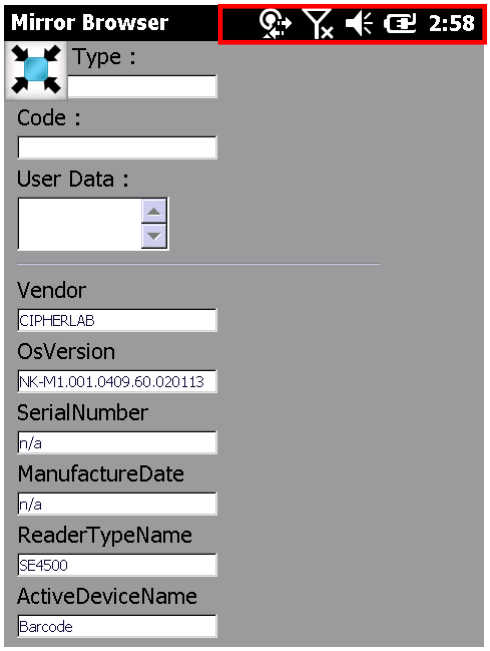


Featured settings are:

Setting	Description
Set HW key (SHIFT+ESC) to control SIP	Sets whether to enable SHIFT+ESC as the hotkey for opening/closing the on-screen keyboard. See Use Hotkeys. ▸ This is disabled by default.
Show taskbar on full screen	Sets whether to display the OS taskbar under full screen mode. The taskbar presents Wi-Fi signal, phone status, volume status, battery level, time information and more. ▸ When this is selected, the "When full screen" group box is automatically disabled and greyed out.
When full screen	This group box sets the items to display under Full Screen Mode, including the battery level icon, Wi-Fi signal strength icon and the SIP icon. The first two icons can be enabled on condition that the taskbar is not enabled. The SIP icon can be enabled regardless of whether the taskbar is shown on full screen.

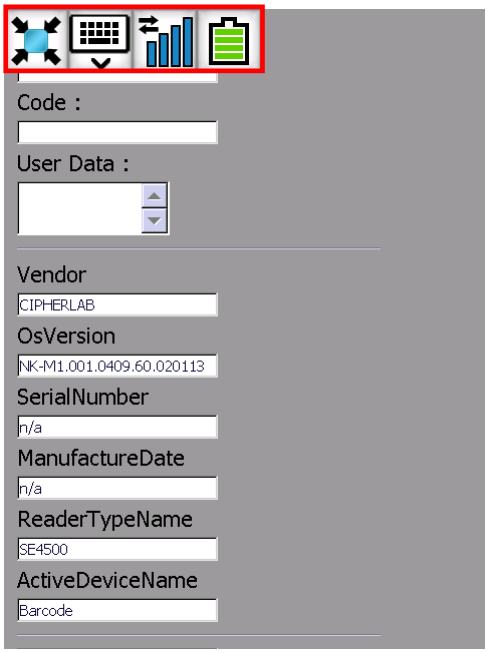
Setting	Description
Show battery status	▸ Select to enable the battery status icon.  ▸ Update Interval (min): Select to define frequency to update battery level. Options are: 1, 3, 5, 15, 30, 60 minutes. ▸ Alert Level (%): Select to define threshold for low battery alert to appear (by a pop-up dialog indicating the remaining battery level). Options are: 10, 20, 30, 40, 50, 60, 70, 80, 90%.
Show Wi-Fi status	▸ Select to enable the Wi-Fi status icon.  ▸ Update Interval (min): Select to define frequency to update battery level. Options are: 1, 3, 5, 15, 30, 60 minutes. ▸ Alert Level (%): Select to define threshold for low Wi-Fi signal alert to appear (by a pop-up dialog indicating the remaining Wi-Fi signal strength). Options are: 10, 20, 30, 40, 50, 60, 70, 80, 90%.
Show SIP	▸ Select to enable the SIP icon. 

When finished with the settings, tap ✔ to apply changes and quit settings. Tap ✖ to quit settings without saving changes.



When “Show taskbar on full screen” is selected, the OS taskbar is retained under full screen mode, thus delivering status information for various items.

- See Actions to take on Full Screen for how to operate the icons on the maximized screen.



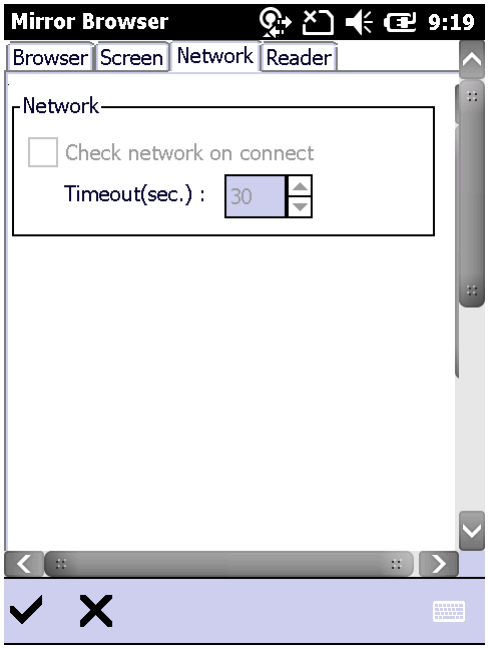
When “Show battery status”, “Show Wi-Fi status” and “Show SIP” are selected, status icons for Wi-Fi signal and battery level will appear under full screen mode. The taskbar is automatically hidden.

- See Actions to take on Full Screen for how to operate the icons on the maximized screen.

5.1.3. NETWORK SETTINGS

Network settings can check Wi-Fi network connection each time data is requested from the web server.

Note: Network settings are available only after an activation key is entered on the device. See Activate.

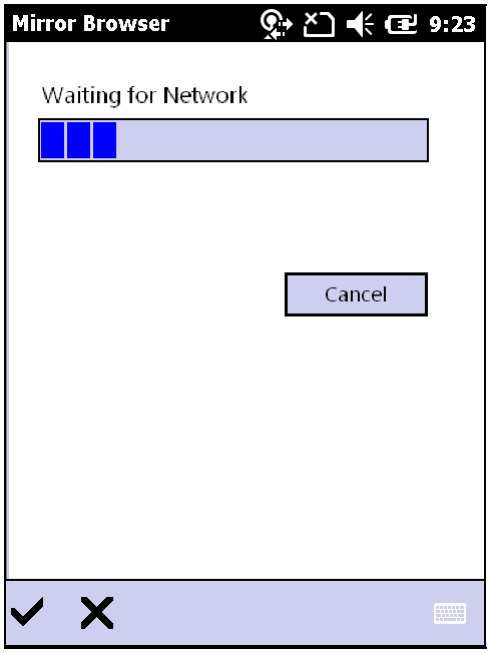


Setting	Description
Network	This group box performs network checking for Wi-Fi connection. Available on activated devices only.

Setting	Description
Check network on connect	Select to enable checking of Wi-Fi connection when data is requested from the web server, for instance, when a new URL is entered in the address bar, or when the web page is refreshed.
Timeout (sec)	Set the timeout period for network checking. The device will stop checking after the set amount of time has passed.

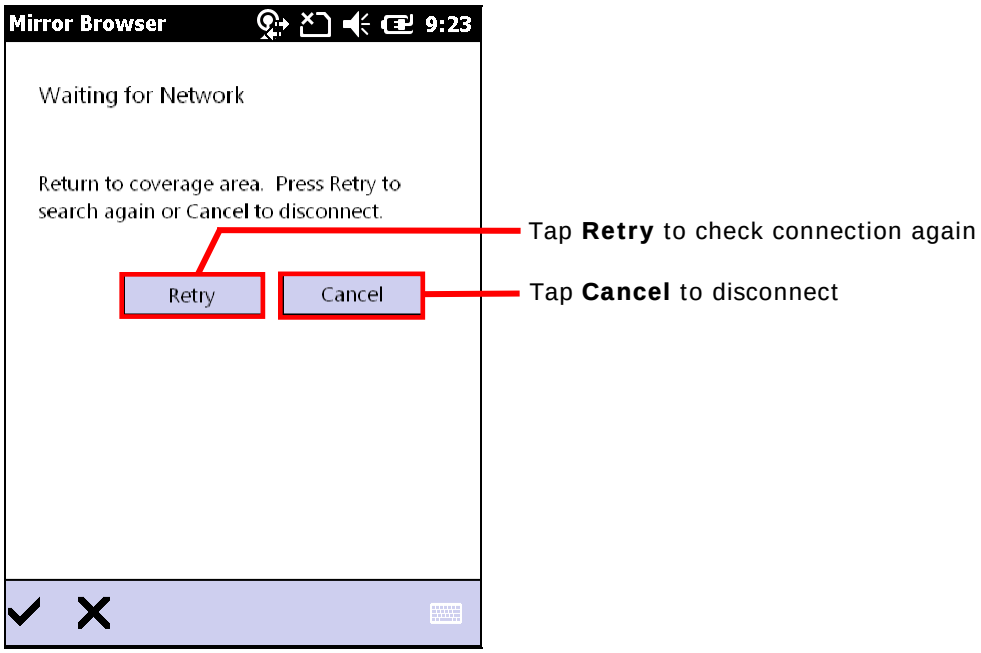
NETWORK CHECK

If network checking is enabled, when Browser view is directed towards a new URL, a "Waiting for Network" window pops up on the screen. The device begins checking for Wi-Fi connection status.



If a successful Wi-Fi connection is detected on the device, the "Waiting for Network" window closes and the target web page opens in Browser view.

When the timeout period ends and no Wi-Fi connection is established, a notification message appears asking you to return to the Wi-Fi coverage range for better connectivity. Once you have obtained a better Wi-Fi signal, tap **Retry** to test connectivity again.



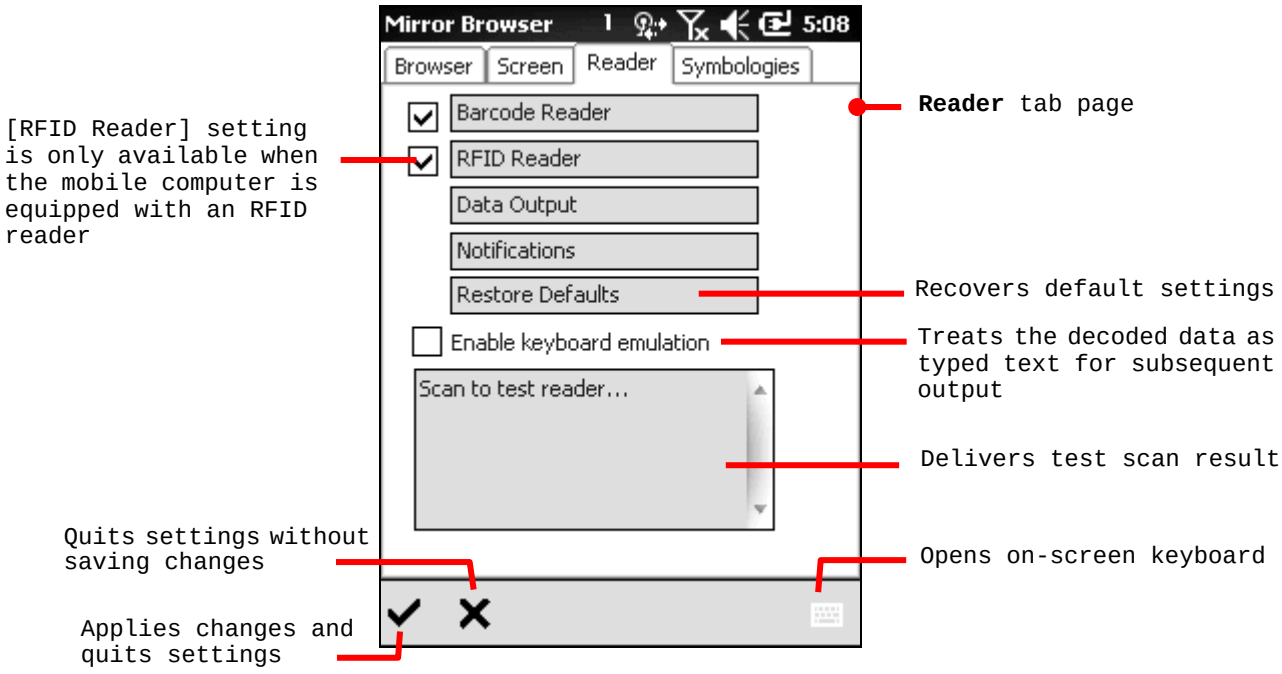
5.1.4. READER SETTINGS FOR CP30/CP50

Reader tab page presents reader settings. They configure reader module(s) to read RFID tags or printed barcodes the way you wish them to.

To open **Reader** tab page on CP30 and CP50:

1) Open MIRROR Browser as described in Launch MIRROR Browser.
2) Tap **Menu** | **Option** | **Settings** | **Reader** tab page.

   Reader tab page opens. This page is where all reader settings are accessed from except symbology settings. **Reader** tab page presents its settings with rectangle GUI buttons and check boxes. Most of the rectangle GUI buttons can be tapped to open their respective settings page. A field is also featured on **Reader** tab page to deliver test-scan results.



Featured settings are:

Setting	Facility	Description	Default
Barcode Reader	Check box	Enables/disables 1D (laser) reader or 2D imager.	Selected
	GUI button	Opens a page that sets barcode reading.	--
RFID Reader	Check box	Enables/disables RFID scan engine Available only when RFID reader is available	Selected
	GUI button	Opens a page that sets RFID reading. Available only when RFID reader is available	--
Data Output	GUI button	Opens a page that sets how to output decoded data, i.e. the output format.	--
Notifications	GUI button	Opens a page that sets how to give notice of good read	--
Restore Defaults	GUI button	Recovers default settings.	--
Enable keyboard emulation	Check box	Treats the decoded data as a typed text for subsequent output to the active application. Deselected by default, which means data is only read and decoded but isn't output.	Deselected
Scan to test reader…		This is a field to deliver user's test-scanned data including the following: Printed barcodes: barcode type, barcode data, user defined data RFID tags: tag type, UID, user defined data	--

BARCODE READER

Barcode Reader check box is selected by default, which means the printed barcode reader, either 1D (laser) reader or 2D imager, is enabled. Deselect the check box to disable the barcode reader. This is occasionally needed when an RFID reader coexists on the mobile computer and you only want the RFID tags to be read, lest printed barcodes should be read and decoded.

Save for a check box to enable/disable the printed barcode reader, **Barcode Reader** also features a settings page to control the 1D (laser) reader or 2D imager on board the mobile computer. Tap **Barcode Reader** button to open **Barcode Reader** settings page.

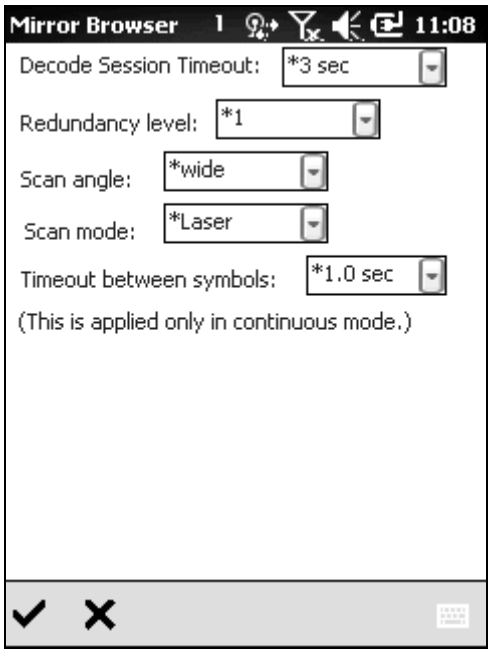
To open **Barcode Reader** settings page:

1) Open **Reader** tab page as described in Reader Settings for CP30/CP50.
2) Tap **Barcode Reader** button.

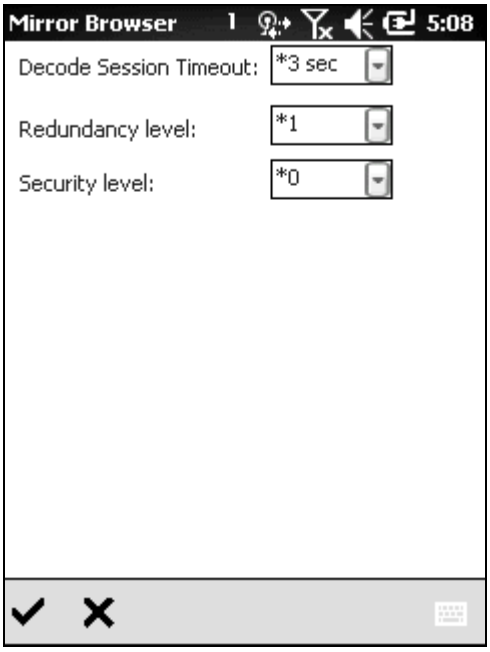
   Barcode Reader settings page opens in context with the barcode reader on board the mobile computer.

