



StayLinked

Scan Configuration

CONTENTS

1	INTRODUCTION	1
1.1	Testing Scan Input.....	1
1.2	Input Types	2
2	MAKING ADJUSTMENTS TO SCAN INPUT	3
2.1	Special Characters in Barcodes	4
2.2	Symbology Identifiers and scanner.ini.....	4
3	SCANNER-AS-WEDGE.....	5
3.1	Configuring the StayLinked Client	5
3.2	Configuring the Scanner and Device.....	5
3.3	Testing Your Configuration	6
4	AUTOWEDGE.....	7
5	SCAN ENGINE SETTINGS.....	8
5.1	Tethered Scanners	8
6	ANDROID DEVICES BY HONEYWELL	9
7	ANDROID DEVICES BY ZEBRA.....	10
7.1	Creating a DataWedge Profile Manually	10
7.2	Testing Intent Configuration	16
7.3	Adjusting the Scan Engine	17
7.4	Importing DataWedge Profiles	17
8	SPP BLUETOOTH SCANNERS	18
9	ANDROID DEVICES BY DATALOGIC.....	19
10	APPENDIX – SYMBOL/MOTOROLA SCANNER.INI.....	20
11	CONTACT INFORMATION.....	26

1 Introduction

StayLinked supports a huge range of mobile devices and scanners. Most of these devices will automatically detect the difference between scan and keyboard inputs. Differentiating input types allows StayLinked to apply scan input adjustments like automatic returns as well as making the Client-Server transactions more reliable and efficient (See Input Types, below).

The first step is to use the StayLinked Client scan test. StayLinked clients come in two general forms, customized and generic. The custom clients are developed with the hardware in mind. Support for the device keyboard, radio and scan engine are included in the client. Custom clients work through the operating system to communicate with the hardware, and natively support various features. Generic clients do not contain these specific features and have limited capabilities when attempting to interface with the device hardware. Some custom clients do not include scanner support or may be used with any number of attached external scanners.

General Overview:

- 1) Install StayLinked client on your device.
- 2) Test the scanner using the StayLinked Client Scan Test.
 - a. If the scan data appears in the scan test, stop.
 - b. If scan data does not appear, see below for configuration options.
- 3) Enable the required Intents (Zebra/DataLogic), Scan Configuration scanner.ini (Honeywell), Scanner-as-Wedge, or AutoWedge settings on your device.
- 4) Retest the scanner using the client scan test. Check your configuration if data still does not appear in the test. Be sure to test with multiple barcodes to make sure the issue is not specific to a barcode symbology or contents.

Additional configuration may also be required to see the symbology or barcode type.

1.1 Testing Scan Input

All StayLinked clients include a scanner and keyboard test. Using these tests, you can see what data appears as scanner or keyboard input. It is possible on some devices that scan input will appear in both the scan and keyboard tests. This makes the scan test the best confirmation that the data is recognized as a scan. Data appearing in both places will only be transmitted to the server as scan data.

StayLinked also offers a Server-side scan test. The test is available from the StayLinked Menu (see user experience guide for more details on the StayLinked Menu). This menu is executed during a session from the Tools Menu or using a keyboard map entry. The client-side scan test shows the exact data as it is provided by the device. The Server-side test shows the barcode data, length, type, and a trans field that shows the data after any server-side rules are applied.

```
File Host Tools
Scanner Test
Scan: _____
Trns: _____
Len:
Type:
Code:  ()
(Q)uit
```

1.2 Input Types

The primary feature in any StayLinked Client is to reliably transmit user inputs to the StayLinked Server in the order they are input. These inputs can come from the keyboard, scanner or touch screen. Each transaction is submitted to the StayLinked Server with a sequence number, and each client has a limit to the number of transactions it can buffer. This buffer defaults to 16 transactions on most devices.

Each tap or keyboard input is transmitted to the Server as a single transaction. Each barcode scan is one transaction, except when scanning is not configured, and the data appears as high-speed keyboard input. This means that a 12-digit barcode is either a single 12-digit scan transaction or 12 one-digit keyboard transactions. This transaction buffer can be easily overflowed causing data to be lost when scanning long barcodes, providing input during weak network connectivity, or when scanning multiple barcodes in quick succession.

StayLinked iQ features and Statistics Logging can show information about how many transactions are being entered as keyboard input and scans. Individual log files for the 'Session' and 'Handler' log files will show on a case-by-case basis which entries are submitted as scan data or keyboard input. StayLinked iQ features are described in the iQ documentation, statistics logging is enabled from the Usage Tracking features and the session-specific logs are enabled in Server Settings but can be viewed from the Connections List.

The following are examples of the same barcode input, first as a proper scan transaction and then as it would appear if it was submitted as keyboard input. These log entries are pulled from the StayLinkedHandler.log file from a common mobile device.

Scan:

```
09/16/2016 10:47:25:218 - esp_ProcessDeviceInput() - RECV: (000C) 'AD~12~0~08~05244410~~'
```

Wedge:

```
09/16/2016 10:47:56:027 - esp_ProcessDeviceInput() - RECV: (0005) 'AC~414101~Ú~' "A"
09/16/2016 10:47:56:138 - esp_ProcessDeviceInput() - RECV: (0006) 'AC~303000~~'
09/16/2016 10:47:56:298 - esp_ProcessDeviceInput() - RECV: (0007) 'AC~353500~E~'
09/16/2016 10:47:56:411 - esp_ProcessDeviceInput() - RECV: (0008) 'AC~323200~Ü~'
09/16/2016 10:47:56:472 - esp_ProcessDeviceInput() - RECV: (0009) 'AC~343400~S~'
09/16/2016 10:47:56:585 - esp_ProcessDeviceInput() - RECV: (000A) 'AC~343400~C~'
09/16/2016 10:47:56:749 - esp_ProcessDeviceInput() - RECV: (000B) 'AC~343400~Ö~'
09/16/2016 10:47:56:862 - esp_ProcessDeviceInput() - RECV: (000C) 'AC~313100~~'
09/16/2016 10:47:56:918 - esp_ProcessDeviceInput() - RECV: (000D) 'AC~303000~~'
```

The scan entry shows the input type as AD, while the keyboard inputs appear as AC. You can see that the keyboard transactions appear within milliseconds, though a high-latency connection could impact what the user sees at the device. In this case, the characters might appear one at a time or in short bursts of network communications.

While not obvious in this log, you would also see the first character transmitted is a capital A. This represents the symbology type that was included by the scan engine and is normally removed during the transaction to populate the symbology type in the StayLinked Client. ASCII character values are included in red to demonstrate the barcode data.

1.2.1 Requiring Scan Input

Once scan input can be differentiated from keyboard input, StayLinked can use IBM host field attributes or Screen Recognition features to restrict input based on input method. This can be used to prevent users from typing data that should be scanned. Other options are available, but the use of the barcode scanner is the most common goal of handheld mobile devices in most cases.