Featured settings for 1D (laser) reader and 2D imager are different:

Settings for 1D (laser) reader



Settings for 2D imager



Featured settings are separately explicated as follows:

1D (LASER) READER SETTINGS

Setting	Description	Default
Decode Session Timeout	Defines the maximum time for a decoding act. 1 through 9 sec configurable.	3 sec
Redundancy level	Sets how many successful readings are required before a barcode can be decoded.	1
Scan angle	Sets the angle to scan.	Wide
Scan mode	Sets the mode to scan. ▸ Options are Continuous and Laser modes.	Laser
Timeout between symbols	Sets the time for the barcode reader to resurrect its ability to once more decode a barcode it just decoded. ▸ Only available for Continuous mode.	1 sec

2D IMAGER SETTINGS

<table>
<tr><th>Setting</th><th>Description</th><th>Default</th></tr>
<tr><td>Decode Session Timeout</td><td>Defines the maximum time for a decoding act. 1 through 9 sec are configurable.</td><td>3 sec.</td></tr>
<tr><td>Redundancy level</td><td>Sets how many successful readings are required before a printed barcode can be decoded.</td><td>Level 1</td></tr>
<tr><td>Security Level</td><td>Sets the security level to ensure decoding accuracy considering the printed quality of barcodes such as Code 128, Code 93, and UPC/EAN. The higher the level is, the more security is ensured. Options are:
<table>
<tr><th>Level</th><th>Description</th></tr>
<tr><td>0</td><td>With this default, the scan engine is aggressive enough to decode most “in-spec” barcodes.</td></tr>
<tr><td>1</td><td>Select this level if misdecodes have occurred. This level fixes most misdecodes.</td></tr>
<tr><td>2</td><td>Select this level if Level 1 should fail to eliminate misdecodes.</td></tr>
<tr><td>3</td><td>Select this level if Security Level 2 should fail to prevent misdecodes.
As this level actually impairs the decoding ability of a scan engine to reduce misdecodes, it is recommended that you improve barcode print quality rather than use this level if this level should be needed.</td></tr>
</table></td><td>Level 0</td></tr>
</table>

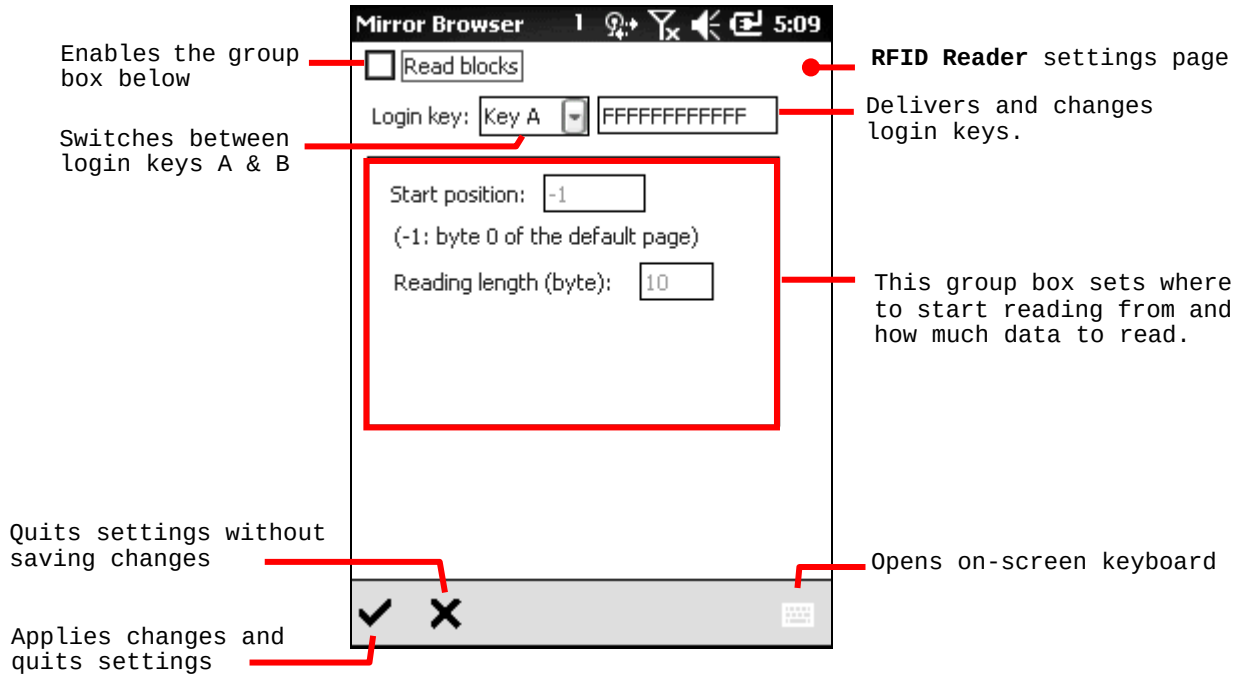
RFID READER (FOR CP50 ONLY)

If an RFID reader is detected when MIRROR Browser launches, RFID Reader setting becomes available.

RFID reader setting presents a check box and a rectangle button. The check box is selected by default. Deselect it to disable RFID reader. This is occasionally needed when you only want to read printed barcodes, lest RFID tags should be read and decoded. The GUI button can be tapped to open a settings page that controls the RFID reader on board.

To open **RFID Reader** settings page:

1) Open **Reader** tab page as described in Reader Settings for CP30/CP50.
2) Tap **RFID Reader** button to open RFID tab page.



Note: MIRROR Browser supports “reading” RFID tags only. To write data to RFID tags, use **Reader Configuration** instead.

Featured settings are:

Setting	Description
Read blocks	**Read blocks** is deselected by default. Select it to enable **Start position** and **Reading length (byte)** for RFID reading. Different RFID tags may have different default blocks (also “pages”), and different amount of bytes and number of blocks. See Appendix IV: HF RFID Reader to find out more about RFID Tag Default Block.
Login key	**Login key** deals with two keys that justify an RFID reader to access an RFID tag. **Login key** includes a drop box and a text field. The drop box switches between keys A and B while the field delivers and changes each key. In default state, both keys are a sequence of twelve “F” characters because these are Mifare tag’s factory default for keys A and B. ▸ Change the login key(s) by entering the prospective key(s) in the field. Each key must be a hex string with 12 bytes.
Start position	Defines where to start reading from. ▸ The default is -1, which means the RFID reader starts reading from byte 0 of the default block. ▸ Setting available only when **Read blocks** is selected
Reading length (byte)	Defines how many bytes of data to read. ▸ The default is 10 (bytes). ▸ **Reading length (byte)** is only available when **Read blocks** is selected

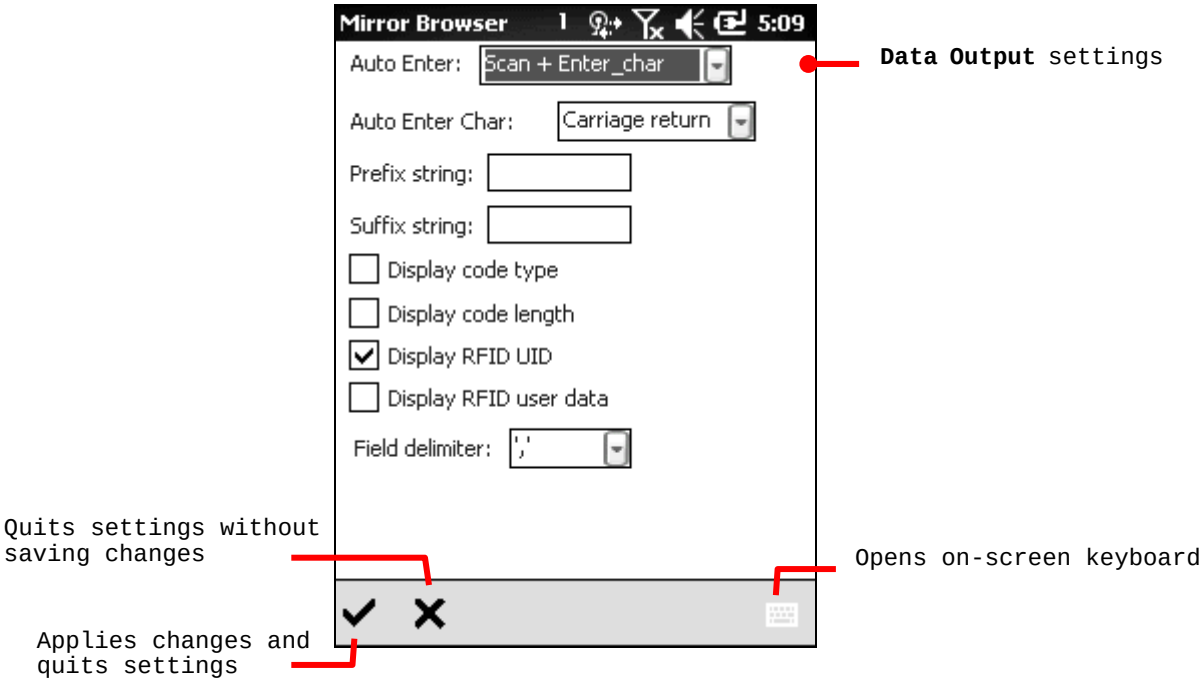
DATA OUTPUT

Data Output sets how to output decoded data. Simply put, it formulates the precise form to present the decoded data.

Data Output is available only when **Barcode Reader** is selected on **Reader** tab page.

To open **Data Output** settings page:

1) Open **Reader** tab page as described in Reader Settings for CP30/CP50.
2) Select **Barcode Reader** if you haven't.
3) Tap **Data Output** to open **Data Output** settings:



Featured settings are:

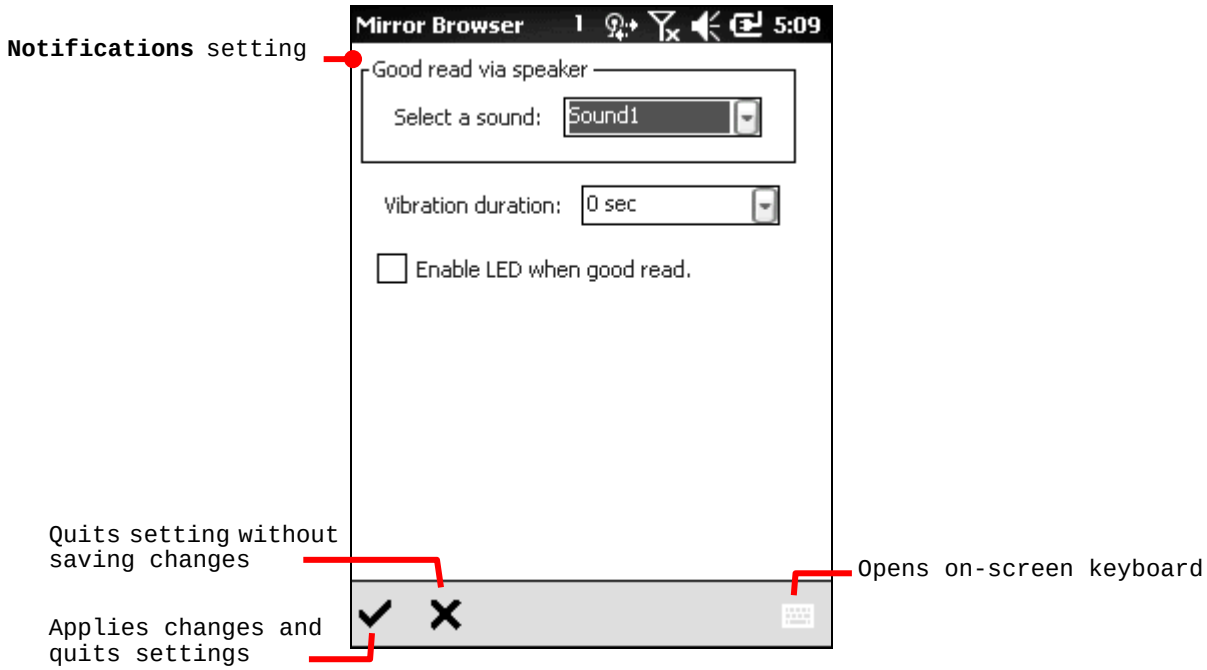
Setting	Description	Default
Auto Enter	Auto-adds an ENTER character before or after each scanning act. This function saves the trouble pressing [Enter] key to confirm each scan. Options are: • None • Scan + ENTER • ENTER + Scan	Scan + ENTER
Auto Enter Char	Adds a character before or after the decoded data. Options are: • None • Carriage Return • Tab • Space • Comma • Semicolon This setting is only available when **Auto Enter** is enabled.	Carriage Return
Prefix String	Affixes 1 to 9 characters to the left of the output data.	--
Suffix String	Affixes 1 to 9 characters to the right of the output data.	--
Display Code Type	Prefixes the output data with code type data.	Deselected (Disabled)
Display Code Length	Suffixes the output data with code length data.	Deselected (Disabled)
Display RFID UID	Outputs the UID of the RFID tag to read. • Setting available only when RFID reader is available	Selected (Enabled)
Display RFID User Data	Outputs the user data of the RFID tag to read. • Setting available only when RFID reader is available	Deselected (Disabled)
Field Delimiter	Sets the delimiter that separate an output data to pieces — • For printed barcodes: This delimiter separates code type, decoded barcode data, and code length (if code type and code length are applicable.) • For RFID tags: This delimiter separates UID and user data if they are applicable.	‘ , ’ (comma)

NOTIFICATIONS

Notifications controls how to give notice of successful decoding (good read), whether through audible, visible or tactile feedback.

To open **Notifications** settings:

1) Open **Reader** tab page as described in Reader Settings for CP30/CP50.
2) Tap Notifications button to open **Notifications** settings:



Settings on **Notifications** settings page are detailed as follows:

Setting	Descriptions	Default
Good read via speaker	Enables/disables the speaker to sound for good read. Sounds 1 to 9 are configurable.	Enabled & Sound 1
Vibration duration	Enables/disables tactile feedback (vibration) for good read and sets duration. Configurable duration is 0 to 5 seconds.	0 sec (=disabled)
Enable LED when good read	Enables/disables the mobile computer's Green LED to light for good read.	Deselected (= disabled)

RESTORE DEFAULTS

Restore Defaults restores all settings to default state.

To use **Restore Defaults**:

1) Open **Reader** tab page as described in Reader Settings for CP30/CP50.
2) Tap **Restore Defaults** button.

   Default settings are recovered.

ENABLE KEYBOARD EMULATION

Enable keyboard emulation enables treating decoded data as typed text for subsequent output to MIRROR Browser.

Enable Keyboard emulation is deselected by default, which means data is only read and decoded but isn't output.

To use Enable keyboard emulation:

1) Open **Reader** tab page as described in Reader Settings for CP30/CP50.
2) Select/deselect **Enable keyboard emulation**.

TEST SCAN FIELD

Reader tab page presents a field to deliver user's test-scan result that delivers decoded data and decoded barcode type (or RFID tag type and UID).

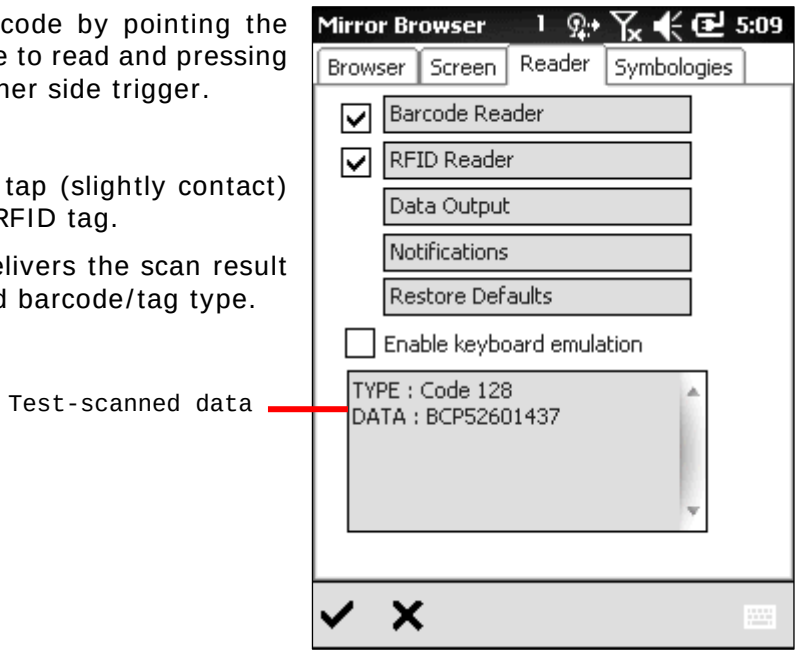
To use test-scan field:

1) Open **Reader** tab page as described in Reader Settings for CP30/CP50.
2) Test-scan any printed barcode by pointing the scan window at the barcode to read and pressing physical scan button or either side trigger.

   OR

   To test-read an RFID tag, tap (slightly contact) the scan window with the RFID tag.

   The test-scan field then delivers the scan result including decoded data and barcode/tag type.



SYMBOLOGIES SETTINGS

Symbologies tab page sets the symbologies to read. This page also enables/disables some feature(s) for a symbology to read such as:

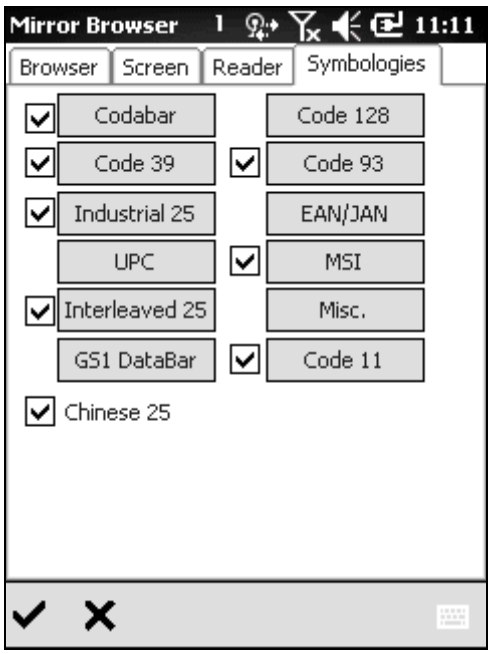
- Customize/transmit start/stop characters
- Verify and transmit check digits
- Enable/disable add-on digits
- Convert to another symbology
- Transmit symbology ID

To open **Symbologies** tab page:

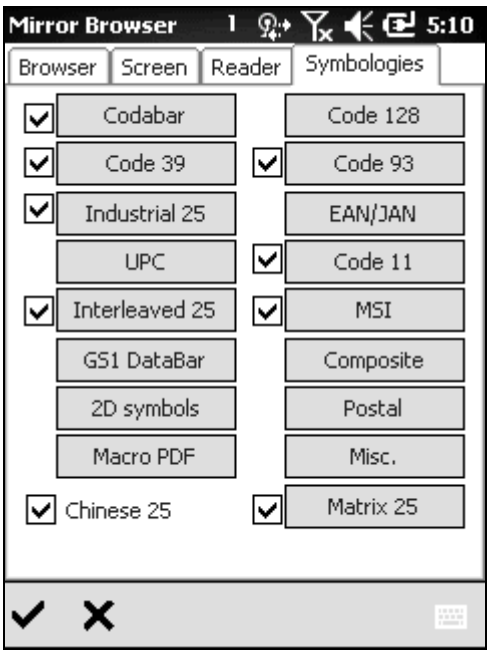
1) Open MIRROR Browser as described in Launch MIRROR Browser.
2) Tap **Menu** | **Option** | **Settings** | **Symbologies** tab page.

Symbologies tab page opens in context with the barcode reader on board the mobile computer. It delivers a settings page for either a 1D (laser) reader or 2D imager:

1D (laser) reader symbologies



2D imager symbologies



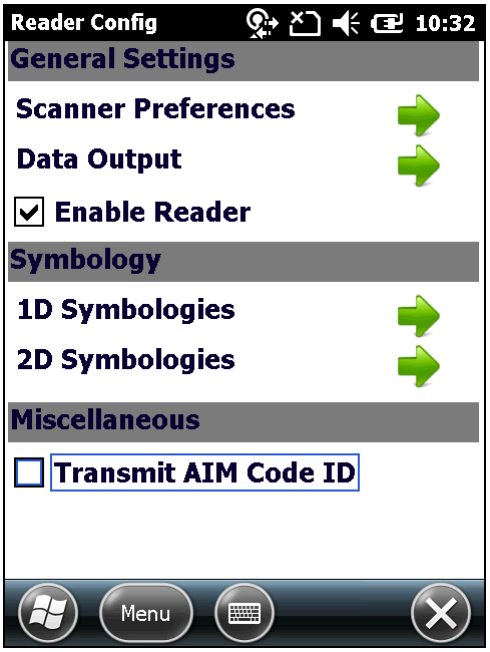
For details about the featured symbologies:

- See Appendix II: 1D (Laser) Reader Symbology Settings for CP30/CP50.
- See Appendix III: 2D Imager Symbology Settings for CP30/CP50.

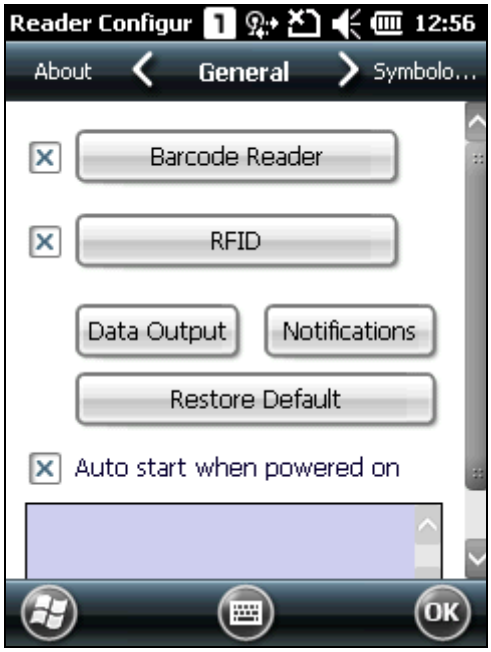
5.1.5. READER SETTINGS FOR CP60/9200

On CP60 and 9200, tapping the **Menu** | **Option** | **Settings** | **Reader** tab page will open the Reader Configuration utility on the mobile computer (same as tapping **Start Screen** | **Settings** | **System** | **Reader Configuration**). For the settings provided, see the Reference Manual of the specific mobile computer.

Reader settings for CP60 mobile computer



Reader settings for 9200 mobile computer



For details about the featured symbologies:

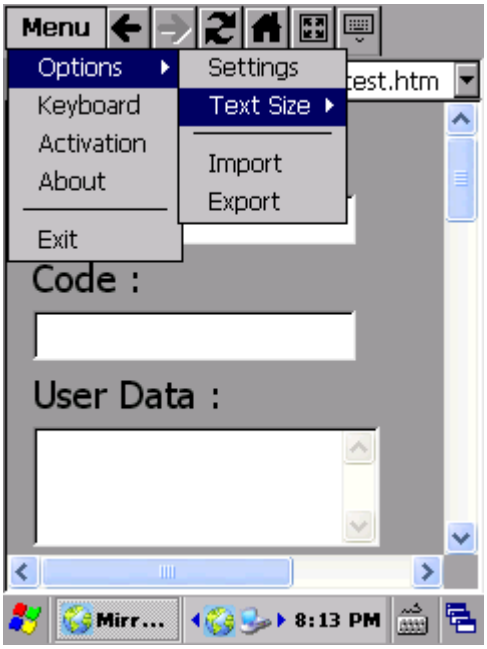
- See Appendix I: CCD Reader Symbology Settings for 9200
- See Appendix II: 1D (Laser) Reader Symbology Settings for CP30/CP50/9200 and Symbology Settings for CP60
- See Appendix III: 2D Imager Symbology Settings for CP60/9200

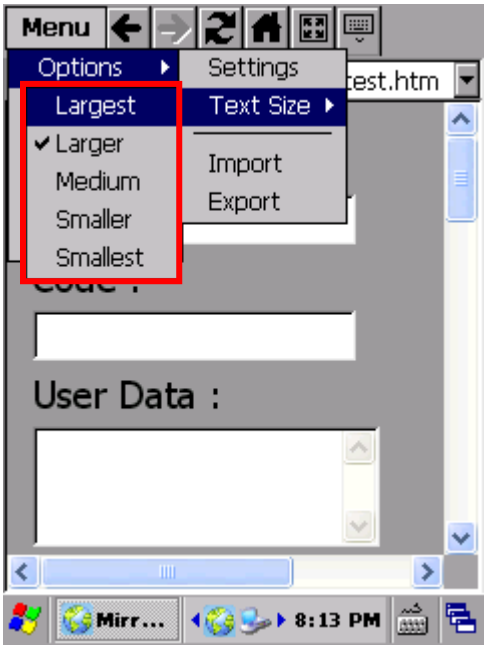
5.2. TEXT SIZE ADJUSTMENT

You may adjust the font size of the web page text which displays in Browser view. Five different font sizes are available.

To change font size:

1) Tap **Menu** | **Options** | **Text Size** to open a menu that shows five font sizes from large to small. The default font size of Browser view is **Larger**.





2) Tap one of the font sizes in the menu. The text on the web page opened in Browser view will change to the selected font size.

   You may also change font size in General Browser Settings.

5.3. IMPORT/EXPORT SETTINGS

MIRROR Browser supports importing and exporting settings on **Browser** and **Screen** tab pages as a separate profile (.ini format). Depending on the model type of your mobile computer, settings made to the reader might be stored in the profile as well.

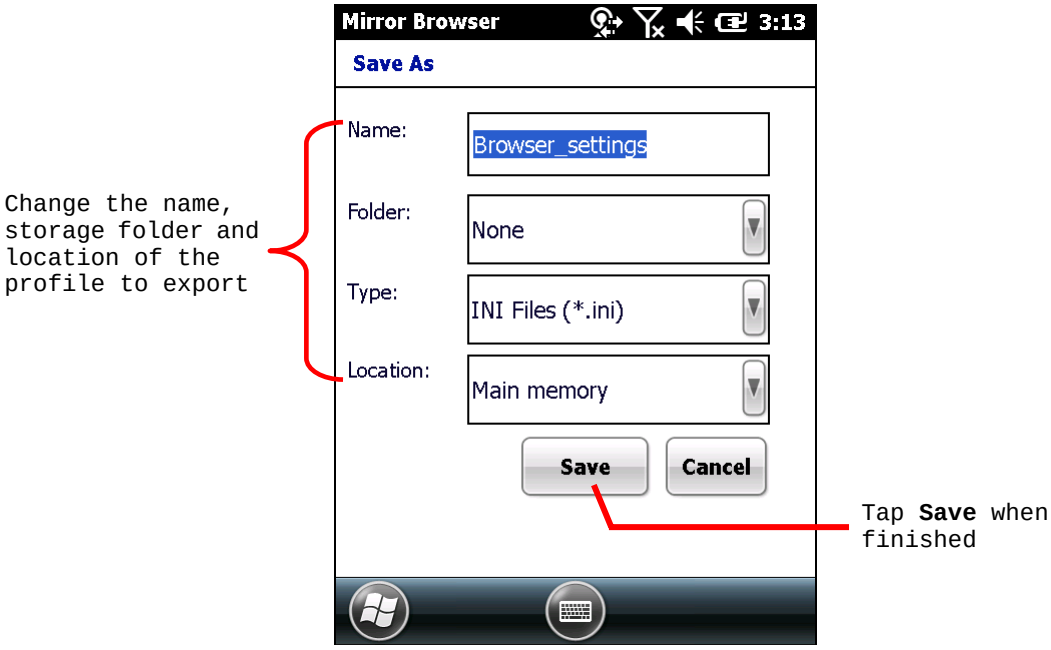
Previously exported profiles can be imported again on the mobile computer. Through exporting and importing profiles, you can save copies of different settings and select when to use them at your ease, or you can copy MIRROR Browser settings on one device and reproduce them on all other devices.

EXPORT SETTINGS

To export browser settings:

1) Open MIRROR Browser as described in Launch MIRROR Browser.
2) Tap **Menu** | **Option** | **Export**.

   An export page opens allowing you to enter the name of the profile, the destination folder and storage space.



3) After changing the settings according to your preferences, tap **Save** button to save the settings.

   The profile will be created in the specified destination.

Note: The **All Folders** directory refers to all folders under **My Device\My Documents**. If no subfolder is selected upon export, the exported file will be stored directly under this directory.

The following table explains whether or not Reader settings will be saved along with the exported profile.

Mobile Computer	Description
CP30/CP50	Stores Reader settings in the exported profile.
CP60/9200	Does not store Reader settings in the exported profile.

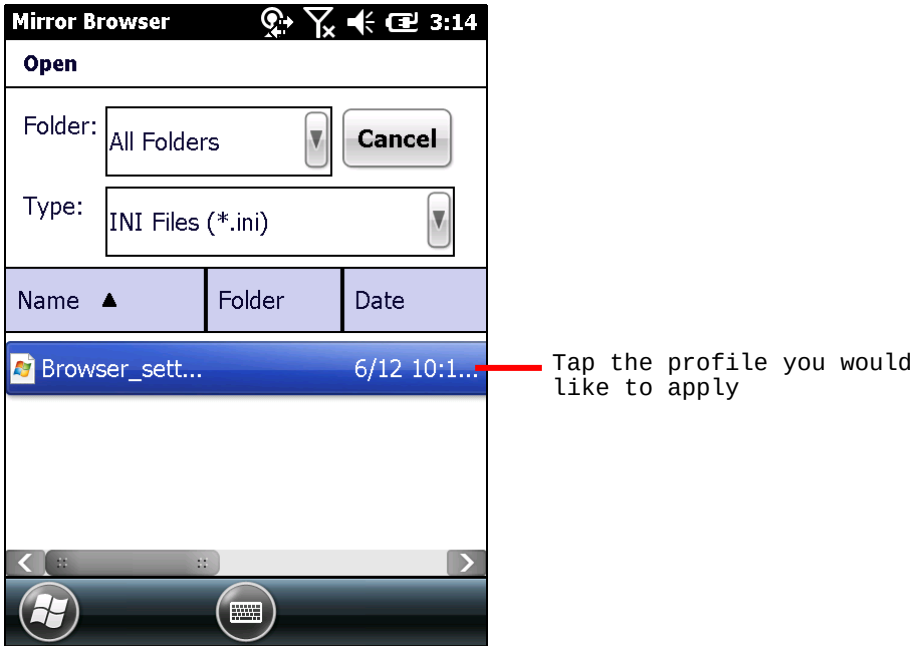
IMPORT SETTINGS

To import browser settings:

1) Obtain the profile and place it under **My Device\My Documents** directory on your mobile computer.
2) On the mobile computer, open MIRROR Browser as described in Launch MIRROR Browser.
3) Tap **Menu** | **Option** | **Import**.

   An import page opens allowing you to select the profile to apply.

   Tap the profile you would like to apply. Browser View will re-open with the selected profile applied as browser settings.



5.4. USING ON-SCREEN KEYBOARD

There are several ways to use the on-screen keyboard in Browser view:

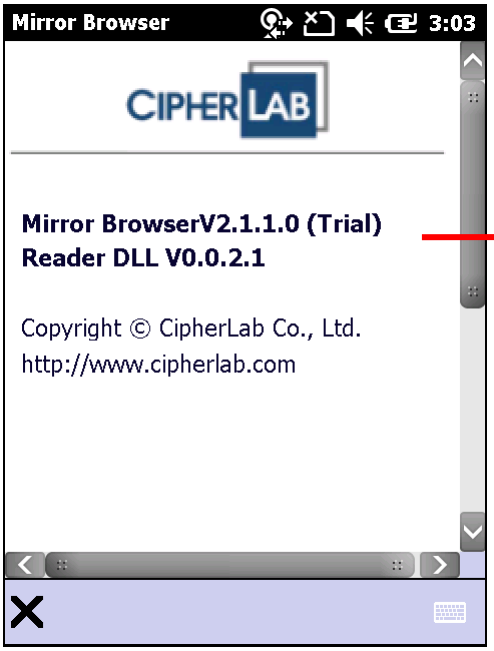
Open On-screen Keyboard	Close On-screen Keyboard
Tap the keyboard icon on the toolbar.	Tap the keyboard icon once more.
If Set HW key (SHIFT+ESC) to control SIP is selected under **Screen** Settings, press Shift + Esc to open the on-screen keyboard.	If Set HW key (SHIFT+ESC) to control SIP is selected under **Screen** Settings, press Shift + Esc to close the on-screen keyboard.
When Browser view is switched to full screen mode, double tap the SIP icon  to open the on-screen keyboard. • Show SIP must be selected for the SIP icon to be displayed on the maximized screen.	When Browser view is switched to full screen mode, double tap the SIP icon  to close the on-screen keyboard. • Show SIP must be selected for the SIP icon to be displayed on the maximized screen.

5.5. ABOUT

To view software version and copyright information:

1) Open MIRROR Browser as described in Launch MIRROR Browser.
2) Tap **Menu** | **About**.

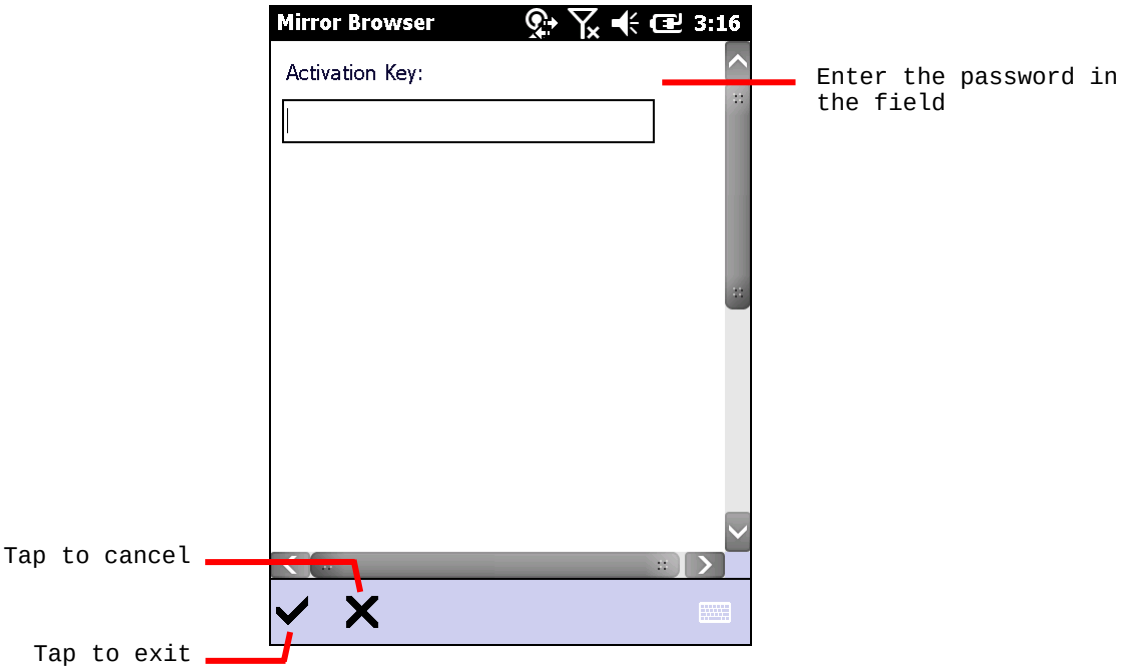
   Information about the software will be displayed onscreen.



5.6. EXIT APPLICATION

To exit MIRROR Browser, tap **Menu** | **Exit**.

If a password is set up for exiting the application in Browser Settings, a password inquiry page will open showing a blank field to enter the password. Tap ✔ after you have entered the password to exit the application, or tap ✖ to cancel and return to the previous screen.



DEVELOP YOUR HTML DOCUMENT

This chapter introduces the scripts to be embedded in your HTML document to load on the mobile computer, a client device. The embedded script links to the data collecting form consigned with decoded data from the decoder. “Scripts” are actually “programs” that execute when a HTML document loads.

A series of APIs that retrieve system information and reader information are also provided in this chapter to help you develop your own HTML document.

IN THIS CHAPTER

6.1 Data Collecting Script & Form 60
6.2 JavaScript APIs..... 61

6.1. DATA COLLECTING SCRIPT & FORM

The script is a data collecting script that comes with a data collecting form. Together they receive the decoded data coming from decoder and scan engines.

The data collecting script and form are provided in the sample page “test.htm”. To use the script & form:

1) Browse to where MIRROR Browser is installed on the mobile computer.

   If you haven’t selected an alternate location, MIRROR Browser is installed to the device’s internal storage at **My Device\Program Files\Mirror Browser**.

2) Locate test.htm. Open it with a text editor.

   The data collecting script is enclosed by the tags <SCRIPT> and </SCRIPT>, followed by a data collecting form enclosed in <table> and </table> tags.

3) Copy and paste the said script and form to your HTML document.

   The script and form will then work with the active scan engine(s) on the mobile computer to collect data. The following section How Data Collection Works on HTML Doc will explain how they work together.

Note this script relies on the physical SCAN key (or side triggers) to take action.

6.1.1. HOW DATA COLLECTION WORKS ON HTML DOC

The following text explains how data collection works on an HTML document, through the example of reading printed barcodes:

Once the physical Scan key (or side trigger key) is pressed, the scanning light beams and a barcode is read.

- If decoding succeeds, function **onScanBarcode()** in the script is invoked, and decoded data is passed to the arguments of **onScanBarcode()**.
- If decoding fails, function **onScanBarcodeError()** in the script is invoked, and error code is passed to **onScanBarcodeError()**.

6.2. JAVASCRIPT APIS

The JavaScript APIs herein enable your HTML document to retrieve system information and control reader module(s), speaker and vibrator.

Note all browser-triggered functions are JavaScript functions, and function names are case-sensitive.

6.2.1. SYSTEM INFORMATION

getManufactureDate	
Purpose	Gets mobile computer manufacturing date.
Syntax	Result = window.external.getManufactureDate()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	y=window.external.getManufactureDate();
	alert(y);
	}
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	The result is a string that delivers manufacturing date. This information is identical to the one displayed under **Start Screen** | **Settings** | **System** | **Device Information** on the mobile computer. On CP60, tap **Start** | **Settings** | **System** | **System Information** – **Manufacture Info**.

getOsVersion	
Purpose	Gets OS version information
Syntax	Result = window.external.getOsVersion()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test() {
	y=window.external.getOsVersion();
	alert(y);
	}
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	The result is a string that delivers OS image version. The resulted information is exactly as displayed under **Start Screen** | **Settings** | **System** | **Device Information** - **OS Version/Software Version**. On CP60, tap **Start** | **Settings** | **System** | **System Information** – Version.

getSerialNumber	
Purpose	Gets device serial number.
Syntax	Result = window.external.getSerialNumber()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test() {
	y=window.external.getSerialNumber();
	alert(y);
	}
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	The result is a string of device serial number. This information is exactly as delivered under **Start Screen** | **Settings** | **System** | **Device Information** | **Serial Number**. On CP60, tap **Start** | **Settings** | **System** | **System Information** - Manufacture Info - Shipping Number**.**

getVendor	
Purpose	Gets device manufacturer info.
Syntax	Result = window.external.getVendor()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test() {
	y=window.external.getVendor();
	alert(y);
	}
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	The result is a string of device manufacturer name (="CIPHERLAB"). This information is exactly as delivered under **Start Screen** | **Settings** | **System** | **Device Information** - **Manufacturer**. On CP60, tap **Start** | **Settings** | **System** | **System Information** - Manufacture Info - Manufacturer.

getWiFiStatus	
Purpose	Gets Wi-Fi status.
Syntax	Result = window.external.GetWiFiStatus()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test() {
	y=window.external.GetWiFiStatus();
	alert(y);
	}
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	The result is the Wi-Fi signal status, expressed as a value between 0 and 100. The higher the value, the greater the signal strength.

getPowerStatus	
Purpose	Gets power status.
Syntax	Result = window.external.GetPowerStatus()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test() {
	y=window.external.GetPowerStatus();
	alert(y);
	}
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	The result is the remaining battery capacity, expressed as a value between 0 and 100. The higher the value, the more the remaining battery capacity. .

6.2.2. SYMBOLOGY

enableSymbology

Purpose: Enables symbology by name.

Syntax: window.external.enableSymbology (*name*, *enable*)

Parameters:

name

Symbology key. Refer to the Symbology Key Table.

enable

[in] Integer to enable/disable the symbology.