2 Making Adjustments to Scan Input

Each layer involved in the input of data can be configured to adjust that data. For example, devices with integrated scanners often have control panel applets that allow you to create rules. Devices running programs like DataWedge are typically not hardware-specific but can likewise include rules or adjustments to barcode data. Data passed from the device to StayLinked can then have StayLinked adjust the data before submitting it to the telnet host. The telnet host may include its own adjustment to incoming data such as standardizing formats or data length.

The following notes and considerations may be important when deciding where barcode adjustments should be made in your environment:

On the device (DataWedge / Scan Engine) – Making adjustments at this level often require touching or configuring every device. Some third-party tools may help standardize the configuration, but different device manufacturers offer different tools and utilities for manipulating barcode data. Some manufacturers do not provide simple scan engine configuration, instead relying on third-party vendors to configure the hardware as required for their application.

In some cases, a scan beam may shine over a barcode as though it was a blank paper. When the scan beam does not acknowledge a barcode, one of two things is usually the issue:

- 1) The scan engine is not configured to decode the barcode type, or the barcode length is outside the minimum or maximum length specific for that barcode type, or
- 2) The barcode itself may be defective. Using another type of device to try and read the barcode is usually the best way to see if a barcode is valid. Most modern smart phones can download a free application to recognize barcodes and decode their contents.

Within StayLinked – Making adjustments to data can be configured for all devices, or groups of devices with one or more rules. StayLinked supports adjustments to barcode data in three different places. In areas 1 and 2 below, the adjustments are specific to a barcode symbology. You can confirm what StayLinked receives as the barcode type by using the StayLinked Client Scan Test as mentioned above. The three areas for barcode adjustments in StayLinked include:

- 1) Telnet Host Entry – When you configure the rules for communicating with a telnet host, you can select the Manage pulldown menu from the top of the screen to set barcode rules that apply only to that telnet host.
- 2) Device Groups – Rules added to the device group have priority over settings in the telnet host. This allows you to create global rules in the host entry, and then exceptions in the individual groups that require special adjustments.
- 3) Screen Recognition – These features can be used to apply 'Input Modifiers' to specific screens or fields. More information on Input Modifiers can be found in the SmartTE and Screen Recognition User's Guide.

All three of these locations support the use of regular expressions. Several common configuration options, such as trimming spaces, replacing certain characters and prefix/suffix additions can be made in the basic configuration dialogues. More complicated adjustments may require the use of regular expressions. More information on regular expressions can be found in the Regular Expression Guide or using online materials. These expressions make up a simplified syntax for identifying patterns and returning the desired results.

At the Hosted Application – Some hosted applications may track data or restrict input based on barcode data. One example would be the use of a prefix character on barcodes that cannot be typed. When this character appears at the beginning of a barcode, the host application can strip the character off and then proceed with the transaction with the confirmation that the data has been scanned. Special prefixes may also be used to identify the type of barcode scanned, helping to eliminate the entry of incorrect barcodes into the hosted application.

2.1 Special Characters in Barcodes

Some symbologies such as GS1 can display a bracket character at the beginning of each barcode. Bracket characters are processed by StayLinked as part of a mnemonic. These mnemonics, like [enter] and [tab] perform a function instead of simply typing the word into the application. These mnemonic examples would perform [enter] as a Carriage Return and Line Feed (0x0A0D) or a tab key (0x09).

Single brackets are treated as a failed mnemonic, preventing the barcode data from being submitted to the hosted application by the StayLinked Server. The barcodes would appear in the StayLinked Client Scan Test, but the server-side test would show the issue in the Translation. Double brackets are translated as a single. The adjustment can be made in the scan engine, or at the StayLinked Server as mentioned above. The most common setting to allow the bracket character to the host application would be a substitution of the single bracket to a double bracket as shown here to the right.

Add New Barcode

Regular Expression

Barcode Type:

Decode this type: ☒

Minimum Length: Maximum Length:

Trim all spaces from: Neither Side

Substitute String:] with String:]] Set both Min Length and Max Length to zero to impose no length restriction

Substitute all embedded Control Codes with this character:

Trim from Beginning: 0 Trim from End: 0

Use First ## Digits: (Enter Zero to keep the entire barcode)

Regular Expression Processing:

Expression	Replacement

Scan Prefix: Scan Suffix:

Notes:

Tab Characters can sometimes be entered in concatenated barcodes to split one barcode across multiple fields of data. This may work in 5250 because of the block-mode nature of the emulation. VT emulation does not contain actual fields and involves some amount of processing between changes in displayed input location. This means VT can only parse the data across input entries if the host can process each input fast enough to keep up or can store the data as it is entered for the forthcoming input prompts.

Some barcodes can include embedded control codes. These codes are passed on to the host application as part of the input string but may cause the host issues if it is not prepared to handle the events. These control codes can be globally replaced in the barcode rules above by setting the replacement character. The replacement can be a standard printable character or a simple NULL value 0x00. Control codes adjusted by this setting are any value in hexadecimal below 0x20. 0x20 and codes above will not be affected by this option.

2.2 Symbology Identifiers and scanner.ini

Devices that require scanner configuration may provide an option for a symbology identifier. This value tells the program what type of barcode has been scanned. Without this option, all barcodes may be reported as a barcode type of Unknown/Unsupported. It is always important to make sure any barcode rules are applied to the correct symbology type and will not adversely affect barcodes used on other screens, of different lengths, or barcode that contain different types of data.

Devices that require customization of these symbology identifiers can use a custom configuration file called scanner.ini. More information on this configuration file can be found in the StayLinked Client User Guide for the correct version of your StayLinked Client.

3 Scanner-as-Wedge

The most reliable method of capturing scan data on devices without native scanner communications is to use Scanner-as-Wedge. This feature allows the StayLinked Client to monitor inputs for a 'delimiter' character. When this character appears as part of incoming data, the Client will bundle all data between delimiter characters and submit it to the StayLinked Server as a scan transaction. This delimiter character defaults on most clients to the caret symbol ^. Whatever character is selected, it must not be a character required in scanned barcode data and cannot be used by the device operating system for system functions. For example, the ^ symbol is used by some device manufacturers as a reference to the escape key and would not work on those devices.

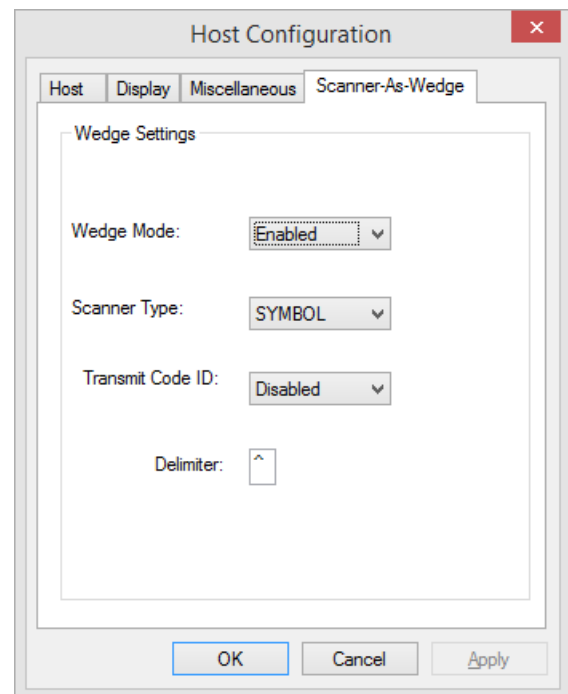
Configuration is typically done in a few places. When everything is correctly configured, your barcode data should appear in the StayLinked client scan test.