True	Enable the symbology
False	Disable the symbology

Example:

```
<HTML>
<BODY onload=”test()”>
<SCRIPT>
    Var name = ‘Codabar’
    Var enabled = true;
    window.external.EnableSymbology(name, enable);
…
</SCRIPT>
</BODY>
</HTML>
```

IsSymbologyEnabled

Purpose: Gets status of symbology by name.

Syntax: window.external.enableSymbology (*name*)

Parameters:

name

Symbology key. Refer to the Symbology Key Table.

Example:

```
<HTML>
<BODY onload=”test()”>
<SCRIPT>
    Var name = ‘Codabar’
    Var enable = window.external.IsSymbologyEnabled(name);
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value: If successful, it returns a string to indicate symbology status:

True	Symbology is enabled
False	Symbology is disabled

SYMBOLOGY KEY TABLE

The table below lists which symbologies can be configured using Javascript APIs on each type of reader. Refer to Scan Engine Settings to see which types of readers are available on each device.

1D Symbologies		1D Reader			2D Imager	
Symbologies	API Key	SM1	S955	SE965	SE4500	PL4507
Codabar	Codabar	✓	✓	✓	✓	✓
Code11	Code11	✗	✓	✓	✓	✓
Code39	Code39	✓	✓	✓	✓	✓
Code93	Code93	✓	✓	✓	✓	✓
Code128	Code128	✓	✓	✓	✓	✓
	GS1128	✓	✓	✓	✓	✓
	ISBT128	✓	✓	✓	✓	✓
Code 2 of 5	Chinese25	✗	✓	✓	✓	✓
	Discrete25	✓	✓	✓	✓	✓
	Interleaved25	✓	✓	✓	✓	✓
	Matrix25	✗	✗	✗	✓	✓
Composite	CompositeCCAB	✗	✗	✗	✓	✓
	CompositeCCC	✗	✗	✗	✓	✓
	CompositeTLC39	✗	✗	✗	✓	✓
GS1 DataBar	GS1DataBar14	✓	✓	✓	✓	✓
	GS1DataBarLimited	✓	✓	✓	✓	✓
	GS1DataBarExpanded	✓	✓	✓	✓	✓
Korean 3 of 5	Korean3of5	✗	✗	✗	✓	✓
MSI	MSI	✓	✓	✓	✓	✓
Postal Code	AustralianPostal	✗	✗	✗	✓	✓
	JapanPostal	✗	✗	✗	✓	✓

	NetherlandKIXCode	✘	✘	✘	✓	✓
	USPostnet	✘	✘	✘	✓	✓
	USPlanet	✘	✘	✘	✓	✓
	UKPostal	✘	✘	✘	✓	✓
UPC/EAN	Ean8	✓	✓	✓	✓	✓
	Ean13	✓	✓	✓	✓	✓
	UPCA	✓	✓	✓	✓	✓
	UPCE	✓	✓	✓	✓	✓
	UPCE1	✓	✓	✓	✓	✓
2D Symbologies		**1D Reader**			**2D Imager**	
API key		**SM1**	**S955**	**SE965**	**SE4500**	**PL4507**
Aztec		✘	✘	✘	✓	✓
DataMatrix		✘	✘	✘	✓	✓
Maxicode		✘	✘	✘	✓	✓
MicroPDF417		✘	✘	✘	✓	✓
MicroQR		✘	✘	✘	✓	✓
PDF417		✘	✘	✘	✓	✓
QRCode		✘	✘	✘	✓	✓

6.2.3. READER

GetReaderType	
Purpose	Gets the reader type on the device.
Syntax	window.external.GetReaderType()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	window.external.GetReaderType();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	Refer to the Programming Guide of each specific device for details on reader types.

GetBcReaderType	
Purpose	Gets the barcode reader type on the device.
Syntax	window.external.GetBcReaderType()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	window.external.GetBcReaderType();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	Refer to the Programming Guide of each specific device for details on reader types.

GetRFIDReaderType	
Purpose	Gets the RFID reader type on the device.
Syntax	window.external.GetRFIDReaderType()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	  window.external.GetRFIDReaderType();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	Refer to the Programming Guide of each specific device for details on reader types.

GetReaderTypeName	
Purpose	Gets the name of the reader type on the device.
Syntax	window.external.GetReaderTypeName()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	  window.external.GetReaderTypeName();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Return Value	If successful, it returns a string to indicate the reader type name:

SE955E
SE955I
SE4500
SE965
SE4500+PL4507
Intermec EX25
SE1524
SM1
SE950
SE4407
SE4507
RFID

GetActiveDevice

Purpose	Gets the active reader type(s).
Syntax	window.external.GetActiveDevice()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test() {
	window.external.GetActiveDevice();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Return Value	If successful, it returns a value to indicate the active reader type(s):

0	No Active Device
7	Barcode Reader
32	RFID Reader
255	All devices

GetActiveDeviceName	
Purpose	Gets the name of the active reader type(s).
Syntax	window.external.GetActiveDeviceName()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	  window.external.GetActiveDeviceName();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Return Value	If successful, it returns a string to indicate the active reader type(s): Barcode RFID Barcode, RFID none

enableBarcodeScanner

Purpose	Enables barcode reader.
Syntax	window.external.enableBarcodeScanner (*enable*)
Parameters	*enable* [in] Integer to turn on/off barcode reader.

True	Turns on barcode reader
False	Turns off barcode reader

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.enableBarcodeScanner(true);    // enable the
scanner
}
…
</SCRIPT>
</BODY>
</HTML>
```

IsBarcodeScannerEnabled

Purpose	Gets barcode reader status.
Syntax	window.external.IsBarcodeScannerEnabled ()
Parameters	None

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.IsBarcodeScannerEnabled();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value: If successful, it returns a string to indicate barcode reader status:

True	Barcode reader is on
False	Barcode reader is off

enableRFIDScanner

Purpose	Manipulates RFID reader.
Syntax	window.external.enableRFIDScanner (*enable*)
Parameters	*enable* [in] Integer that determines whether to turn on RFID reader or not.

True	Enables
False	Disables

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.enableRFIDScanner(true);        // enable the RFID scanner
}
…
</SCRIPT>
</HTML>
```

IsRFIDScannerEnabled

Purpose	Gets RFID reader status.
Syntax	window.external.IsRFIDScannerEnabled ()
Parameters	None

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.IsRFIDScannerEnabled();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value: If successful, it returns a string to indicate RFID reader status:

True	RFID reader is on
False	RFID reader is off

ReadFromDevice	
Purpose	Reads the symbologies from the device.
Syntax	window.external.ReadFromDevice()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	window.external.ReadFromDevice();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

WriteToDevice	
Purpose	Writes the symbologies to the device.
Syntax	window.external.WriteToDevice()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	window.external.WriteToDevice();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Remarks	If EnableAutoWriteToDevice is set as “false”, use this function to have reader settings written to the device when finished calling all other functions.

EnableAutoReadFromDevice

Purpose: Enables auto reading of symbologies from the device.

Syntax: window.external.EnableAutoReadFromDevice (*enable*)

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

enable

[in] Integer to turn on/off auto reading of symbologies from the device.

True	Enables auto read
False*	Disables auto read

Example:

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.EnableAutoReadFromDevice(true);     // enable
auto read
}
…
</SCRIPT>
</BODY>
</HTML>
```

IsAutoReadFromDeviceEnabled

Purpose: Gets auto read status (auto reading of symbologies from the device).

Syntax: window.external.IsAutoReadFromDeviceEnabled()

Parameters: None

Example:

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.IsAutoReadFromDeviceEnabled();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value: If successful, it returns a string to indicate auto read status:

True	Auto read is on
False	Auto read is off

EnableAutoWriteToDevice

Purpose	Enables auto writing of symbologies to the device.
Syntax	window.external.EnableAutoWriteToDevice (*enable*)
Parameter	*enable* [in] Integer to turn on/off auto writing of symbologies to the device.

True	Enables auto write
False	Disables auto write

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.EnableAutoWriteToDevice(true);        // enable
auto write
}
…
</SCRIPT>
</BODY>
</HTML>
```

Remarks

- If only a few reader functions are called, it is recommended to set this function as "true". When each function is called, settings will be written to the device automatically.
- If more than several APIs are called (for instance, more than 10), it is recommended to set this function as "false"; instead, when finished calling all functions, call WriteToDevice to have the settings written to the device.

IsAutoWriteToDeviceEnabled

Purpose	Gets auto write status (auto writing of symbologies to the device).
Syntax	window.external.IsAutoWriteToDeviceEnabled()
Parameters	None

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.IsAutoWriteToDeviceEnabled();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value: If successful, it returns a string to indicate auto write status:

True	Auto write is on
False	Auto write is off

onScanBarcode

Purpose	Decodes a barcode.
Syntax	function onScanBarcode(*codeType*, *code*)
Parameters	*codeType*, *code* [in] The decoded code type and barcode will be passed to the arguments. The argument codeType is a hex string of fixed length (= 2 characters). As per symbologies supported on the scan engine, code types are listed below:

Scan engine: 1D CCD/Laser or 2D Imager

Code Type	Symbology
'40'	ISBT 128
'41'	Code 39
'42'	Italian Pharmacode (Code 32)
'43'	French Pharmacode (CIP 39)
'44'	Industrial 25
'45'	Interleaved 25
'46'	Matrix 25
'47'	Codabar (NW7)
'48'	Code 93
'49'	Code 128
'4A'	UPC-E0
'4B'	UPC-E0 with Addon 2
'4C'	UPC-E0 with Addon 5
'4D'	EAN-8
'4E'	EAN-8 with Addon 2
'4F'	EAN-8 with Addon 5
'50'	EAN-13 (also UPC-A on CCD/Laser scan engine)
'51'	EAN-13 with Addon 2
'52'	EAN-13 with Addon 5
'53'	MSI
'54'	Plessey
'55'	GS1-128 (EAN-128)
'56'	Undefined
'57'	Undefined
'58'	Undefined
'59'	Undefined
'5A'	Telepen

'5B'	GS1 DataBar Omnidirectional (RSS-14)
'5C'	GS1 DataBar Limited (RSS Limited)
'5D'	GS1 DataBar Expanded (RSS Expanded)
'5E'	UPC-A
'5F'	UPC-A with Addon 2
'60'	UPC-A with Addon 5
'61'	UPC-E1
'62'	UPC-E1 with Addon 2
'63'	UPC-E1 with Addon 5
'64'	TLC 39 (TCIF Linked Code 39)
'65'	Trioptic (Code 39)
'66'	Bookland (EAN)
'67'	Code 11
'68'	Code 39 Full ASCII
'69'	IATA[Note] (Code 25 used on flight tickets)
'6A'	Industrial 25 (Discrete 25)
'6B'	PDF417
'6C'	MicroPDF417
'6D'	Data Matrix
'6E'	Maxicode
'6F'	QR Code
'70'	US Postnet
'71'	US Planet
'72'	UK Postal
'73'	Japan Postal
'74'	Australian Postal
'75'	Dutch Postal
'76'	Composite Code
'77'	Macro PDF
'78'	Coupon Code
'79'	Chinese 25
'7A'	Aztec
'7B'	MicroQR
'7C'	USPS 4CB / One Code / Intelligent Mail
'7D'	UPU FICS Postal
'7E'	Macro MicroPDF417

Example

```
<HTML>
<SCRIPT>
function onScanBarcode(codeType, code) {
if(codeType == '41') {
       t1.value = "Code 39";
       t2.value = code;
       t3.value = "";
   }
   else if(codeType == '47') {
       t1.value = "Codabar (NW7)";
       t2.value = code;
       t3.value = "";
   }
}
</SCRIPT>
</HTML>
```

Remarks

Once a barcode reading succeeds, the HTML document invokes this function.

onScanRFID

Purpose	Decodes an RFID tag.
Syntax	function onScanRFID (*codeType*, *code*, *userData*)
Parameters	*codeType*, *code*, *userData* [in] The decoded code type (= tag type), code (= UID) and user data will be passed to the arguments. The tag types are listed below:

Tag Type	RFID Tag / Standard
'I'	Icode
'M'	Mifare Ultralight ISO 14443A
'S'	SR176
'T'	Tagit
'V'	ISO 15693
'Z'	ISO 14443B

Example

```
<HTML>
<SCRIPT>
function onScanRFID(codeType, code, userData) {
    switch(codeType) {
    case 'I':
        t1.value = "Icode";
        t2.value = code;
        t3.value = userData;
        break;
    default:
        t1.value = codeType;
        t2.value = code;
        t3.value = userData;
        break;
    }
}
…
</SCRIPT>
</HTML>
```

Remarks: When RFID reading succeeds, Browser invokes this function.

playSound

Purpose: Sounds a .wav file.

Syntax: window.external.playSound (*soundIdx*, *path*)

Parameters:

soundIdx

[in] An integer that specifies which .wav file to sound:

0	Mutes
1~9	Sounds .wav file 1 ~ 9
-1	Sounds user-defined .wav file

path

[in] A string that specifies the file path if "-1" is specified in the first parameter.

Example:

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.playSound(1, ' ');
}
…
</SCRIPT>
</BODY>
</HTML>
```

Remarks:

This function sounds a .wav file.

Embedded are 9 .wav files to be selected by the index number passed to parameter soundIdx.

Play your own .wav file by passing "-1" to parameter soundIdx and assigning the file path in the second parameter.

startVibration

Purpose: Controls the built-in vibrator.

Syntax: window.external.startVibration (*enable*)

Parameters:

enable

[in] An integer that specifies if to turn on vibrator and how long it vibrates.

0	Turns off
Non-zero	Turns on and vibrates for the specified time, by unit of second.

Example:

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.startVibration(1);     // to vibrate for 1 second
}
…
</SCRIPT>
</BODY>
</HTML>
```

EnableKeyboardEmulation

Purpose	Enable keyboard emulation on the device.
Syntax	window.external.EnableKeyboardEmulation (*enable*)
Parameters	*enable* [in] Integer to turn on/off keyboard emulation.

True	Turns on keyboard emulation
False	Turns off keyboard emulation

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.EnableKeyboardEmulation(true);        // enable keyboard emulation
}
…
</SCRIPT>
</BODY>
</HTML>
```

Remarks: To collect barcode or RFID data in the data collecting script, this function must be set as "true". However, if Javascript APIs are used to collect barcode or RFID data, this function must be set as "false".

IsKeyboardEmulationEnabled

Purpose	Gets keyboard emulation status.
Syntax	window.external.IsKeyboardEmulationEnabled()
Parameters	None

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
   window.external.IsKeyboardEmulationEnabled();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value: If successful, it returns a string to indicate keyboard emulation status:

True	Keyboard emulation is on
False	Keyboard emulation is off

SetPrefixCode	
Purpose	Sets the prefix code.
Syntax	window.external.SetPrefixCode (*sCode*)
Parameters	*sCode* [in] An integer that specifies the prefix code to add to the left of the decoded data.

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test() {
	window.external.SetPrefixCode(XYZ[);
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

GetPrefixCode	
Purpose	Gets the prefix code.
Syntax	window.external.GetPrefixCode()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test() {
	window.external.GetPrefixCode();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

SetSuffixCode	
Purpose	Sets the suffix code.
Syntax	window.external.SetSuffixCode (*sCode*)
Parameters	*sCode* [in] An integer that specifies the suffix code to add to the right of the decoded data.

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	window.external.SetSuffixCode(XYZ[);
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

GetSuffixCode	
Purpose	Gets the suffix code.
Syntax	window.external.GetSuffixCode()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test() {
	window.external.GetSuffixCode();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

6.2.4. RFID READER

ReadRfidData

Purpose

Reads RFID data by tapping a GUI button (without intercepting WM_DECODEDATA).

Syntax

window.external.ReadRfidData (*nType*, *nStartBlock*, *nReadLen*, *nTimeout*, *sLoginKey*, *nLoginKeyType*)

Parameters

nType

A value that specifies which data to read (UID or encoded data).

Name	Value
READ_UID	1
READ_DATA	2

nStartBlock

A value that specifies the byte or block to start reading data from.

-1 Means to start reading from the default block (of an RFID tag).

Tabulated below is a number of RFID tags' default block to start reading data from:

Tag Type	Standard	Default Start Block
Mifare	(ISO 14443A)	**4**
SR176	(ISO 14443B)	**4**
ICODE SLI	(ISO 15693)	**3**
LRI512	(ISO 15693)	**0**
SRF55VxxP	(ISO 15693)	**3**
EM4135	(ISO 15693)	**0**
Tag-it	(ISO 15693)	**0**
Others	(ISO 15693)	**0**
ICODE	(Phillips)	**5**

nReadLen

Number of bytes to read. As different tag types may have different numbers of blocks allowed to be read, when you pass a value greater than allowed, it auto-adjusts itself to the maximum value allowed on a tag.

nTimeout Reserved

sLoginKey

A key to access a specific RFID tag such as Mifare Standard 1K/4K or SLE66R35 tags.

Such key must be a hex string of 12 bytes length.

- This login key applies to all the sectors covered by the range formulated by "nStart" and "nReadLen".

LoginKey Example	Key String
0	FFFFFFFFFFFF
“F1F2F3F4F5F6”	F1F2F3F4F5F6

nLoginKeyType

A value that identifies the login key.

0 *	Key A
1	Key B

Example

```
<HTML>
<BODY onload=”test()”>
<SCRIPT>
function test()
{
    var nType               = 1;//1=READ_UID;2=READ_DATA;
    var nStartBlock         = -1;
    var nReadLen     = 10;
    var nTimeout     = 3;
    var sLoginKey    = 'FFFFFFFFFFFF';
    var nLoginKeyType       = 0;

    var nReadRfidData =
window.external.ReadRfidData(nType,nStartBlock,nReadLen,nTimeout,sL
oginKey,nLoginKeyType);
    alert("ReadRfidUID="+nReadRfidData);

    If(nType==1) {
        var sCardUid=window.external.GetCardUid ();
    }
    else {
        var sCardData=window.external.GetCardData();
    }
}
…
</SCRIPT>
</BODY>
</HTML>
```

Other Info

Name	Type	Description
GetCardUid	String	Returns card UID
GetCardData	String	Returns card data

Return Value

If successful, it returns the number of bytes obtained (excluding terminating null character).

Otherwise, it returns a negative value to indicate an error condition:

-101	E_READER_NOT_INIT
-272	E_TAG_READ_FAILED
-273	E_TAG_VALUE_OVER_RANGE
-274	E_TAG_INVALID_LENGTH
-278	E_TAG_NO_TAG
-280	E_TAG_CANNOT_READ
-281	E_TAG_READ_TIMEOUT

WriteRfidData

Purpose

Writes RFID data to an RFID tag upon pressing a physical trigger.

Syntax

window.external.WriteRfidData(*sData*, *nStartBlock*, *nWriteLen*, *nTimeout*, *nMode*, *sLoginKey*, *nLoginKeyType*)

Parameters

sData

Writes data.

nStartBlock

A value that specifies the byte or block to start reading data from.

-1 Means to start reading from the default block (of an RFID tag).

Tabulated below is a number of RFID tags' default block to start reading data from:

Tag Type	Standard	Default Start Block
Mifare	(ISO 14443A)	**4**
SR176	(ISO 14443B)	**4**
ICODE SLI	(ISO 15693)	**3**
LRI512	(ISO 15693)	**0**
SRF55VxxP	(ISO 15693)	**3**
EM4135	(ISO 15693)	**0**
Tag-it	(ISO 15693)	**0**
Others	(ISO 15693)	**0**
ICODE	(Phillips)	**5**

nWriteLen

Maximum number of bytes to write. As the number of blocks allowed to be written may vary from tag type to tag type, when you pass a value greater than allowed, it auto-adapts to the maximum value allowed on a tag.

nTimeout Reserved

nMode

A value that specifies writing mode.

0	Starts writing immediately.
Non-zero	Starts writing upon pressing the trigger.

sLoginKey

A key to access a specific RFID tag such as Mifare Standard 1K/4K or SLE66R35 tags.

Such key must be a hex string of 12 bytes length.

- This login key applies to all the sectors covered by the range formulated by "nStart" and "nWriteLen".

LoginKey Example	Key String
0	FFFFFFFFFFFF
"F1F2F3F4F5F6"	F1F2F3F4F5F6

nLoginKeyType

A value that identifies the login key.

0 *	Key A
1	Key B

Example

```
<HTML>
<BODY onload=”test()”>
<SCRIPT>
function test()
{
    var sData               =”AABBCCDD”;
    var nStartBlock         = -1;
    var nWriteLen    = 8;
    var nTimeout     = 3;
    var nMode               = 0;
    var sLoginKey    ='FFFFFFFFFFFF';
    var nLoginKeyType       =0;

    var nWriteRfidData =
window.external.WriteRfidData(sData,nStartBlock,nWriteLen,nTimeout,
nMode,sLoginKey,nLoginKeyType);
    alert("WriteRfidData="+nWriteRfidData);
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value

If successful, it returns 0.

Otherwise, it returns a negative value to indicate an error condition:

-101	E_READER_NOT_INIT
-273	E_TAG_VALUE_OVER_RANGE
-274	E_TAG_INVALID_LENGTH
-275	E_TAG_GET_DATA_FAILED
-276	E_TAG_WRITE_FAILED
-277	E_TAG_CANNOT_WRITE
-278	E_TAG_NO_TAG
-279	E_TAG_WRITE_TIMEOUT

WorkingType

Purpose

Sets the reader antenna (RF) working model between ISO-14443A, ISO-14443B and ISO-15693. Before executing this function, you must execute OpenCard first.

- Only one working mode can be set at a time. The default working mode is ISO-14443A.

Syntax

window.external.WorkingType(*nType*)

Parameters

nType

Sets up the access mode of the reader.

Name	Value
ISO14443A	0x41
ISO14443B	0x42
ISO15693	0x31
Default setting to access SR176 and SRIX4K cards within ISO-14443B.	0x73

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
    window.external.WorkingType(0x41);
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value

If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

AntennaControl

Purpose	Before executing this function, you must execute Open Reader. It allows the system to turn off the antenna when it goes into standby, which reduces equipment power and fulfills power saving purposes.
Syntax	window.external.AntennaControl(*nSelect*)
Parameters	*nSelect* For ANTENNA_SELECT_POWER_LOW: if the function is executed successfully without any error codes, the antenna will switch to low power consumption mode after 8 seconds.

Name	Value
Antenna off	0x00
Antenna on	0x01
Auto transfer	0x08
Low-power	0x10

Example

```
<HTML>
<BODY onload=”test()”>
<SCRIPT>
function test() {
    window.external.AntennaControl(0x01);
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value: If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

OpenCard

Purpose: Select ISO-14443A card and get card spec type information (supports multi-card). Before operating an ISO-14443A card, execute this function first.

Syntax: window.external.OpenCard()

Parameters: None

Example:

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test()
{
    var nAntennaControl=window.external.OpenCard();
    var sCardUid=window.external.GetCardUid();
    var sCardAtqa=window.external.GetCardAtqa();
    var sCardSak=window.external.GetSak();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Other Info:

Name	Type	Description
GetCardUid()	String	Return the card's serial numbers. Length is LEN (11) array, UID_END = 77 is the card's serial end value.
GetCardAtqa()	String	Return ATQA card type, length is ATQA_LEN (2) (Unit: BYTE).
GetCardSak()	String	Return SAK card type.

Return Value: If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

CloseCard

Purpose: Close ISO-14443A card. To finish writing data to an ISO-14443A card, execute this function and then continue to read another card.

Syntax: window.external.CloseCard()

Parameters: None

Example:

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test() {
    window.external.CloseCard();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value: If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

ReadMifareOneBlock

Purpose

Read ISO-14443A card data according to the specified block location.

Syntax

window.external.ReadMifareOneBlock(*nKeyType*,*nBlock*,*sKey*)

Parameters

nKeyType

[in] Application file key type.

Name	Constant	Value
Key A	CARD_KEY_A	0x00
Key B	CARD_KEY_B	0x01

nBlock

Block number to read. Block number starts from 0; S50: Block0~Block63 (does not include Security Block, total 45 blocks), S70: Block0~Block255 (does not include Security Block, total 210 blocks).

sKey

Card default key; length is CARD_KEY_LEN(6) array.

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test()
{
    var nKeyType=0x00;//KeyA;
    var nBlock=4;
    var sKey='FFFFFFFFFFFF';
    var sReadMifareOneBlock=window.external.ReadMifareOneBlock(
    nKeyType,nBlock,sKey);
    var sCardData = window.external.GetCardData();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Other Info

Name	Type	Description
GetCardData()	String	Return read data; length is BLOCK_LEN (16) array.

Return Value

If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

WriteMifareOneBlock

Purpose

Write data into ISO-14443A card according to the specified block location.

Syntax

window.external.WriteMifareOneBlock(*nKeyType*, *nBlock*, *sKey, sData*)

Parameters

nKeyType

[in] Application file key type.

Name	Constant	Value
Key A	CARD_KEY_A	0x00
Key B	CARD_KEY_B	0x01

nBlock

Block number to read. Block number starts from 0; S50: Block0~Block63 (does not include Security Block, total 45 blocks), S70: Block0~Block255 (does not include Security Block, total 210 blocks).

sKey

Card default key; length is CARD_KEY_LEN(6) array.

sData

Data to write; length is BLOCK_LEN(16) array.

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test()
{
    var nKeyType=0x00;//KeyA;KeyB;
    var nBlock=4;
    var sKey='FFFFFFFFFFFF'';
    var sData='1234567890';
    var nWriteMifareOneBlock=window.external.WriteMifareOneBlock
    (nKeyType,nBlock,sKey,sData);

}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value

If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

ReadUltraLight

Purpose	Read data from UltraLight card according to the specified block location; this function is for Mifare UltraLight card only.
Syntax	window.external.ReadUltraLight(*nBlock*)
Parameters	*nBlock* Block number to read data.

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test()
	{
	var nBlock=0;
	var nReadUltraLight=window.external.ReadUltraLight(nBlock);
	var sCardData=window.external.GetCardData();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Other Info	
Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

Name	Type	Description
GetCardData()	String	Return read data; length is ULTRA_LIGHT_READ_LEN (16) array.

WriteUltraLight

Purpose	Write data into UltraLight card according to the specified block location; this function is for Mifare UltraLight card.
Syntax	window.external.WriteUltraLight(*nBlock*)
Parameters	*nBlock* Block number to write data.

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test()
	{
	var nBlock =0;
	var sData ="1234";
	var nWriteUltraLight = window.external.WriteUltraLight(nBlock, sData)
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Other Info	

Name	Type	Description
nBlock	BYTE	Block number to write data.
sData	String	Data to write; length is ULTRA_LIGHT_WRITE_LEN (4) array.

Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

STCardSelect

Purpose	Select ISO-14443B card and get card number. Before operating an ISO-14443B card, please execute this function first.
Syntax	window.external.STCardSelect()
Parameters	None

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test()
	{
	var nSTCardSelect=window.external.STCardSelect();
	var sCardSelectNum=window.external.GetCardSelectNum();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Other Info	

Name	Type	Description
GetCardSelectNu()	String	Return card number.

Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

SR176ReadBlock

Purpose	Read SR176 card data according to the specified block location.
Syntax	window.external.SR176ReadBlock(*nBlkNo*)
Parameters	*nBlkNo* Read block number; Block 0~15.

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test()
	{
	var nBlkNo=0;
	var nSR176ReadBlock = window.external.SR176ReadBlock(nBlkNo);
	var sCardData = window.external.GetCardData();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Other Info	
Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

Name	Type	Description
GetCardData()	String	Return read data, length is SR176_READ_LEN (16) array.

SR176WriteBlock	
Purpose	Write data into SR176 card according to the specified block location.
Syntax	window.external.SR176WriteBlock(*nBlkNo, sData*)
Parameters	*nBlkNo* Data block number to write; Block 4~14. *sData* Data to write, length is SR176_BLOCK_LEN(2) array.

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test()
	{
	var nBlkNo=0;
	var sData="12";
	var nSR176WriteBlock = window.external.SR176WriteBlock(nBlkNo, sData);
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

SRIX4KReadBlock

Purpose: Read SRIX4K card data according to the specified block location.

Syntax: window.external.SRIX4KReadBlock(*nBlkNo*)

Parameters:

nBlkNo

Read block number; Block 0~127.

Example:

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test()
{
    var nBlkNo=0;
    var nSRIX4KReadBlock = window.external.SRIX4KReadBlock(nBlkNo);
    var sCardData = window.external.GetCardData();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Other Info:

Name	Type	Description
GetCardData()	String	Return read data, length is SRIX4K_BLOCK_LEN (4) array.

Return Value: If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

SRIX4KWriteBlock

Purpose: Write data into SRIX4K card according to the specified block location.

Syntax: window.external.SRIX4KWriteBlock(*nBlkNo, sData*)

Parameters:

nBlkNo

Data block number to write; Block 7~127.

sData

Data to write, length is SRIX4K_BLOCK_LEN(4) array.

Example:

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test()
{
    var nBlkNo=0;
    var sData="1234";
    var nSRIX4KWriteBlock = window.external.SRIX4KWriteBlock(nBlkNo,
    sData);
}
…
</SCRIPT>
</BODY>
</HTML>
```

Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

SRIX4KReadUID

Purpose	Read SRIX4K unique identifier.
Syntax	window.external.SRIX4KReadBlock()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test()
	{
	var nSRIX4KReadUID = window.external.SRIX4KReadUID();
	var sCardUid=window.external.GetCardUid();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Other Info	

Name	Type	Description
GetCardUid()	String	Return unique identifier code; length is SRIX4K_UID_LEN(10) array.

Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

ISO15693Inventory

Purpose	Select ISO-15693 card and set card into Ready mode, get card unique identifier (UID).
Syntax	window.external.ISO15693Inventory()
Parameters	None

Example	<HTML>
	<BODY onload="test()">
	<SCRIPT>
	function test()
	{
	var nInventory = window.external.ISO15693Inventory();
	var sCardMask = window.external.GetCardMask();
	var sCardDsfid = window.external.GetCardDsfid();
	var sCardUID = window.external.GetCardUid();
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Other Info	

Name	Type	Description
GetCardMask()	String	Mask; length is ISO15693_MASK_LEN(8) array.
GetCardDsfid()	String	Data storage format identifier, marks which type of format this card uses to store information. Please refer to ISO15693 related documents.
GetCardUid()	String	Return UID; length is ISO15693_UID_LEN(8) array.

Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

ISO15693Read

Purpose	Read ISO-15693 card data according to the specified block location.
Syntax	window.external.ISO15693Read(*sUID*, *nBlockStart*, *nBlockCount*)
Parameters	*sUID* Return UID; length is ISO15693_UID_LEN(8) array. *nBlockStart* Start to read file block number; total block number varies for different card part numbers. *nBlockCount* Total block numbers; total block number varies for different card part numbers; Block1~15, MAX is 15 blocks.

Example