3.1 Configuring the StayLinked Client

StayLinked Client configuration panels contain a tab for Scanner-As-Wedge. The first option on this dialogue will disable all the other configuration choices if it is set to disabled. A setting of 'Disabled' means that the client will attempt to communicate with the scanner natively and will not monitor the keyboard interface for the delimiter. Various tethered scanner types can be selected to match the manufacturer of your scanner. The code ID is an extra character that can be included by the scan engine to communicate the barcode symbology. The delimiter is the character that should encapsulate your barcode data.

If the code ID is incorrectly configured, you will see one too many or one too few characters at the beginning of the barcode data.

These values can be set at the StayLinked Server using the Client Settings features. It is important that scanner-as-wedge is not accidentally enabled for devices with native scanner communication as it will prevent them from properly submitting barcode transactions to the StayLinked server. More information is listed above in the section for Input Types.



3.2 Configuring the Scanner and Device

The device or wedge program will need to be configured to include a prefix and suffix that matches the StayLinked Scanner-As-Wedge Delimiter. The default delimiter is the caret symbol '^', typically displayed on a PC keyboard as a shifted 6. Your device may also have a configuration option for the symbology ID, which is a prefix that identifies the type of barcode being scanned.

The configuration choices in the scanner and StayLinked client will need to match for your barcodes to be correctly processed in the StayLinked client.

Please refer to the manufacturer or third-party documentation for the best method of configuring your scanner.

3.3 Testing Your Configuration

Once configuration of the scanner is complete, the device should be able to scan into notepad or WordPad in one of the two following formats:

`^Xbarcodedata^`
`^barcodedata^`

The caret symbols represent the delimiter and can be configured for an alternative character when desired. Some devices use the caret symbol to represent an escape, which is incompatible. The `X` represents the symbology ID, if you have enabled it in the scanner and set the transmit code id in StayLinked to disabled.

The barcode data should represent only the data contained in the barcode you are scanning.

Once the configuration is complete, the StayLinked client scan test should display the barcode data only and the keyboard test should not respond to barcode scans. If the data appears to quickly scroll by the keyboard test, and does not appear in the scan test, check the WordPad scan results as mentioned above.

Since StayLinked simply watches the keyboard interface for the barcode string, any missing data or inconsistencies will need to be configured in the scanner configuration. Please refer to the manufacturer of your device for details on configuring their products.

4 AutoWedge

AutoWedge is an 'automatic' mode for configuring Scanner-as-Wedge. This feature was added to clients starting with v11.0 build 119 and newer. This allows the client to 'estimate' when scan transactions are coming through the keyboard interface based on the speed of the input. This feature has some limitations, since false positives and single character scans cannot be captured by the program logic. This feature is intended as a 'common use' solution and has limited adjustments other than timing and an enabled/disable option. If more precise control is required, the scan delimiters discussed above are the best available solution.

Enabling AutoWedge is done by using a combination of two StayLinked client settings;

- 1) The first is the scanner as wedge option, which must be set to enabled.
- 2) The second required setting is to set the wedge delimiter to 0.

A third available setting can adjust the timing between characters.

Scanner as Wedge Mode = Enabled

None of the configuration choices for Scanner-As-Wedge will operate unless the setting group is enabled. Once enabled, the client will not use any standard communication with the scan engine and features that enable/disable the scanner will not operate.

Tethered Scan Delimiter = 0

Setting the delimiter to a 0 is the key to enabling AutoWedge. Any other delimiter will act as a standard delimiter to contain barcode information.

Wedge Character Gap Time = 10

(10 is recommended only for use with this feature)

This setting controls the number of milliseconds between input that will be grouped as a scan by the client. The default setting is 100 milliseconds, which may be long enough to capture keys that are manually depressed and automatically repeat. When a character comes across the keyboard input channel, the client will wait the listed delay for additional characters. If more than one character is seen within this time limit, they will be packaged, transmitted and processed by the server as a scan. This can include scan suffix or barcode rules that have been configured at the StayLinked Server.

The screenshot shows the 'Add Client Setting' dialog box. The 'Setting Information' section has 'Group' set to 'Scanner and Keyboard' and 'Setting' set to 'Scanner as Wedge'. The description reads: 'Set to on to have the client software process scanner data from the wedge reader interface.' The 'Enabled' checkbox is checked, and 'Show Advanced' is also checked. The 'Level Information' section shows 'Setting Level' as 'Device Type' and 'Device Type' as '*'. The 'Setting Value' section shows 'Value' as 'On'. There is a 'Notes' text area at the bottom.

The screenshot shows the 'Add Client Setting' dialog box. The 'Setting Information' section has 'Group' set to 'Scanner and Keyboard' and 'Setting' set to 'Tethered Scanner Delimiter'. The description reads: 'Sets the character to use as the scan data delimiter when using the tethered scanner. (Default="')'. The 'Enabled' checkbox is checked, and 'Show Advanced' is also checked. The 'Level Information' section shows 'Setting Level' as 'Device Type' and 'Device Type' as '*'. The 'Setting Value' section shows 'Value' as '0'. There is a 'Notes' text area at the bottom.

The screenshot shows the 'Add Client Setting' dialog box. The 'Setting Information' section has 'Group' set to 'Scanner and Keyboard' and 'Setting' set to 'Wedge Character Gap Time'. The description reads: 'Determines the number of milliseconds to wait for barcode characters between delimiters. (10 to 1000, Default=100)'. The 'Enabled' checkbox is checked, and 'Show Advanced' is also checked. The 'Level Information' section shows 'Setting Level' as 'Device Type' and 'Device Type' as '*'. The 'Setting Value' section shows 'Value' as '10'. There is a 'Notes' text area at the bottom.

5 Scan Engine Settings

HHP, Symbol and Motorola devices running Windows-based operating systems typically do not include any scan-engine configuration options provided by the manufacturer. Symbol's ScanSamp2 is often included on these device models but will only provide configuration option while running and will be overwritten by the default configuration options when the StayLinked Client attempts to activate the scan engine.

On Windows CE and Windows Mobile operating systems, DataWedge may allow you to configure the scanner, but these settings are not always persistent (surviving a device cold reset). Using DataWedge will also affect the StayLinked Client's ability to automatically communicate with the device scanner. Be sure to test your DataWedge configuration after a cold reset of the device if you plan to use the configuration options that it can provide. After DataWedge configuration has been established, be sure to use the StayLinked Client scan test to confirm your configuration still allows StayLinked to differentiate scan and keyboard input.