```
<HTML>
<BODY onload="test()">
<SCRIPT>
function test()
{
    var sUID=window.external.GetCardUid();
    var nBlockStart=0;
    var nBlockCount=1;
    var nISO15693Read = window.external.ISO15693Read(sUID,
    nBlockStart,nBlockCount);
    var sCardData = window.external.GetCardData();
}
…
</SCRIPT>
</BODY>
</HTML>
```

Other Info

Name	Type	Description
GetCardData()	String	Return read data, length is SRIX4K_BLOCK_LEN (4) array.

Return Value: If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

ISO15693Write	
Purpose	Write data into ISO-15693 card according to the specified block location.
Syntax	window.external.ISO15693Write(*sUID, nBlkNo, sData*)
Parameters	*sUID* Return UID; length is ISO15693_UID_LEN(8) array. *nBlkNo* File block number which data will be written into; block number varies for cards of different part numbers. *sData* Data to write; length is ISO15693_BLOCK_LEN(4) array.

Example	<HTML>
	<BODY onload=”test()”>
	<SCRIPT>
	function test()
	{
	var sUID=window.external.GetCardUid();
	var nBlkNo=0;
	var sData=”1234”;
	var nISO15693Write = window.external.ISO15693Write(sUID,nBlkNo,
	sData);
	}
	…
	</SCRIPT>
	</BODY>
	</HTML>

Return Value	If executed successfully, it will respond LRSUCCESS. For other codes, please refer to Appendix VIII: Response Code Instructions.

Appendix I

SCAN ENGINE SETTINGS

MIRROR Browser supports the following reader types. Reader availability depends on the hardware integrated on the mobile computer.

Scan Engine		ID	CP30	CP50	CP60	9200
1D	*CCD*	*SM1*				✓
	Laser	*SE955*	✓	✓	✓	✓
2D	*2D Imager*	*PL4507*	✓	✓		
		SE4500 Software Decode SDK			✓	✓
RFID	*ID_MOD_MP_RFID*			✓	✓	✓

Reader combination allowed is either 1D plus RFID or 2D plus RFID. Each combination is able to initialize both readers at the same time (dual mode operation). For example, when you hit a physical scan key with MIRROR Browser active on the mobile computer, it reads a printed barcode in position (or an RFID tag in proximity, depending on which comes first).

Note: 1D and 2D scan engines don't coexist on the mobile computer because they are both barcode readers and only one barcode reader is allowed on board the mobile computer.

IN THIS CHAPTER

Symbologies Supported.....106
RFID Tags Supported.....108

SYMBOLOGIES SUPPORTED

Depending on the scan engine installed, the supported symbologies are different as listed below. For details on configuring associated settings, refer to Appendix II and III.

		9200 CCD	9200 Laser CP30 Laser CP50 Laser CP60 Laser	9200 2D CP30 2D CP50 2D CP60 2D
Codabar		✓	✓	✓
Code 11		✗	✓	✓
Code 39	Code 39	✓	✓	✓
	Trioptic Code 39	✗	✓	✓
	Italian Pharmacode (Code 32)	✓	✓	✓
Code 93		✓	✓	✓
Code 128	Code 128	✓	✓	✓
	GS1-128 (EAN-128)	✓	✓	✓
	ISBT 128	✓	✓	✓
Code 2 of 5	Chinese 25	✗	✓	✓
	Industrial 25 (Discrete 25)	✓	✓	✓
	Interleaved 25	✓	✓	✓
	Convert Interleaved 25 to EAN-13	✗	✓	✓
	Matrix 25	✗	✗	✓
Composite Code	Composite CC-A/B	✗	✗	✓
	Composite CC-C	✗	✗	✓
	Compostie TLC 39	✗	✗	✓
GS1 DataBar (RSS)	GS1 DataBar-14 (RSS-14)	✓	✓	✓
	GS1 DataBar Limited (RSS Limited)	✓	✓	✓
	GS1 DataBar Expanded (RSS Expanded)	✓	✓	✓
	Convert to UPC/EAN	✗	✓	✓
MSI		✓	✓	✓
Postal Codes	Australian Postal	✗	✗	✓
	Japan Postal	✗	✗	✓
	Netherlands KIX Code	✗	✗	✓
	US Postnet	✗	✗	✓

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

RFID TAGS SUPPORTED

Supported labels include ISO 15693, ISO 14443A, ISO 14443B and ISO 18092.

RFID TAGS SUPPORTED ON CP50

		UID Only	Read Page	Write Page
ISO 14443A	Mifare Standard 1K	✓	✓	✓
	Mifare Standard 4K	✓	✓	✓
	Mifare Ultralight	✓	✓	✓
	Mifare DESFire	✓	---	---
	Mifare S50	✓	✓	✓
	SLE44R35	✓	✓	✓
	SLE66R35	✓	✓	✓
ISO 14443B	SR176	✓	✓	✓
ISO 15693	ICODE SLI	✓	✓	✓
	SRF55V02P	✓	✓	✓
	SRF55V02S	✓	✓	✓
	SRF55V10P	✓	✓	✓
	TI Tag-it HF-I	✓	✓	✓
	ST LRI512	✓	✓	✓

RFID TAGS SUPPORTED ON CP60

		UID Only	Read Page	Write Page
ISO 14443A	Mifare Standard 1K S50	✓	✓	✓
	Mifare Standard 4K S70	✓	✓	✓
	Jcop 41 (Mifare 1K & 4K compatible)	✓	✓	✓
	Mifare Ultralight (UL/ULC)	✓	✓	✓
	Mifare ProX	✓	✓	✓
	Mifare DESFire	✓	---	---
	Mifare Plus	✓	---	---
	Mifare S20 Mini	✓	✓	✓
	SLE 66CLX320P	✓	---	---
	SLE 55R04/08	✓	---	---
	Smart MX	✓	---	---
	Jewel	✓	✓	✓
	Topaz	✓	✓	✓
	Dual interface – ISO 14443A compliant	✓	---	---
ISO 14443B	SLE 6666CL160S	✓	---	---
	SR176	✓	✓	✓
	SRIX 4K	✓	✓	✓
	SLIX 4K	✓	✓	✓
	Dual interface – ISO 14443B compliant	✓	---	---
ISO 15693	EM 4135	✓	✓	✓
	ICode SLI	✓	✓	✓
	LRI12	✓	✓	✓
	LRI64	✓	✓	✓
	LRI128	✓	✓	✓
	LRI2K	✓	✓	✓
	SRF55VxxP	✓	✓	✓
	SRF55VxxS	✓	---	---
	Tag-it HF-I Std	✓	✓	✓
	TempSense	✓	---	---
	ICode1 with EAS & AFI	✓	✓	✓

	ICode	✓	✓	✓
FeliCa	Sony FeliCa	✓	---	---

RFID TAGS SUPPORTED ON 9200

		UID Only	Read Page	Write Page
ISO 14443A	Mifare Classic S50 1K	✓	✓	✓
	Mifare S50 1K UID 7 Bytes	✓	✓	✓
	Mifare Classic S70 4K	✓	✓	✓
	Mifare Ultralight (UL/ULC)	✓	✓	✓
	Mifare S20 Mini	✓	✓	✓
	Mifare DESFire V0.6 4K	✓	✓ Note	✓ Note
	Mifare DESFire EV1 2K/4K/8K	✓	✓ Note	✓ Note
	Mifare PLUS S 2K/4K	✓	✓ Note	✓ Note
	SLE 66R35	✓	✓	✓
ISO 14443B	SR176	✓	✓	✓
	SRI512	✓	✓	✓
	SRIX4K	✓	✓	✓
ISO 15693	I-CODE SLI-S/L	✓	✓	✓
	I-CODE SLIX	✓	✓	✓
	SRF 55V10P	✓	✓	✓
	SRF 55V02P	✓	✓	✓
	Tag-it HF-I Pro/Plus	✓	✓	✓

Note: To read/write data or support additional functions, users must obtain permission from the RFID chip manufacturer.

Appendix II

CCD (SM1)

IN THIS CHAPTER

Symbology Settings for 9200 111

Symbology settings for CCD reader (SM1) are:

SYMBOLOGY SETTINGS FOR 9200

Laser Engine	Description	Default
Codabar		**Enable**
CLSI editing	When applied, the CLSI editing strips the start/stop characters and inserts a space after the first, fifth, and tenth characters of a 14-character Codabar barcode. ▸ The 14-character barcode length does not include start/stop characters.	No
NOTIS editing	Sets whether to include start/stop characters in the transmitted data. ▸ NOTIS Editing is to strip the start/stop characters, i.e. to disable "Transmit Start/Stop Characters".	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Code 128		**Enable**
Code 128	Read standard Code 128 barcodes (= without leading FNC1 character).	Enable
GS1-128 (UCC/EAN-128)	Read GS1-128 barcodes with leading FNC1 character.	Enable
ISBT 128	Read ISBT 128 barcodes.	Enable
Code 39		**Enable**
Convert Code 39 to Code 32	Convert to Italian Pharmacode.	No
Code 32 Prefix	Prefix character “A” to Code 32 barcodes. ▸ “Convert to Code 32” must be enabled for this to function properly.	No
Check digit verification	Decide whether to verify the check digit. If the check digit is incorrect, the barcode will not be accepted.	No
Transmit check digit	Decide whether to include the check digit in the data being transmitted. ▸ “Check digit verification” must be enabled.	No

Full ASCII conversion	Code 39 Full ASCII includes all alphanumeric and special characters (full ASCII character set).	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Code 93		**Enable**
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Industrial 25 (Discrete 25)		**Enable**
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
EAN/JAN		**---**
Enable EAN-8		Enable
Enable EAN-13		Enable
Bookland EAN (ISBN)	EAN-13 barcodes starting with 978 will be converted to ISBN.	Enable
Enable Addons	Decide whether to decode EAN-8, EAN-13, UPC-E0, UPC-E1, UPC-A with addons (including Addon 2 and 5). ▸ Ignore Addons ▸ Decode Only With Addons ▸ Auto-discriminate	Ignore Addons
Addons redundancy	When “Auto-discriminate” is applied, decide the number of times (2~30) of supplementary decoding of the same barcode that makes a valid reading.	10 times
EAN-8 extend	Convert EAN-8 to EAN-13 format.	No
UPC		**---**
UPC-A/UPC-E/UPC-E1	For the default value, the UPC-A and UPC-E are enabled.	Enable
Enable Addons	Decide whether to decode EAN-8, EAN-13, UPC-E0, UPC-E1, UPC-A with addons (including Addon 2 and 5). ▸ Ignore Addons ▸ Decode Only With Addons ▸ Auto-discriminate	Ignore Addons
Addons redundancy	When “Auto-discriminate” is applied, decide the number of times (2~30) of supplementary decoding the same barcode that makes a valid reading.	10 times
Transmit check digit	Decide whether to include the UPC-A/UPC-E/UPC-E1 check digit in the data being transmitted.	Yes
Convert UPC-E0/ UPC-E1 to UPC-A	The UPC-E0/UPC-E1 barcode will be expanded into UPC-A, and the next processing will follow the settings configured for UPC-A.	No
Transmit preamble	Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted.	System Character
MSI		**Enable**

Check digit verification	One check digit is mandatory for decoding MSI barcodes. Select whether a second check digit should be verified. If the check digit is incorrect, the barcode will not be accepted. If Two Check Digits option is selected, select one of the algorithms below. ▸ Mod 10/Mod 11 ▸ Mod 10/Mod 10	Mod 10/Mod 10
Transmit check digit	Decide whether to include the check digit in the data being transmitted.	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Interleaved 25		**Enable**
Check digit verification	Decide whether to verify the check digit. If desired, select one of the algorithms below. If the check digit is incorrect, the barcode will not be accepted. ▸ Disable ▸ USS check digit ▸ OPCC check digit	Disable
Transmit check digit	Decide whether to include the check digit in the data being transmitted.	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
GS1 DataBar (RSS)		**---**
GS1 DataBar-14	GS1 DataBar-14 is short for GS1 DataBar Omnidirectional. This group consists of (1) GS1 DataBar Omnidirectional, (2) GS1 DataBar Truncated, (3) GS1 DataBar Stacked, and (4) GS1 DataBar Stacked Omnidirectional.	Enable
GS1 DataBar Limited		Enable
GS1 DataBar Expanded	This group consists of (1) GS1 DataBar Expanded, and (2) GS1 DataBar Expanded Stacked.	Enable

Appendix III

LASER (SE955)

IN THIS CHAPTER

Symbology Settings for CP30/CP50/9200 115
Symbology Settings for CP60 120

Symbology settings for 1D laser reader (SE955) are:

SYMBOLOGY SETTINGS FOR CP30/CP50/9200

Laser Engine	Description	Default
Codabar		**Enable**
CLSI editing	When applied, the CLSI editing strips the start/stop characters and inserts a space after the first, fifth, and tenth characters of a 14-character Codabar barcode. ▸ The 14-character barcode length does not include start/stop characters.	No
NOTIS editing	Sets whether to include start/stop characters in the transmitted data. ▸ NOTIS Editing is to strip the start/stop characters, i.e. to disable "Transmit Start/Stop Characters".	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Code 128		**Enable**
GS1-128 (UCC/EAN-128)	Read GS1-128 barcodes with leading FNC1 character.	Enable
ISBT 128	Read ISBT 128 barcodes.	Enable
Code 39		**Enable**
Trioptic	Decide whether to decode Trioptic Code 39. ▸ Trioptic Code 39 is a variant of Code 39 used in the marking of computer tape cartridges. It always contains six characters.	No
Convert Code 39 to Code 32	Convert to Italian Pharmacode.	No
Code 32 Prefix	Prefix character “A” to Code 32 barcodes. ▸ “Convert to Code 32” must be enabled for this to function properly.	No
Check digit verification	Decide whether to verify the check digit. If the check digit is incorrect, the barcode will not be accepted.	No

<table>
<tr><td>Transmit check digit</td><td>Decide whether to include the check digit in the data being transmitted.
“Check digit verification” must be enabled.</td><td>No</td></tr>
<tr><td>Full ASCII conversion</td><td>Code 39 Full ASCII includes all alphanumeric and special characters (full ASCII character set).</td><td>No</td></tr>
<tr><td>Select Length (min/max)</td><td>One or two fixed lengths
Range (1~55)</td><td>4~55</td></tr>
<tr><td colspan="2">Code 93</td><td>Enable</td></tr>
<tr><td>Select Length (min/max)</td><td>One or two fixed lengths
Range (1~55)</td><td>4~55</td></tr>
<tr><td colspan="2">Industrial 25 (Discrete 25)</td><td>Enable</td></tr>
<tr><td>Select Length (min/max)</td><td>One or two fixed lengths
Range (1~55)</td><td>4~55</td></tr>
<tr><td colspan="2">EAN/JAN</td><td>---</td></tr>
<tr><td>Enable EAN-8</td><td></td><td>Enable</td></tr>
<tr><td>Enable EAN-13</td><td></td><td>Enable</td></tr>
<tr><td>Bookland EAN (ISBN)</td><td>The EAN-13 barcode starting with 978 will be converted to ISBN.</td><td>Enable</td></tr>
<tr><td>Enable Addons</td><td>Decide whether to decode EAN-8, EAN-13, UPC-E0, UPC-E1, UPC-A with addons (including Addon 2 and 5).
Ignore Addons
Decode Only With Addons
Auto-discriminate</td><td>Ignore Addons</td></tr>
<tr><td>Addons redundancy</td><td>When “Auto-discriminate” is applied, decide the number of times (2~30) of supplementary decoding the same barcode that makes a valid reading.</td><td>10 times</td></tr>
<tr><td>EAN-8 extend</td><td>Convert EAN-8 to EAN-13 format.</td><td>No</td></tr>
<tr><td>UCC coupon extended code</td><td>Read UPC-A barcodes starting with digit "5", EAN-13 barcodes starting with digits "99", and UPC-A/GS1-128 Coupon Codes.
UPC-A, EAN-13, and GS1-128 must be enabled first!
Use “Addon Redundancy” to control auto-discrimination of the GS1-128 (right half) of a coupon code.</td><td>No</td></tr>
<tr><td>EAN security level</td><td>Sets the security level to ensure decoding accuracy considering the printed quality of the barcodes such as Code 128, Code 93, and UPC/EAN. The higher the level is, the more security is ensured. Options are:
<table>
<tr><th>Level</th><th>Description</th></tr>
<tr><td>0</td><td>With this default, the scan engine is aggressive enough to decode most “in-spec” barcodes.</td></tr>
<tr><td>1</td><td>Select this level if misdecodes have occurred. It fixes most misdecodes.</td></tr>
<tr><td>2</td><td>Select this level if Level 1 should fail to eliminate misdecodes.</td></tr>
</table></td><td>2</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>3</td><td>Select this level if Security Level 2 should fail to prevent misdecodes. However, as this level actually impairs the decoding ability of scan engine, it would be better to improve the barcode's print quality if this level should be needed.</td></tr>
</table></td><td></td></tr>
<tr><td colspan="2">UPC</td><td>---</td></tr>
<tr><td>UPC-A/UPC-E/UPC-E1</td><td>For the default value, the UPC-A and UPC-E are enabled.</td><td>Enable</td></tr>
<tr><td>Enable Addons</td><td>Decide whether to decode EAN-8, EAN-13, UPC-E0, UPC-E1, UPC-A with addons (including Addon 2 and 5).
▸ Ignore Addons
▸ Decode Only With Addons
▸ Auto-discriminate</td><td>Ignore Addons</td></tr>
<tr><td>Addons redundancy</td><td>When “Auto-discriminate” is applied, decide the number of times (2~30) of supplementary decoding the same barcode that makes a valid reading.</td><td>10 times</td></tr>
<tr><td>Transmit check digit</td><td>Decide whether to include the UPC-A/UPC-E/UPC-E1 check digit in the data being transmitted.</td><td>Yes</td></tr>
<tr><td>Convert UPC-E0/ UPC-E1 to UPC-A</td><td>The UPC-E0/UPC-E1 barcode will be expanded into UPC-A, and the next processing will follow the settings configured for UPC-A.</td><td>No</td></tr>
<tr><td>UCC coupon extended code</td><td>Read UPC-A barcodes starting with digit "5", EAN-13 barcodes starting with digits "99", and UPC-A/GS1-128 Coupon Codes.
▸ UPC-A, EAN-13, and GS1-128 must be enabled first!
▸ Use “Addon Redundancy” to control auto-discrimination of the GS1-128 (right half) of a coupon code.</td><td>No</td></tr>
<tr><td>Transmit preamble</td><td>Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted.</td><td>System Character</td></tr>
<tr><td>UPC security level</td><td>Sets the security level to ensure decoding accuracy considering the printed quality of the barcodes such as Code 128, Code 93, and UPC/EAN. The higher the level is, the more security is ensured. Options are:<table>
<tr><th>Level</th><th>Description</th></tr>
<tr><td>0</td><td>With this default, the scan engine is aggressive enough to decode most “in-spec” barcodes.</td></tr>
<tr><td>1</td><td>Select this level if misdecodes have occurred. It fixes most misdecodes.</td></tr>
<tr><td>2</td><td>Select this level if Level 1 should fail to eliminate misdecodes.</td></tr>
<tr><td>3</td><td>Select this level if Security Level 2 should fail to prevent misdecodes. However, as this level actually impairs the decoding ability of scan engine, it’d be better to improve the barcode’s print quality if this level should be needed.</td></tr>
</table></td><td>2</td></tr>
</table>

MSI		**Enable**
Check digit verification	One check digit is mandatory for decoding MSI barcodes. Select whether a second check digit should be verified. If the check digit is incorrect, the barcode will not be accepted. If Two Check Digits option is selected, select one of the algorithms below. ▸ Mod 10/Mod 11 ▸ Mod 10/Mod 10	Mod 10/Mod 10
Transmit check digit	Decide whether to include the check digit in the data being transmitted.	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Interleaved 25		**Enable**
Check digit verification	Decide whether to verify the check digit. If desired, select one of the algorithms below. If the check digit is incorrect, the barcode will not be accepted. ▸ Disable ▸ USS check digit ▸ OPCC check digit	Disable
Transmit check digit	Decide whether to include the check digit in the data being transmitted.	No
Convert Interleaved 25 to EAN-13	Convert a 14-character barcode into EAN-13 if the following requirements are met: ▸ The barcode must have a leading 0 and a valid EAN-13 check digit.	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
GS1 DataBar (RSS)		**---**
GS1 DataBar-14	GS1 DataBar-14 is short for GS1 DataBar Omnidirectional. This group consists of (1) GS1 DataBar Omnidirectional, (2) GS1 DataBar Truncated, (3) GS1 DataBar Stacked, and (4) GS1 DataBar Stacked Omnidirectional.	Enable
GS1 DataBar Limited		Enable
GS1 DataBar Expanded	This group consists of (1) GS1 DataBar Expanded, and (2) GS1 DataBar Expanded Stacked.	Enable
Convert GS1 DataBar To UPC/EAN	Convert to UPC/EAN" only applies to GS1 Databar-14 and GS1 Databar Limited barcodes not decoded as part of a Composite barcode. **Convert to EAN-13** Strip the leading "010" from barcodes. ▸ "01" is the Application ID and must be followed by a single zero (the first digit encoded)	No

Convert GS1 DataBar To UPC/EAN	**Convert to UPC-A** Strip the leading "0100" from barcodes. ▸ "01" is the Application ID and must be followed by two or more zeros (but not six zeros)	No
Code 11		**Enable**
Check digit verification	Decide whether to verify check digits. If the check digit is incorrect, the barcode will not be accepted. ▸ Disable ▸ One Check Digit ▸ Two Check Digits	Disable
Transmit check digit	Decide whether to include check digits in the transmitted data. ▸ "Check digit verification" must be enabled.	No
Select Length	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Chinese 25		**Enable**

MISCELLANEOUS

Laser Engine	Description	Default
Miscellaneous Options		**---**
Transmit Code ID	Decide whether to include AIM Code ID in the beginning of data. none AIM code ID character Each AIM Code ID contains the three-character string "**]cm**" –] = Flag Character (ASCII 93) c = Code Character (see below) m = Modifier Character (see below)	none

SYMBOLOGY SETTINGS FOR CP60/9200

Symbology	Description	Default
CODABAR		
CodaBar		
Codabar	Checkbox to enable Codabar decoding.	Enable
Length option	Sets the length of the Codabar symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
CLSI	When applied, the CLSI editing strips the start/stop characters and inserts a space after the first, fifth, and tenth characters of a 14-character Codabar barcode. The 14-character barcode length does not include start/stop characters.	Disable
NOTIS	Sets whether to include start/stop characters in the transmitted data. NOTIS Editing is to strip the start/stop characters, i.e. to disable "Transmit Start/Stop Characters".	Disable
CODE 11		
Code 11		
Code 11	Checkbox to enable Code 11 decoding.	Enable
Check Digit Option	Sets whether to verify check digits according to the selected option. If the check digits are incorrect, the barcode will not be accepted. Disable One Check Digit Two Check Digits	Disable
Transmit Check Digit	Selects whether to include check digits in the transmitted data. Check Digit Option" must be enabled.	Disable
Length option	Sets the length of the Code 11 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
CODE 39		
Code 39		
Code 39	Checkbox to enable Code 39 decoding.	Enable
Trioptic Code 39	Selects whether to decode Trioptic Code 39. Trioptic Code 39 is a variant of Code 39 used in the marking of computer tape cartridges. It always contains six characters.	Disable

Convert to Code 32	Selects whether to convert decoded data to Italian Pharmacode.	Disable
Code 32 Prefix	Prefix character “A” to Code 32 barcodes. “Convert to Code 32” must be enabled for this to function properly.	Disable
Verify Check Digit	Selects whether to verify the Modulo 43 check digit. If the check digit is incorrect, the barcode will not be accepted.	Disable
Transmit Check Digit	Decide whether to include the check digit in the data to transmit. “Verify Check Digit” must be enabled.	Disable
Support Full ASCII	Selects whether to enable Code 39 Full ASCII decoding. Characters are paired to encode the full ASCII character set.	Disable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
CODE 93		
Code 93		
Enable Code 93	Checkbox to enable Code 93 decoding.	Enable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
CODE 128		
Code 128		
Enable Code 128	Checkbox to enable Code 128 decoding.	Enable
ISBT 128		
Enable ISBT 128	Checkbox to enable ISBT 128 decoding.	Enable
GS1-128		
Enable GS1-128	Checkbox to enable GS1-128 decoding.	Enable
CODE 2 OF 5		
Chinese 25		
Enable Chinese 25	Checkbox to enable Chinese 25 decoding.	Enable
Discrete 25		
Discrete 25	Checkbox to enable Discrete 25 decoding.	Enable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)

Interleaved 25		
Interleaved 25	Checkbox to enable Interleaved 2 of 5 decoding.	Enable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
Verify Check Digit	Disable USS Check Digit OPCC Check Digit	Disable
Transmit Check Digit	Decide whether to include the check digit in the data being transmitted.	Disable
Convert To EAN-13	Convert a 14-character Interleaved 25 barcode into EAN-13 if the following requirements are met: The barcode must have a leading 0 and a valid EAN-13 check digit.	Disable
GS1 DATABAR		
GS1 DataBar-14		
Enable GS1 DataBar-14	Checkbox to enable GS1 DataBar-14 decoding.	Enable
GS1 DataBar Limited		
Enable GS1 DataBar Limited	Checkbox to enable GS1 DataBar Limited decoding.	Enable
GS1 DataBar Expanded		
Enable GS1 DataBar Expanded	Checkbox to enable GS1 DataBar Expanded decoding.	Enable
Convert to UPC/EAN		
Convert to UPC/EAN	This only applies to GS1 DataBar-14 and GS1 DataBar Limited barcodes not decoded as part of a Composite barcode. **Convert to EAN-13** Strips the leading “010” from barcodes. The barcode must be composed of a leading “01” as the application identifier (AI) and a first digit of zero. **Convert to UPC-A** Strips the leading “0100” from barcodes. The barcode must be composed of a leading “01” as the application identifier (AI) and two or more zeros (but not six zeros).	Disable

MSI		
MSI		
MSI	Checkbox to enable MSI decoding.	Enable
Length option	Sets the length of the MSI symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
Check Digit Option	One check digit is mandatory for decoding MSI barcodes. Select whether a second check digit should be verified. If the check digits are incorrect, the barcode will not be accepted. One Check Digit Two Check Digits	One Check Digit
Transmit Check Digit	Decide whether to include the check digit in the data being transmitted.	Disable
Algorithm	When two check digits are set for verification, two choices are available for the pair of check digits. Modulo10 / Modulo11 Double Modulo 10	Double Modulo 10
UPC/EAN		
EAN-8		
EAN-8	Checkbox to enable EAN-8 decoding.	Enable
EAN-8 Extend	Checkbox to enable converting EAN-8 to EAN-13 format.	Disable
EAN-13		
EAN-13	Checkbox to enable EAN-13 decoding.	Enable
Bookland EAN	Checkbox to enable ISBN decoding. If enabled, select Bookland ISBN Format in the drop-down box below.	Enable
Bookland ISBN Format	Decodes Bookland data starting with 978 in 10-digit format along with the Bookland check digit, or Bookland data starting with 978/979 as EAN-13 format.	Bookland ISBN-10
UPC-A		
UPC-A	Checkbox to enable UPC-A decoding.	Enable
Preamble	Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted. No transmit: transmits none of the above Transmit System Character: transmits system number only Transmit Sys. Character and Country Code: transmits system number and country code	Transmit System Character
Transmit Check Digit	Decide whether to include the UPC-A check digit (the last character in the barcode) in the data being transmitted.	Enable

UPC-E		
UPC-E	Checkbox to enable UPC-E decoding.	Enable
Preamble	Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted. No transmit: transmits none of the above Transmit System Character: transmits system number only Transmit Sys. Character and Country Code: transmits system number and country code	Transmit System Character
Convert to UPC-A	The UPC-E barcode will be expanded into UPC-A format, and the next process will follow the settings configured for UPC-A.	Disable
Transmit Check Digit	Decide whether to include the UPC-E1 check digit (the last character in the barcode) in the data being transmitted.	Enable
UPC-E1		
UPC-E1	Checkbox to enable UPC-E1 decoding.	Disable
Preamble	Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted. No transmit: transmits none of the above Transmit System Character: transmits system number only Transmit Sys. Character and Country Code: transmits system number and country code	Transmit System Character
Convert to UPC-A	The UPC-E1 barcode will be expanded into UPC-A format, and the next process will follow the settings configured for UPC-A.	Disable
Transmit Check Digit	Decide whether to include the UPC-E1 check digit (the last character in the barcode) in the data being transmitted.	Enable
General Preference		
Support Coupon Code	Reads UPC-A barcodes starting with “5”, EAN-13 barcodes starting with “99”, and UPC-A/EAN-128 Coupon Codes. UPC-A, EAN-13, and GS1-128 must be enabled first! Use “Addon Redundancy” to control auto-discrimination of the GS1-128 (right half) of a coupon code.	Disable
EAN Addon Option	Decide whether to decode EAN-8, EAN-13 with addons (including Addon 2 and 5). Ignore Addon Decode only with addons Auto-discriminate	Ignore Addon
EAN Addon Redundancy	When “Auto-discriminate” is applied, decide the number of times of supplementary decoding the same barcode to count as a valid read. Configurable between 2 and 30.	10
UPC Addon Option	Decide whether to decode UPC-E0, UPC-E1, UPC-A with addons (including Addon 2 and 5). Ignore Addon Decode only with addons Auto-discriminate	Ignore Addon

<table>
<tr><td>UPC Addon Redundancy</td><td>When “Auto-discriminate” is applied, decide the number of times of supplementary decoding the same barcode to count as a valid read. Configurable between 2 and 30.</td><td>10</td></tr>
<tr><td>UPC Security Level</td><td>Sets the security level to ensure decoding accuracy considering the printed quality of the barcodes such as Code 128, Code 93, and UPC/EAN. The higher the level is, the more security is ensured. Options are:
<table>
<tr><th>Level</th><th>Description</th></tr>
<tr><td>0</td><td>With this default, the scan engine is aggressive enough to decode most “in-spec” barcodes.</td></tr>
<tr><td>1</td><td>Select this level if misdecodes have occurred. It fixes most misdecodes.</td></tr>
<tr><td>2</td><td>Select this level if Level 1 should fail to eliminate misdecodes.</td></tr>
<tr><td>3</td><td>Select this level if Security Level 2 should fail to prevent misdecodes. However, as this level actually impairs the decoding ability of scan engine, it’d be better to improve the barcode’s print quality if this level should be needed.</td></tr>
</table></td><td>Level 2</td></tr>
</table>

MISCELLANEOUS

Laser Engine	Description	Default
Miscellaneous Options		---
AIM Code ID character	Decide whether to include AIM Code ID in the data. Each AIM Code ID contains a three-character string “**]cm**”:] = Flag Character (ASCII 93) c = Code Character (see below) m = Modifier Character (see below)	Disable

AIM CODE ID – CODE CHARACTERS

Code Character	Code Type
A	Code 39, Code 39 Full ASCII, Code 32
C	Code 128, Coupon (Code 128 portion)
d	Data Matrix
E	UPC/EAN, Coupon (UPC portion)
e	GS1 DataBar (RSS)
F	Codabar
G	Code 93
H	Code 11
I	Interleaved 25
L	PDF417, Macro PDF417, Micro PDF417
M	MSI
Q	QR Code
S	Industrial 25 (Discrete 25), IATA 2 of 5
U	Maxicode
X	Code 39 Trioptic, Bookland EAN, US Postnet, US Planet, UK Postal, Japan Postal, Australian Postal, Dutch Postal

AIM CODE ID – MODIFIER CHARACTERS

Code Type	Option Value	Option
Code 39	0	No check character or Full ASCII processing.
	1	Check digit has been verified.
	3	Check digit has been verified and stripped.
	4	Full ASCII conversion has been performed.
	5	Result of option values 1 and 4.
	7	Result of option values 3 and 4.
Code 128	0	Standard data packet. No Function Code 1“FNC1” in the first character position.
	1	Function Code 1“FNC1” in the first character position.
	2	Function Code 1“FNC1” in the second character position.
Interleaved 25	0	No check digit processing.
	1	Check digit has been verified.
	3	Check digit has been verified and stripped.
Codabar	0	No check digit processing.
Code 93	0	Always transmit 0.

MSI	0	Modulo 10 check digit verified and transmitted.
	1	Modulo 10 check digit verified but not transmitted.
Industrial 25 (Discrete 25)	0	Always transmit 0.
UPC/EAN	0	Standard data packet in full EAN country code format, which is 13 digits for UPC-A and UPC-E (not including addons).
	3	Standard data packet with two-digit or five-digit addons.
	4	EAN-8 data packet.
	A UPC-A with Addon 2 barcode, 012345678905-10, is transmitted to the host as a 18-character string, **]E3**001234567890510.	
Bookland EAN	0	Always transmit 0.
Trioptic Code 39	0	Always transmit 0.
Code 11	0	Single check digit (has been verified.)
	1	Two check digits (has been verified.)
	3	Check digit has been verified but not transmitted.
GS1 DataBar (RSS)	0	Always transmit 0.
	RSS-14 and RSS Limited will be transmitted with an Application Identifier “01”. For example, an RSS-14 barcode, 10012345678902, is transmitted as **]e0**0110012345678902.	

Note: In GS1-128 emulation mode, RSS is transmitted using Code 128 rules (= “]C1”).

EAN.UCC Composites (RSS, GS1-128, 2D portion of UPC composite)	Native mode transmission	
	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	
	1	Data packet is a GS1-128 barcode (= data is preceded with “]JC1”).

Note: UPC portion of composite is transmitted using UPC rules.

PDF417, Micro PDF417	0	Scan engine is set to conform to protocol defined in 1994 PDF417 symbology specifications. 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	Scan engine is set to follow the ECI protocol (Extended Channel Interpretation). All data characters 92_{DEC} are doubled.
	2	Scan engine is set for Basic Channel operation (no escape character transmission protocol). Data characters 92_{DEC} are not doubled. 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 barcode contains a GS1-128 symbol, and the first codeword is 903-907, 912, 914, 915.
	4	The barcode contains a GS1-128 symbol, and the first codeword is in the range 908-909.
	5	The barcode contains a GS1-128 symbol, and the first codeword is in the range 910-911.
	A PDF417 barcode, 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	Mode 4 or 5
	1	Mode 2 or 3
	2	Mode 4 or 5, ECI protocol implemented.
	3	Mode 2 or 3, ECI protocol implemented in secondary message.
QR Code	0	Model 1
	1	Model 2, ECI protocol not implemented.
	2	Model 2, ECI protocol implemented.
	3	Model 2, ECI protocol not implemented, FNC1 implied in first position.
	4	Model 2, ECI protocol implemented, FNC1 implied in first position.
	5	Model 2, ECI protocol not implemented, FNC1 implied in second position.
	6	Model 2, ECI protocol implemented, FNC1 implied in second position

Note: For JPEG files, these BPP settings are ignored for it always uses 8 bits per pixel!

Appendix IV

2D IMAGER (SE4500DL)

IN THIS CHAPTER

Symbology Settings for CP30/CP50.....129
Symbology Settings for CP60.....138

Symbology settings for 2D imager (SE4500DL) are:

SYMBOLOGY SETTINGS FOR CP30/CP50

1D SYMBOLOGIES

2D Engine	Description	Default
Codabar		**Enable**
CLSI editing	When applied, the CLSI editing strips the start/stop characters and inserts a space after the first, fifth, and tenth characters of a 14-character Codabar barcode. ▸ The 14-character barcode length does not include start/stop characters.	No
NOTIS editing	Sets whether to include start/stop characters in the transmitted data. ▸ NOTIS Editing is to strip the start/stop characters, i.e. to disable "Transmit Start/Stop Characters".	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Intercharacter Gap Size	The Code 39 and Codabar symbologies have an intercharacter gap that is typically quite small. Due to various barcode printing technologies, this gap can grow larger than the maximum size allowed, preventing the scan engine from decoding a barcode. If this problem occurs, set it to “Large Intercharacter Gaps” to tolerate these out-of-specification barcodes. ▸ Normal intercharacter gaps ▸ Large intercharacter gaps	Normal
Code 128		**---**
Code 128	Read standard Code 128 barcodes (= without leading FNC1 character).	Enable
GS1-128 (UCC/EAN-128)	Read GS1-128 barcodes with leading FNC1 character.	Enable
ISBT 128	Read ISBT 128 barcodes.	Enable

ISBT Concatenation	Decide whether to decode and concatenate pairs of ISBT barcodes. ▸ Disable ▸ Enable – When this option is selected, there must be two ISBT barcodes for the reader to decode and perform concatenation. ▸ Auto-discriminate – When this option is selected, the reader decodes and concatenates pairs of ISBT barcodes immediately. If only a single ISBT barcode is present, the reader must decode 10 times before transmitting its data to confirm that there is no additional ISBT barcode.	Disable
ISBT Concatenation Redundancy	When “Auto-discriminate” is applied, decide the concatenation redundancy (2~20 times).	10 times
Code 39		**Enable**
Trioptic	Decide whether to decode Trioptic Code 39. ▸ Trioptic Code 39 is a variant of Code 39 used in the marking of computer tape cartridges. It always contains six characters.	No
Convert Code 39 to Code 32	Convert to Italian Pharmacode.	No
Code 32 Prefix	Prefix character “A” to Code 32 barcodes. ▸ “Convert to Code 32” must be enabled for this to function properly.	No
Check digit verification	Decide whether to verify the check digit. If the check digit is incorrect, the barcode will not be accepted.	No
Transmit check digit	Decide whether to include the check digit in the data being transmitted. ▸ “Check digit verification” must be enabled.	No
Full ASCII conversion	Code 39 Full ASCII includes all alphanumeric and special characters (full ASCII character set).	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Intercharacter Gap Size	The Code 39 and Codabar symbologies have an intercharacter gap that is typically quite small. Due to various barcode printing technologies, this gap can grow larger than the maximum size allowed, preventing the scan engine from decoding a barcode. If this problem occurs, set it to “Large Intercharacter Gaps” to tolerate these out-of-specification barcodes. ▸ Normal intercharacter gaps ▸ Large intercharacter gaps	Normal
Code 93		**Enable**
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Industrial 25 (Discrete 25)		**Enable**
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55

EAN-8		**Enable**
Enable EAN-8		Enable
Enable EAN-13		Enable
Bookland EAN (ISBN)	EAN-13 barcodes starting with 978 will be converted to ISBN.	Enable
Enable Addons	Decide whether to decode EAN-8, EAN-13, UPC-E0, UPC-E1, UPC-A with addons (including Addon 2 and 5). ▸ Ignore Addons ▸ Decode Only With Addons ▸ Auto-discriminate	Ignore Addons
Addons redundancy	When “Auto-discriminate” is applied, decide the number of times (2~30) of supplementary decoding of the same barcode that makes a valid reading.	10 times
EAN-8 extend	Convert EAN-8 to EAN-13 format.	No
UCC coupon extended code	Read UPC-A barcodes starting with digit "5", EAN-13 barcodes starting with digits "99", and UPC-A/GS1-128 Coupon Codes. ▸ UPC-A, EAN-13, and GS1-128 must be enabled first! ▸ Use “Addon Redundancy” to control auto-discrimination of the GS1-128 (right half) of a coupon code.	No
UPC		**Enable**
UPC-A/UPC-E/UPC-E1	For the default value, the UPC-A and UPC-E are enabled.	Enable
Enable Addons	Decide whether to decode EAN-8, EAN-13, UPC-E0, UPC-E1, UPC-A with addons (including Addon 2 and 5). ▸ Ignore Addons ▸ Decode Only With Addons ▸ Auto-discriminate	Ignore Addons
Addons redundancy	When “Auto-discriminate” is applied, decide the number of times (2~30) of supplementary decoding the same barcode that makes a valid reading.	10 times
Transmit check digit	Decide whether to include the UPC-A/UPC-E/UPC-E1 check digit in the data being transmitted.	Yes
Convert UPC-E0/ UPC-E1 to UPC-A	The UPC-E0/UPC-E1 barcode will be expanded into UPC-A, and the next processing will follow the settings configured for UPC-A.	No
UCC coupon extended code	Read UPC-A barcodes starting with digit "5", EAN-13 barcodes starting with digits "99", and UPC-A/GS1-128 Coupon Codes. ▸ UPC-A, EAN-13, and GS1-128 must be enabled first! ▸ Use “Addon Redundancy” to control auto-discrimination of the GS1-128 (right half) of a coupon code.	No
Transmit preamble	Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted.	System Character
MSI		**Enable**

Check digit verification	One check digit is mandatory for decoding MSI barcodes. Select whether a second check digit should be verified. If the check digit is incorrect, the barcode will not be accepted. If Two Check Digits option is selected, select one of the algorithms below. ▸ Mod 10/Mod 11 ▸ Mod 10/Mod 10	Single Modulo 10
Transmit check digit	Decide whether to include the check digit in the data being transmitted.	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Interleaved 25		**Enable**
Check digit verification	Decide whether to verify the check digit. If desired, select one of the algorithms below. If the check digit is incorrect, the barcode will not be accepted. ▸ Disable ▸ USS check digit ▸ OPCC check digit	Disable
Transmit check digit	Decide whether to include the check digit in the data being transmitted.	No
Convert Interleaved 25 to EAN-13	Convert a 14-character barcode into EAN-13 if the following requirements are met: ▸ The barcode must have a leading 0 and a valid EAN-13 check digit. ▸ “Check digit verification” must be disabled.	No
Select Length (min/max)	▸ One or two fixed lengths ▸ Range (1~55)	4~55
GS1 DataBar (RSS)		**---**
GS1 DataBar-14	GS1 DataBar-14 is short for GS1 DataBar Omnidirectional. This group consists of (1) GS1 DataBar Omnidirectional, (2) GS1 DataBar Truncated, (3) GS1 DataBar Stacked, and (4) GS1 DataBar Stacked Omnidirectional.	Enable
GS1 DataBar Limited		Enable
GS1 DataBar Expanded	This group consists of (1) GS1 DataBar Expanded, and (2) GS1 DataBar Expanded Stacked.	Enable
Convert GS1 DataBar To UPC/EAN	“Convert to UPC/EAN” only applies to GS1 Databar-14 and GS1 Databar Limited barcodes not decoded as part of a Composite barcode. **Convert to EAN-13** Strip the leading “010” from barcodes. ▸ “01” is the Application ID and must be followed by a single zero (the first digit encoded)	No

	Convert to UPC-A Strip the leading “0100” from barcodes. ▸ “01” is the Application ID and must be followed by two or more zeros (but not six zeros)	
Code 11		**Enable**
Check digit verification	Decide whether to verify check digits. If the check digit is incorrect, the barcode will not be accepted. ▸ Disable ▸ One Check Digit ▸ Two Check Digits	Disable
Transmit check digit	Decide whether to include check digits in the transmitted data. ▸ “Check digit verification” must be enabled.	No
Select Length	▸ One or two fixed lengths ▸ Range (1~55)	4~55
Chinese 25		**Enable**
Composite Codes		**---**
Composite CC-C	Enable or disable Composite bar codes of type CC-C.	Enable
Composite CC-A/B	Enable or disable Composite bar codes of type CC-A/B.	Disable
Composite TLC-39	Enable or disable Composite bar codes of type TLC-39.	Disable