StayLinked Clients can control configuration options through the use of a custom configuration file called scanner.ini. While the same name as the scanner.ini listed above, this file is specific to the scan engine configuration of specific device models. More details on these features can be found in the Stay-Linked Symbol Integrated Scanner Config Guide and the Client User Guide that matches the device model and client version for your device.

See Appendix 1 at the end of this guide for an example of the symbol scanner.ini file.

5.1 Tethered Scanners

Many scanning devices can use programming barcodes. If the scanning device can be configured independently, it's recommended that configuration is only set in one location. Configurations can be easily confused when devices can be configured in the tethered scanner, at the mobile computer, and in StayLinked. This is further complicated when users move the scanning device from one mobile computer to another.

StayLinked recommends that all customization is configured within StayLinked when possible. As long as StayLinked can differentiate scan and keyboard input, devices can be moved and reset without losing required configuration.

StayLinked offers programming barcodes for a small number of handheld scanners. We always recommend using the manufacturer's preferred method of configuring your scanner in order to work with the StayLinked Scanner-as-Wedge features.

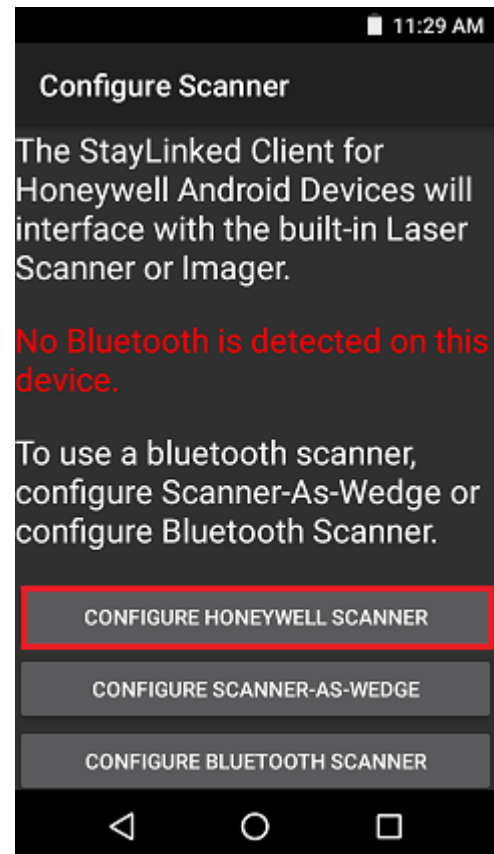
6 Android Devices by Honeywell

Honeywell Android devices offer scan configuration in system dialogues. In order for these settings to work within StayLinked, the client on the device must create an instance of the configuration. This scan configuration is stored in a file called scanner.ini. Various StayLinked client use this file name for different reasons, so a scanner.ini from another manufacturer of device will not have the same format or contents. Once an instance of the scanner.ini has been created, it can be copied to other Honeywell devices for use by StayLinked.

Creating this configuration file is accomplished automatically by launching the scanner configuration utility within StayLinked. **The client will operate the scanner with the default scanning options provided by the manufacturer's SDK** until this utility has been called by StayLinked or a file has been distributed with the proper configuration and file format. Custom configurations of the scanner will not be utilized by StayLinked until they appear in the scanner.ini used by the client on Honeywell devices.

Navigating to this dialogue (right) is accomplished by selecting the Host menu > Configure > enter the client security password (default is esp) > Scanner-As-Wedge. Note that Host > Configure is only available while the client is not connected to a StayLinked server.

Attaching external scanners or devices that are not controlled by the Honeywell Scanner Configuration utility can still be accomplished using the other scan configuration options.



7 Android Devices by Zebra

Android devices support the option for intents. These create a custom channel between DataWedge and the StayLinked Client, allowing scans to be separated from keyboard input. This requires creating a DataWedge profile specific to the application, in this case StayLinked. These DataWedge profiles can be exported and imported, saving some of these steps as long as you can access the export file. See the import section below for more details.

Attached to this guide (not visible if viewed within a browser) are two files `datawedge_full.db` and `datawedge_profile.db`. These are pre-generated DataWedge intent configurations. If you have DataWedge version 5 or newer, this file can be downloaded and sent to your devices using a client setting that will automatically create the required configuration on your devices. The `datawedge_profile.db` file is a single profile that is added to your existing list (recommended). Note that importing the `datawedge_full.db` file will replace all existing DataWedge profiles on your device.

If the device has an older version of DataWedge, you will need to manually import the file. It may need to be renamed `datawedge.db` for DataWedge to recognize the file.

If your DataWedge version supports automatic imports, the autoimport path should be included in the device file name field:
`\enterprise\device\settings\datawedge\autoimport`

Devices using any scanner supported by DataWedge can use this configuration. This includes Bluetooth devices that are compatible with DataWedge. This interface will operate at a higher speed and accuracy than the Scanner-as-Wedge configuration mentioned above and is highly recommended for any device with DataWedge support. Even if your device does not include DataWedge, you may be able to download it and install DataWedge from the manufacturer's website.

The default path and filename for manual imports are:

`/storage/sdcard0/Android/data/com.symbol.datawedge/files/dwprofile_staylinked.db`

If an external SD card is installed, an alternate path can be selected:

`/storage/sdcard1/Android/data/com.symbol.datawedge/files/dwprofile_staylinked.db`

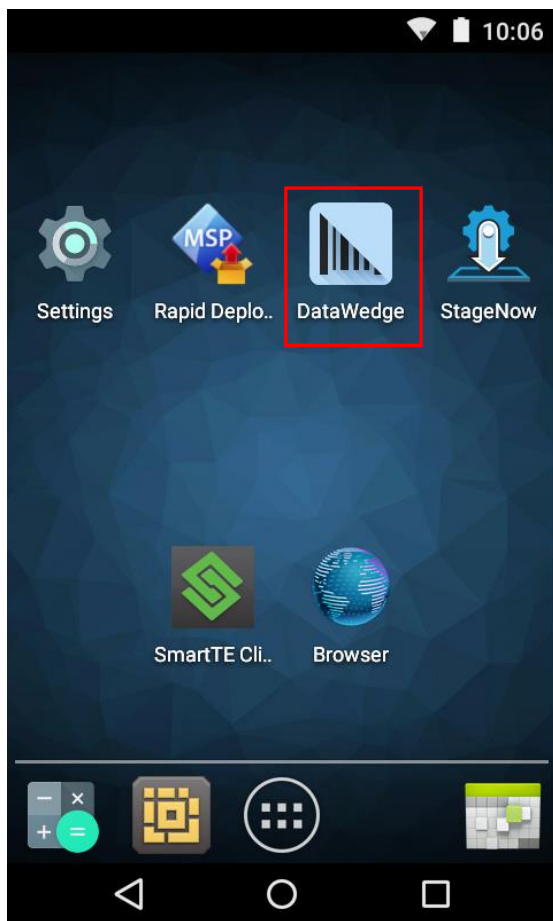
7.1 Creating a DataWedge Profile Manually