UPC Composite Mode	UPC barcodes can be "linked" with a 2D barcode during transmission as if they were one barcode. **UPC Never Linked** Transmit UPC barcodes regardless of whether a 2D barcode is detected. **UPC Always Linked** Transmit UPC barcodes and the 2D portion. If the 2D portion is not detected, the UPC barcode will not be transmitted. CC-A/B or CC-C must be enabled! **Auto-discriminate UPC Composites** Transmit UPC barcodes as well as the 2D portion if present.	UPC Always Linked
GS1-128 Emulation Mode for UCC/EAN Composite Codes	Transmit UCC/EAN Composite Code data as if it was encoded in GS1-128 barcodes.	Disable
Postal Codes		**---**
US Postnet		Enable
Transmit US Check Digit	US Postnet must be enabled first!	Enable

US Planet		Enable
UK Postal		Enable
Transmit UK Postal Check Digit	UK Postal must be enabled first!	Enable
Japan Postal		Enable
Australian Postal		Enable
Dutch Postal		Enable
Macro PDF		**Enable**
Macro PDF is a special feature for concatenating multiple PDF barcodes into one file, known as Macro PDF417 or Macro MicroPDF417.		
Transmit/Decode Mode	Decide how to handle Macro PDF decoding. **Buffer All Symbols / Transmit Macro PDF When Complete** Transmit all decoded data from an entire Macro PDF sequence only when the entire sequence is scanned and decoded. If the decoded data exceeds the limit of 50 symbols, no transmission because the entire sequence was not scanned! **Transmit Any Symbol in Set / No Particular Order** Transmit data from each Macro PDF symbol as decoded, regardless of the sequence. **Passthrough All Symbols** Transmit and decode all Macro PDF symbols and perform no processing. In this mode, the host is responsible for detecting and parsing the Macro PDF sequences.	Passthrough All Symbols
ESC Characters	When enabled, it uses the backslash "\" as an Escape character for systems that can process transmissions containing special data sequences. It will format special data according to the Global Label Identifier (GLI) protocol, which only affects the data portion of a Macro PDF symbol transmission. The Control Header, if enabled, is always sent with GLI formatting.	None
Note: When printing barcodes, keep each Macro PDF sequence separate, as each has a unique identifier. Do not mix barcodes from several Macro PDF sequences, even if they encode the same data. When you scan Macro PDF sequences, scan the entire Macro PDF sequence without interruption!		

Matrix 25		**Enable**
Transmit Check Digit	Decide whether to include the check digit in the data being transmitted.	No
Check digit verification	Decide whether to verify the check digit. If the check digit is incorrect, the barcode will not be accepted.	No
Redundancy	Decide whether to enable decode redundancy.	Disable
Select Length	▸ One or two fixed lengths ▸ Range (1~55)	4~55

2D SYMBOLOGIES

2D Engine	Description	Default
2D Symbologies		---
PDF417		Enable
MicroPDF417		Disable
MicroPDF417 Code 128 Emulation	Transmit data from certain MicroPDF417 barcodes as if it was encoded in Code 128 barcodes. Transmit AIM Code Identifier in **Miscellaneous Options** must be enabled first! When applied, the MicroPDF417 barcodes are transmitted with one of these prefixes: **The first codeword of MicroPDF417 is 903-907, 912, 914, 915:** The original Code ID "]L3" will be changed to "]C1". **The first codeword of MicroPDF417 is 908 or 909:** The original Code ID "]L4" will be changed to "]C2". **The first codeword of MicroPDF417 is 910 or 911:** The original Code ID "]L5" will be changed to "]C0".	Disable
Data Matrix		Enable
Data Matrix Inverse	Decide whether to decode Data Matrix Inverse. **Regular Only** Decode regular Data Matrix barcodes only. **Inverse Only** Decode inverse Data Matrix barcodes only. **Autodetect** Decode both regular and inverse Data Matrix barcodes.	Regular Only
Maxicode		Enable
QR Code		Enable
QR Code Inverse	Decide whether to decode QR Code Inverse. **Regular Only** Decode regular QR Code only. **Inverse Only** Decode inverse QR Code only.	Regular Only

	Autodetect Decode both regular and inverse QR Code.	
MicroQR		Enable
Aztec		Enable
Aztec Inverse	Decide whether to decode Aztec Inverse. **Regular Only** Decode regular Aztec barcodes only. **Inverse Only** Decode inverse Aztec barcodes only. **Autodetect** Decode both regular and inverse Aztec barcodes.	Regular Only

MISCELLANEOUS

2D Engine	Description	Default
Miscellaneous Options		---
Transmit Code ID character	Decide whether to include AIM Code ID in the beginning of data. None AIM code ID character (Refer to AIM Code ID – Code Characters)	none

SYMBOLOGY SETTINGS FOR CP60/9200

1D SYMBOLOGIES

Symbology	Description	Default
CODABAR		
CodaBar		
Codabar	Checkbox to enable Codabar decoding.	Enable
Length option	Sets the length of the Codabar symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
CLSI	When applied, the CLSI editing strips the start/stop characters and inserts a space after the first, fifth, and tenth characters of a 14-character Codabar barcode. The 14-character barcode length does not include start/stop characters.	Disable
NOTIS	Sets whether to include start/stop characters in the transmitted data. NOTIS Editing is to strip the start/stop characters, i.e. to disable "Transmit Start/Stop Characters".	Disable
CODE 11		
Code 11		
Code 11	Checkbox to enable Code 11 decoding.	Enable
Check Digit Option	Sets whether to verify check digits according to the selected option. If the check digits are incorrect, the barcode will not be accepted. Disable One Check Digit Two Check Digits	Disable
Transmit Check Digit	Selects whether to include check digits in the transmitted data. Check Digit Option" must be enabled.	Disable
Length option	Sets the length of the Code 11 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
CODE 39		
Code 39		
Code 39	Checkbox to enable Code 39 decoding.	Enable

Trioptic Code 39	Selects whether to decode Trioptic Code 39. Trioptic Code 39 is a variant of Code 39 used in the marking of computer tape cartridges. It always contains six characters.	Disable
Convert to Code 32	Selects whether to convert decoded data to Italian Pharmacode.	Disable
Code 32 Prefix	Prefix character “A” to Code 32 barcodes. “Convert to Code 32” must be enabled for this to function properly.	Disable
Verify Check Digit	Selects whether to verify the Modulo 43 check digit. If the check digit is incorrect, the barcode will not be accepted.	Disable
Transmit Check Digit	Decide whether to include the check digit in the data to transmit. “Verify Check Digit” must be enabled.	Disable
Support Full ASCII	Selects whether to enable Code 39 Full ASCII decoding. Characters are paired to encode the full ASCII character set.	Disable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
CODE 93		
Code 93		
Enable Code 93	Checkbox to enable Code 93 decoding.	Enable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
CODE 128		
Code 128		
Enable Code 128	Checkbox to enable Code 128 decoding.	Enable
GS1-128		
Enable GS1-128	Checkbox to enable GS1-128 decoding.	Enable
ISBT-128		
ISBT 128	Checkbox to enable ISBT 128 decoding.	Enable
Concatenation	Sets whether to enable decoding ISBT-128 by performing concatenation of ISBT data Disable: Does not perform concatenation Enable: Performs concatenation on all ISBT-128 barcodes. Auto-discriminate: Auto-discriminates between the ISBT-128 barcodes which require concatenation and those which do not need concatenation.	Disable
Redundancy	Sets redundancy between 2-20.	10

CODE 2 OF 5		
Chinese 25		
Enable Chinese 25	Checkbox to enable Chinese 25 decoding.	Enable
Discrete 25		
Discrete 25	Checkbox to enable Discrete 25 decoding.	Enable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
Interleaved 25		
Interleaved 25	Checkbox to enable Interleaved 2 of 5 decoding.	Enable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
Verify Check Digit	Disable USS Check Digit OPCC Check Digit	Disable
Transmit Check Digit	Decide whether to include the check digit in the data being transmitted.	Disable
Convert To EAN-13	Convert a 14-character Interleaved 25 barcode into EAN-13 if the following requirements are met: The barcode must have a leading 0 and a valid EAN-13 check digit.	Disable
Matrix 25		
Matrix 25	Checkbox to enable Matrix 2 of 5 decoding.	Enable
Length option	Sets the length of the Code 39 symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
Redundancy	Sets read redundancy	Disable
Verify Check Digit	Select whether to verify the check digit, which is the last character of the barcode. If the check digit is incorrect, the barcode will not be accepted.	Disable
Transmit Check Digit	Decide whether to include the check digit in the data being transmitted.	Disable

COMPOSITE		
Composite CC-A/B		
Enable Composite CC-A/B	Checkbox to enable Composite CC-A/B decoding.	Disable
Composite CC-C		
Enable Composite CC-C	Checkbox to enable Composite CC-C decoding.	Enable
Composite TLC 39		
Enable Code 128	Checkbox to enable Code 128 decoding.	Disable
General Preference		
UPC Composite Mode	UPC barcodes can be "linked" with a 2D barcode during transmission as if they were one barcode. **UPC Never Linked** Transmit UPC barcodes regardless of whether a 2D barcode is detected. **UPC Always Linked** Transmit UPC barcodes and the 2D portion. If the 2D portion is not detected, the UPC barcode will not be transmitted. CC-A/B or CC-C must be enabled. **Auto-discriminate** Transmit UPC barcodes as well as the 2D portion if present.	UPC Always Linked
GS1-128 Emulation Mode	Sets GS1-128 emulation mode for UCC/EAN Composite Codes.	Disable
GS1 DATABAR		
GS1 DataBar-14		
Enable GS1 DataBar-14	Checkbox to enable GS1 DataBar-14 decoding.	Enable
GS1 DataBar Limited		
Enable GS1 DataBar-Limited	Checkbox to enable GS1 DataBar-Limited decoding.	Enable
GS1 DataBar Expanded		
Enable GS1 DataBar-Expanded	Checkbox to enable GS1 DataBar-Expanded decoding.	Enable
Convert to UPC/EAN		
Convert to UPC/EAN	This only applies to GS1 DataBar-14 and GS1 DataBar Limited barcodes not decoded as part of a Composite barcode.	Disable

	Convert to EAN-13 Strips the leading “010” from barcodes. The barcode must be composed of a leading “01” as the application identifier (AI) and a first digit of zero. **Convert to UPC-A** Strips the leading “0100” from barcodes. The barcode must be composed of a leading “01” as the application identifier (AI) and two or more zeros (but not six zeros).	
INVERSE		
Inverse		
Enable Inverse	Checkbox to enable Inverse 1D decoding.	Disable
KOREAN 3 OF 5		
Korean 3 of 5		
Enable Korean 3 of 5	Checkbox to enable Korean 3 of 5 decoding.	Disable
MSI		
MSI		
MSI	Checkbox to enable MSI decoding.	Enable
Length option	Sets the length of the MSI symbols to decode. One Fixed length (Length 1) Two Fixed lengths (Length 1>Length 2) Max / Min Length (range: 0-255; Length 1<Length 2) Any Length	Max / Min Length (4-55)
Check Digit Option	One check digit is mandatory for decoding MSI barcodes. Select whether a second check digit should be verified. If the check digits are incorrect, the barcode will not be accepted. One Check Digit Two Check Digits	One Check Digit
Transmit Check Digit	Decide whether to include the check digit in the data being transmitted.	Disable
Algorithm	When two check digits are set for verification, two choices are available for the pair of check digits. Modulo10 / Modulo11 Double Modulo 10	Double Modulo 10
POSTAL CODE		
Australian Postal		
Enable Australian Postal	Checkbox to enable Australian Postal decoding.	Enable
Japan Postal		
Enable Japan Postal	Checkbox to enable Japan Postal decoding.	Enable
Netherlands KIX Code		

Enable Netherlands KIX Code	Checkbox to enable Netherlands KIX Code decoding.	Enable
US Postnet		
Enable US Postnet	Checkbox to enable US Postnet decoding.	Enable
US Planet		
Enable US Planet	Checkbox to enable US Planet decoding.	Enable
UK Postal		
Enable UK Postal	Checkbox to enable UK Postal decoding.	Enable
General Preference		
US Postal Check Digit	Decide whether to transmit check digit for US Postnet or US Planet.	Enable
UK Postal Check Digit	Decide whether to transmit check digit for UK Postal.	Enable
UPC/EAN		
EAN-8		
EAN-8	Checkbox to enable EAN-8 decoding.	Enable
EAN-8 Extend	Checkbox to enable converting EAN-8 to EAN-13 format.	Disable
EAN-13		
EAN-13	Checkbox to enable EAN-13 decoding.	Enable
Bookland EAN	Checkbox to enable ISBN decoding. If enabled, select Bookland ISBN Format in the drop-down box below.	Enable
Bookland ISBN Format	Decodes Bookland data starting with 978 in 10-digit format along with the Bookland check digit, or Bookland data starting with 978/979 as EAN-13 format.	Bookland ISBN-10
ISSN EAN	Checkbox to enable ISSN EAN decoding.	Disable
UPC-A		
UPC-A	Checkbox to enable UPC-A decoding.	Enable
Preamble	Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted. No transmit: transmits none of the above Transmit System Character: transmits system number only Transmit Sys. Character and Country Code: transmits system number and country code	Transmit System Character
Transmit Check Digit	Decide whether to include the UPC-A check digit (the last character in the barcode) in the data being transmitted.	Enable
UPC-E		
UPC-E	Checkbox to enable UPC-E decoding.	Enable

Preamble	Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted. No transmit: transmits none of the above Transmit System Character: transmits system number only Transmit Sys. Character and Country Code: transmits system number and country code	Transmit System Character
Convert to UPC-A	The UPC-E barcode will be expanded into UPC-A format, and the next process will follow the settings configured for UPC-A.	Disable
Transmit Check Digit	Decide whether to include the UPC-E1 check digit (the last character in the barcode) in the data being transmitted.	Enable
UPC-E1		
UPC-E1	Checkbox to enable UPC-E1 decoding.	Disable
Preamble	Decide whether to include the UPC-A/UPC-E/UPC-E1 preamble System Character (and Country Code) in the data being transmitted. No transmit: transmits none of the above Transmit System Character: transmits system number only Transmit Sys. Character and Country Code: transmits system number and country code	Transmit System Character
Convert to UPC-A	The UPC-E1 barcode will be expanded into UPC-A format, and the next process will follow the settings configured for UPC-A.	Disable
Transmit Check Digit	Decide whether to include the UPC-E1 check digit (the last character in the barcode) in the data being transmitted.	Enable
General Preference		
Support Coupon Code	Reads UPC-A barcodes starting with “5”, EAN-13 barcodes starting with “99”, and UPC-A/EAN-128 Coupon Codes. UPC-A, EAN-13, and GS1-128 must be enabled first! Use “Addon Redundancy” to control auto-discrimination of the GS1-128 (right half) of a coupon code.	Disable
EAN Addon Option	Decide whether to decode EAN-8, EAN-13 with addons (including Addon 2 and 5). Ignore Addon Decode only with addons Auto-discriminate	Ignore Addon
EAN Addon Redundancy	When “Auto-discriminate” is applied, decide the number of times of supplementary decoding the same barcode to count as a valid read. Configurable between 2 and 30.	10
UPC Addon Option	Decide whether to decode UPC-E0, UPC-E1, UPC-A with addons (including Addon 2 and 5). Ignore Addon Decode only with addons Auto-discriminate	Ignore Addon
UPC Addon Redundancy	When “Auto-discriminate” is applied, decide the number of times of supplementary decoding the same barcode to count as a valid read. Configurable between 2 and 30.	10

2D SYMBOLOGIES

Symbology	Description	Default
Aztec		
Aztec	Selects whether to enable Aztec decoding.	Enable
Aztec Inverse	Decide whether to decode Aztec Inverse. **Regular only** Decode regular Aztec barcodes only. **Inverse only** Decode inverse Aztec barcodes only. **Inverse Auto-detect** Decode both regular and inverse Aztec barcodes.	Regular only
Data Matrix		
Data Matrix	Selects whether to enable Data Matrix decoding.	Enable
Data Matrix Inverse	Decide whether to decode Data Matrix Inverse. **Regular Only** Decode regular Data Matrix barcodes only. **Inverse Only** Decode inverse Data Matrix barcodes only. **Auto Detect** Decode both regular and inverse Data Matrix barcodes.	Regular Only
Decode Mirror Images	Selects whether to enable PDF 417 decoding. **Never** Does not decode Data Matrix barcodes that are mirror images. **Always** Decodes Data Matrix barcodes that are mirror images. **Auto** Decodes both mirrored and unmirrored Data Matrix barcodes.	Never
Maxicode		
MicroPDF417		
MicroPDF417	Selects whether to enable Maxicode decoding.	Disable

Code 128 Emulation	Transmit data from certain Micro PDF 417 barcodes as if it was encoded in Code 128 barcodes. Transmit AIM code ID character in **Miscellaneous** options must be enabled first. When applied, the MicroPDF417 barcodes are transmitted with one of these prefixes: **The first codeword of MicroPDF417 is 903-905:** The original Code ID "]L3" will be changed to "]C1". **The first codeword of MicroPDF417 is 908 or 909:** The original Code ID "]L4" will be changed to "]C2". **The first codeword of MicroPDF417 is 910 or 911:** The original Code ID "]L5" will be changed to "]C0".	Disable
MicroQR		
PDF417		
QR Code		
QR Code	Selects whether to enable QR Code decoding.	Enable
QR Code Inverse	Decide whether to decode QR Code Inverse. **Regular Only** Decodes regular QR Code only. **Inverse Only** Decodes inverse QR Code only. **Inverse Auto-detect** Decodes both regular and inverse QR Codes.	Regular Only

MISCELLANEOUS

2D Engine	Description	Default
Miscellaneous Options		
AIM Code ID character	Decide whether to include AIM Code ID in the data. Each AIM Code ID contains a three-character string "**]cm**":] = Flag Character (ASCII 93) c = Code Character m = Modifier Character Refer to AIM Code ID – Code Characters.	Disable

Appendix IV

HF RFID READER

The HF RFID reader supports ISO14443A and B, 15693 and Secure Access Module (SAM) for CP50 and 9200.

RFID TAG DEFAULT BLOCK

Tag Type	Standard	Default Block/Page
Mifare	ISO 14443A	4
SR176	ISO 14443B	4
ICODE SLI	ISO 15693	3
LRI512	ISO 15693	0
SRF55VxxP	ISO 15693	3
EM4135	ISO 15693	0
Tag-it HF-I	ISO 15693	0
Others	ISO 15693	0
ICODE	ICODE® (Phillips)	5
Tag-it	Tag-it® (TI)	0

Appendix V

RESPONSE CODE INSTRUCTIONS

Name	Value	Instruction
LRSUCCESS	0x00	Successful completion of request
LRSYSTEM	0x01	Unknown error
LRLASTCARD	0x02	Last Card Still Present
LRNOCARD	0x03	Card is not present
LRCTYPE	0x04	Card Type error
LRPARAM	0x05	Request Parameter error
LRACCESS	0x06	Card access error
LRREAD	0x07	Card read error
LRWRITE	0x08	Card write error
LRINCR	0x09	Purse increment error
LRDECR	0x0A	Purse decrement error
LRTRANSFER	0x0B	Purse value transfer error
LRRESTORE	0x0C	Purse restore error
LRPURSE	0x0D	Purse value corrupt
LRMADERR	0x0E	Card Directory error
LRFIXERR	0x0F	Purse fix error
LRFIXED	0x10	Purse found corrupt but fixed
LRNOTOPEN	0x11	Card not open
LRNOFILE	0x12	File not found
LRBADSIZE	0x13	Bad file size
LRABORTED	0x14	Request aborted
LRMANYCARD	0x15	Too many card present
LRFORMAT	0x16	Card format error
LRCREATE	0x17	Card file create error
LRDELETE	0x18	Card file delete error
LRALREADOPEN	0x19	Card has been opened already

LRALREADCLOSED	0x1A	Card has been closed already
LRMSTRKEYLOAD	0x1B	Cannot load master keys
LRAPPKEYLOAD	0x1C	Cannot load application keys
LRKEYCARD	0x1D	Keycard Error
LRUNFORMAT	0x1E	Card has files on it
LRNOKBDCHAR	0x20	No keyboard character
LRAPIWRITE	0x21	API data write error
LRAPIREAD	0x22	API data read error
LRBLOCKERROR	0x23	Block number error
LRNOTIMPL	0x7F	Function not implemented
LRUNKNOWN	0x80	Unknown error
LRCCRBUSY	0xBB	Reader is busy
LRCHANGEWROKTYPE	0xEE	Change Reader working type error
LRRDUNKNOWN	0xEF	Reader Unknown error
LRCRDNOTOPEN	0xFA	Card has not been opened
LRINUSE	0xFB	Card in use by another applications
LRAPPLICERR	0xFC	API system error
LRRDLINKLOS	0xFD	Link to Reader has been lost
LRBADCOMPORT	0xFE	COM port cannot be accessed
LRRDNOINIT	0xFF	Reader has not been opened
LRNOCRYPTBOX	0xF9	Key-Box not found
LRBADAPPACCESS	0xF8	Invalid Application access code
LRBADRDACCESS	0xF7	Invalid Reader access code
LRNOMAIDFILE	0xF6	Cannot open MAID definition file
LRBOXREAD	0xF5	Cannot read from Key-Box
LRBOXWRITE	0xF4	Cannot write to Key-Box
LRBOXNOKEYS	0xF3	No of Keys in Box is zero or invalid
LRSECURE	0xF2	Comms MAC checking failed
LRERRSELREADER	0xF1	Cannot change to selected reader
LRRDNORESP	0xF0	Reader No Response