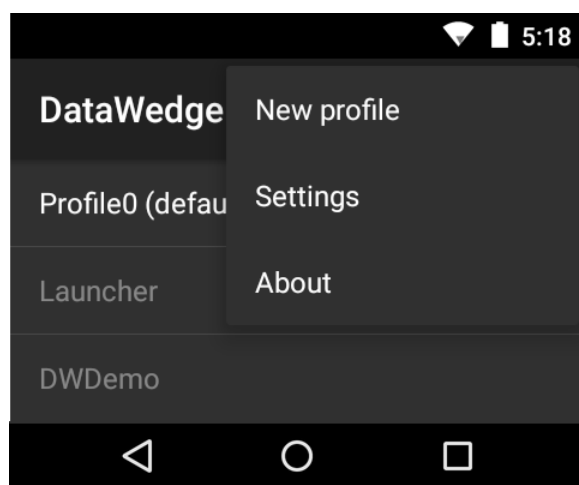
The following instructions include a name and reference to the Motorola ET1. This name must be used in each of the instructions below, regardless of the device model.

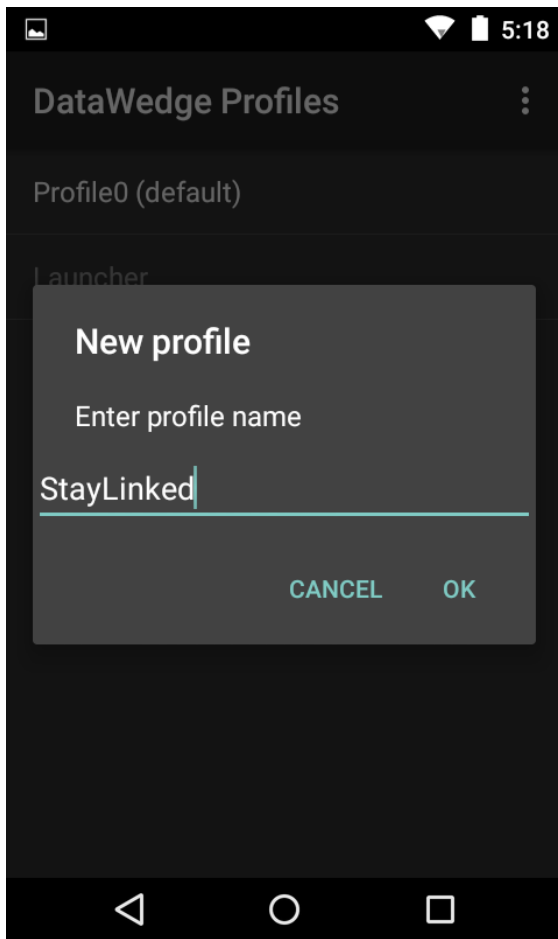
Start by pressing the application menu button on your device to access the installed applications. This is typically a center icon, displayed as a circle with six dots as enclosed in red to the left.

From the Apps list, select the DataWedge program. It may be necessary to slide the page left or right if several apps have been installed on your device.



Once DataWedge is open, you'll want to select the menu icon. This will allow you to create a new profile for use with Stay-Linked.



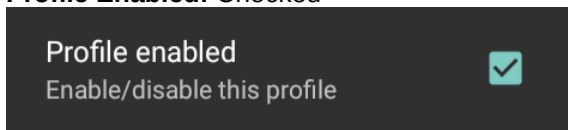


Once the new profile is created, tap and hold on it to get the menu of options for that profile. If you tap the profile name, it will automatically select 'edit profile' instead of presenting the menu.

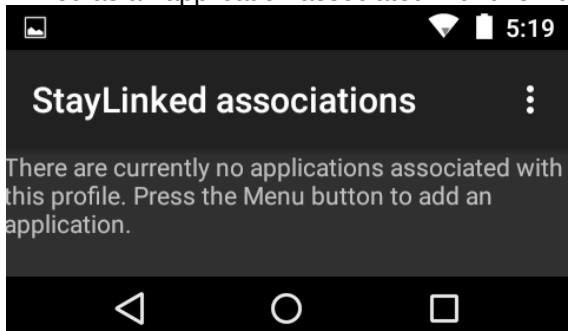
Several configuration options appear, but the screen can be scrolled down to cycle through the complete list. The Stay-Linked scanner profile should be configured with the following options:

The screenshots below have been adjusted to reduce the size of the images and document.

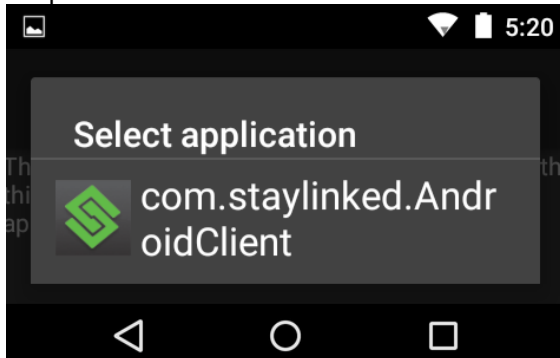
Profile Enabled: Checked



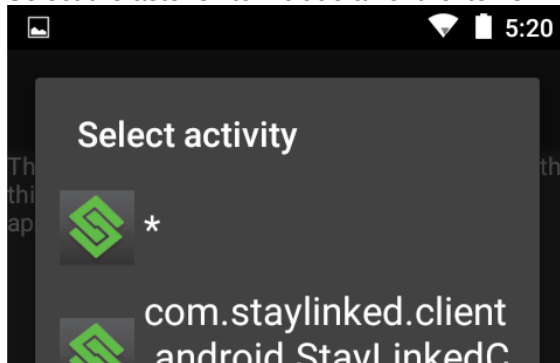
Associated Apps: There are multiple steps to configure this option. Select the menu button to add Stay-Linked as an application associated with this DataWedge profile.



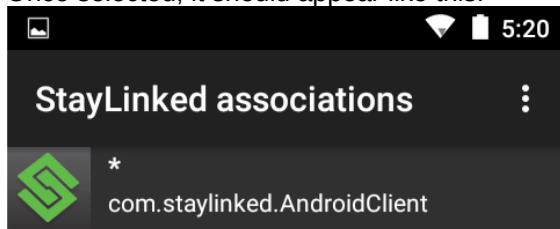
Select com.staylinked.AndroidClient as the application to associate with. The Stay-Linked Android client will appear near the end of the list, and will not appear if it has not yet been installed on your device. Swipe the screen down to scroll down the list.



Select the asterisk to include all of the items in the list.

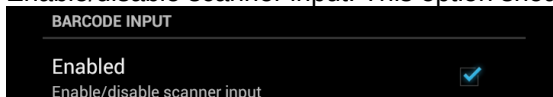


Once selected, it should appear like this:

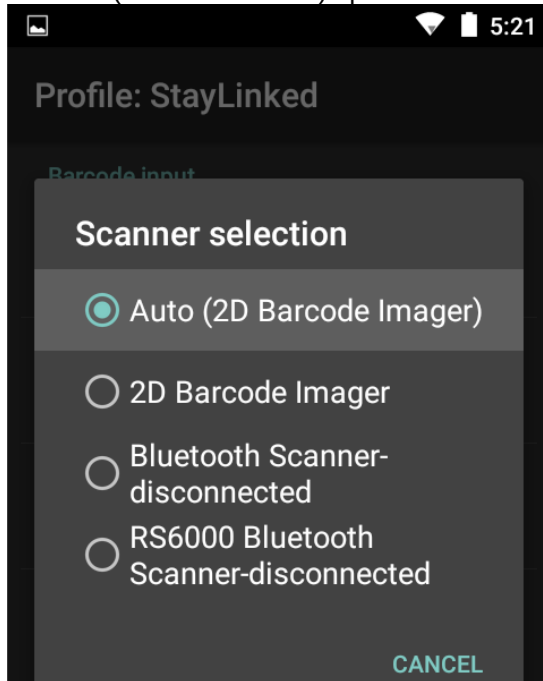


BARCODE INPUT

Enable/disable scanner input: This option should be checked by default.

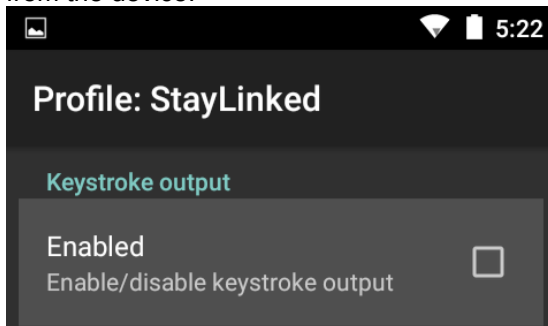


Scanner Selection: This setting will depend on your device configuration. Devices with an attached scanner should select the scanner. Our device had an integrated camera scanner that worked well with the Auto (Camera scanner) option selected.



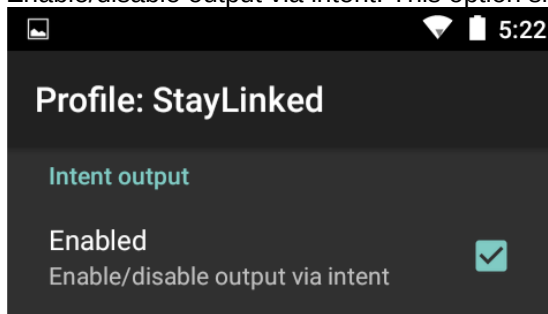
KEYSTROKE OUTPUT

Enable/disable keystroke output: This setting must be unchecked for Stay-Linked to receive scan input from the device.

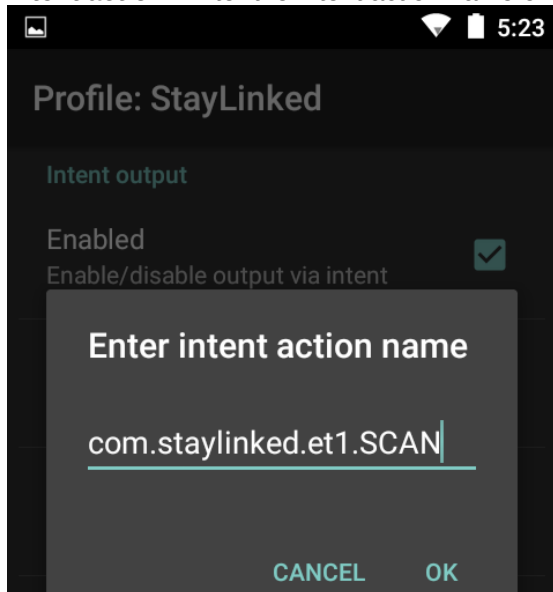


INTENT OUTPUT

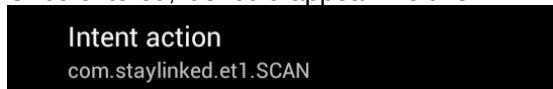
Enable/disable output via intent: This option should be checked.



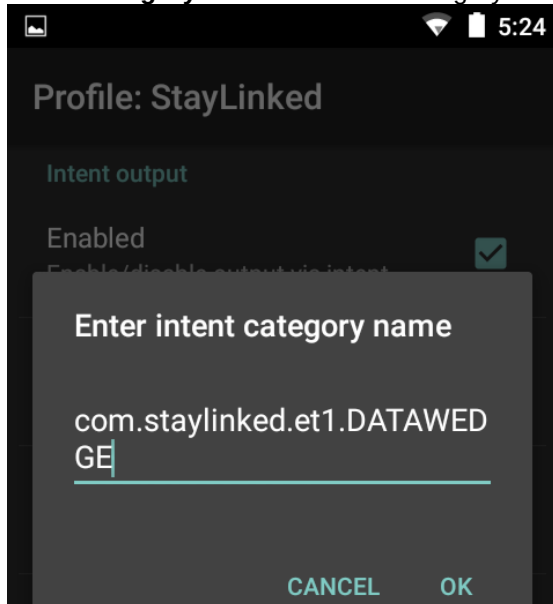
Intent action: Enter the intent action name of com.staylinked.et1.SCAN.



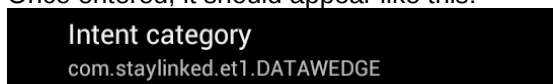
Once entered, it should appear like this:



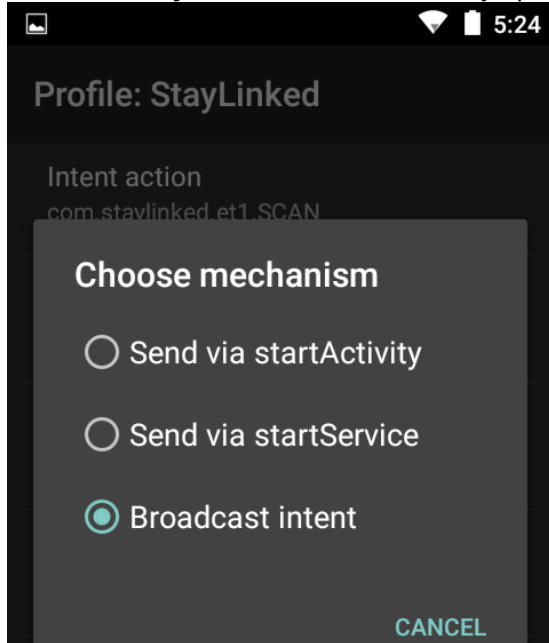
Intent category: Enter the intent category of com.staylinked.et1.DATAWEDGE.



Once entered, it should appear like this:



Intent delivery: Select the intent delivery option of Broadcast intent.



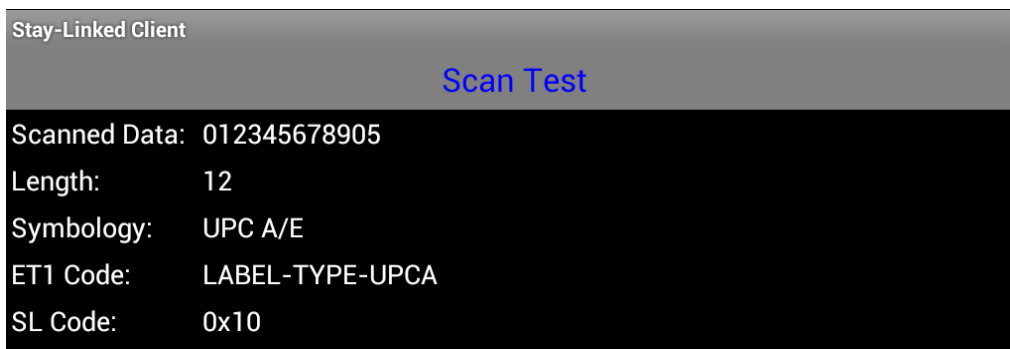
Once entered, it should appear like this:



Once completed, you can select the device Home or Back buttons to return to the device desktop.

7.2 Testing Intent Configuration

The Stay-Linked client includes a scan test under the Tools menu. After running the Stay-Linked client, select tools and scanner test. You should be able to scan nearly any barcode you can find, and data should be returned in the scan test.



Once your configuration is confirmed to work, the device should not need additional configuration for barcode adjustments. Barcode prefix and suffix, as well as many other options, can be configured centrally in the Stay-Linked Server under Device Groups or your Telnet Host entry.

7.3 Adjusting the Scan Engine

The DataWedge program contains all scan engine parameters. These features allow symbology options, such as disabling symbology types and converting barcodes to alternative symbologies. Adjustments should be made to the Stay-Linked profile in DataWedge for these hardware-level settings. These settings would be made in the Decoders, Decode Params, UCC EAN Params, Reader Params and Scan Params sections of the DataWedge profile.

The scanner.ini configuration options mentioned in this document do not apply to Android operating system devices.

7.4 Importing DataWedge Profiles

Importing Intents is accomplished from the Main DataWedge profiles page. Select the Menu option, then Settings > Export or Import. Different devices may have slightly different options. Download our sample copy which is an attachment to this PDF, or export your own if you want to import the profile on future devices.

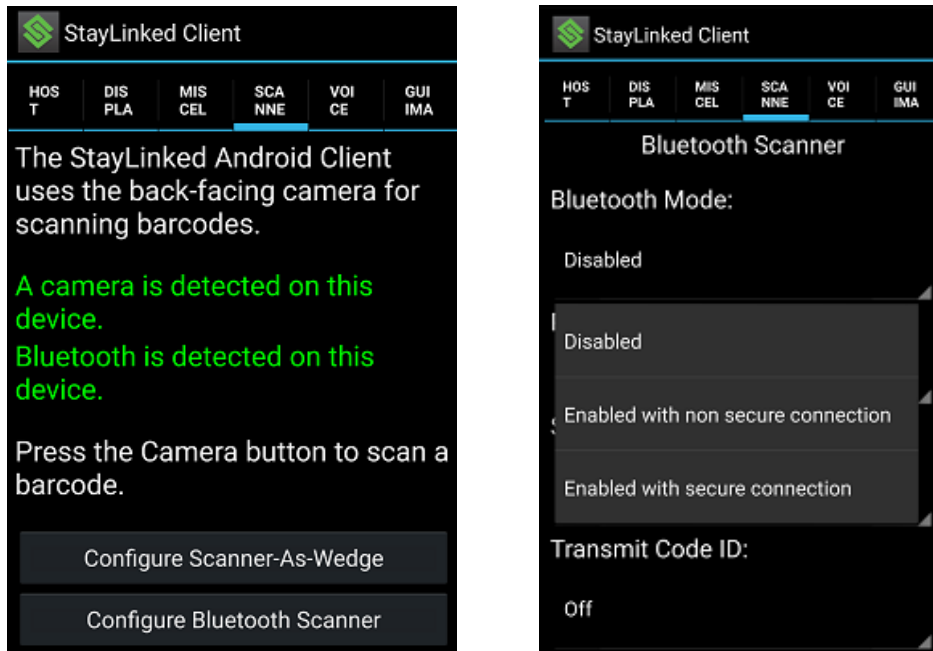
Note that the attachments will not appear in most browser plugins, or if your PDF security settings block attachments.

8 SPP Bluetooth Scanners

All Android-based devices can scan using the Bluetooth Serial Port Profile (SPP) interface. Selecting the Client menu options:

Host > Configure > Scanner > Configure Bluetooth Scanner

Note - The Host > Configure option will not be available if your device is currently displaying an emulation screen or error message. Returning to the Client splash screen will allow this option.



The options available in the Bluetooth configuration are similar to the scanner-as-wedge configuration dialogues on other devices:

Bluetooth Mode – Enabling Bluetooth mode provides an option for Bluetooth with or without security.

Paired Devices – Displays a list of paired devices, allowing you to select the active pair for use with StayLinked.

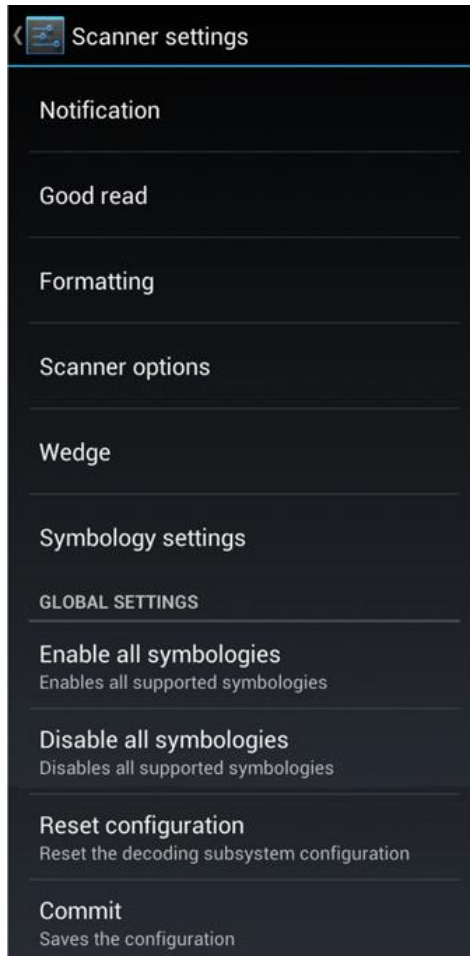
Scanner Type – This option sets the type of symbology IDs that will be included with your barcode scans. Selecting the wrong type of scanner may result in the wrong barcode types being reported by the scanner.

Transmit Code ID – Set this option to consume the first characters to identify the type of barcode that was scanned. Without configuring a symbology ID, the device will report all barcodes to the StayLinked Client as unknown/unsupported.

Devices using Bluetooth Human Input Device (HID) profiles are not supported for native scanner control and require traditional Scanner-As-Wedge configuration as mentioned above.

9 Android Devices by DataLogic

DataLogic devices use intents much the same as on Zebra/Motorola devices. The configuration can be found in the Settings > Scanner Settings options. After pressing the menu option to see all applications, select Settings > Scanner Settings > Wedge.



Select the following options in the Wedge configuration panel:

Enable keyboard wedge – Unchecked

Enable intent wedge – Checked

Intent delivery mode – Broadcast intent

Intent action –

com.datalogic.decodedwedge.decode_action

Intent category –

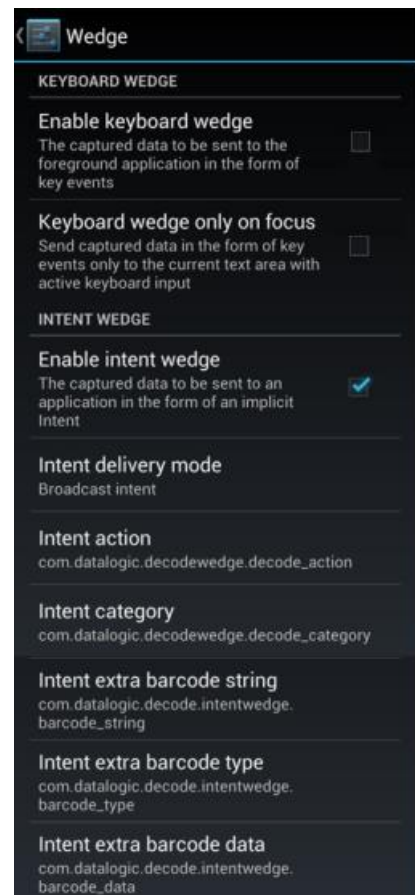
com.datalogic.decodedwedge.decode_category

Intent extra barcode string –

com.datalogic.decode.intentwedge.barcode_string

Intent extra barcode type –

com.datalogic.decode.intentwedge.barcode_type



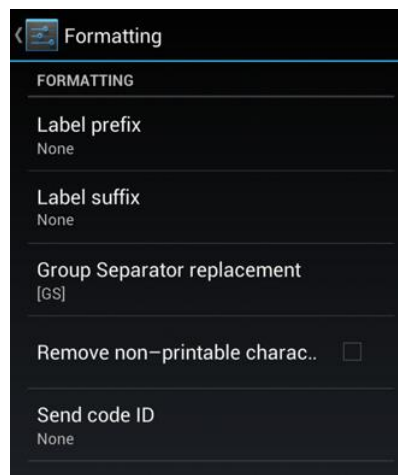
The 'Settings > Scanner Settings > Formatting' must be configured as follows:

Label prefix – None

Label suffix – None

The StayLinked server will apply any prefix and suffix options.

Once configuration is complete, you can test the results using the StayLinked Client scan test to see the barcode data and



symbology information.

10 Appendix – Symbol/Motorola Scanner.ini

The following file represents a typical sample of the Symbol/Motorola scanner configuration file. Different versions of the StayLinked Client may support different configuration options. Be sure to check the latest client user guide for the features support in your installation.

The following symbologies are enabled by default:

UPCE0, UPCA, EAN8, EAN13, CODABAR, CODE39, I2OF5, CODE128

The following symbologies must be explicitly enabled:

UPCE1, MSI, D2OF5, CODE11, CODE93, TRIOPTIC39, POINTER, IMAGE, SIGNATURE, DOCCAP, CUECODE, COMPOSITE_AB, COMPOSITE_C, TLC39

```
[defaults]
scan_reader_type="laser"      ; "laser", "imager"
scan_reader_port="SCN1:"      ; "SCN1:" - COM1
; "SCN2:" - COM2/BLUETOOTH
; "SCN3:" - USB
scan_adaptive_mode=0          ; 0 - ADAPTIVE_SCANNING_DISABLE
; 1 - ADAPTIVE_SCANNING_ENABLE
scan_aim_type=0               ; 0 - AIM_TYPE_TRIGGER
; 1 - AIM_TYPE_TIMED_HOLD
; 2 - AIM_TYPE_TIMED_RELEASE
; 3 - AIM_TYPE_PRESS_AND_RELEASE
; 4 - AIM_TYPE_PRESENTATION (MK500 only)
; 5 - AIM_TYPE_CONTINUOUS_READ
scan_aim_mode=0               ; 0 - AIM_MODE_NONE
; 1 - AIM_MODE_DOT
; 2 - AIM_MODE_SLAB
; 3 - AIM_MODE_RETICLE
scan_aim_duration=0           ; 0-5000 ms.
scan_raster_mode=0            ; 0 - RASTER_MODE_NONE
; 1 - RASTER_MODE_OPEN_ALWAYS
; 2 - RASTER_MODE_SMART
; 3 - RASTER_MODE_CYCLONE
scan_beam_width=0             ; 0 - BEAM_WIDTH_NORMAL
; 1 - BEAM_WIDTH_NARROW
; 2 - BEAM_WIDTH_WIDE
scan_laser_time=5000           ; 0-5000 ms.
scan_led_time=1000            ; 0-5000 ms.
scan_beep_time=200            ; 0-5000 ms.
scan_bidirectional=0          ; 0 - bidirectional redundancy off
; 1 - bidirectional redundancy on
scan_security_level=0         ; 0 - SECURITY_REDUNDANCY_AND_LENGTH,
; 1 - SECURITY_SHORT_OR_CODABAR,
; 2 - SECURITY_ALL_TWICE,
; 3 - SECURITY_LONG_AND_SHORT,
; 4 - SECURITY_ALL_THRICE,
scan_frequency=2670           ; 2500-3500
scan_ean_supps=2              ; 0 - SUPPLEMENTALS_NONE
; 1 - SUPPLEMENTALS_ALWAYS
; 2 - SUPPLEMENTALS_AUTO
scan_wave_file=""

[Symbologies]
#name=enable, parameter 1, through parameter 10 (comma delimited)
UPCE0=1,1,0,0,0,0,0,0,0,0,0
UPCE1=1,1,0,0,0,0,0,0,0,0,0
UPCA=1,1,1,0,0,0,0,0,0,0,0
MSI=0,1,0,0,0,0,0,0,0,0,0
EAN8=1,0,0,0,0,0,0,0,0,0,0
EAN13=1,0,0,0,0,0,0,0,0,0,0
CODABAR=1,1,0,0,0,0,0,0,0,0,0
CODE39=1,0,1,0,0,0,0,0,0,0,0
D25=1,0,0,0,0,0,0,0,0,0,0
I25=1,1,0,0,0,0,0,0,0,0,0
M25=1,0,0,0,0,0,0,0,0,0,0
CODE11=1,1,1,0,0,0,0,0,0,0,0
CODE93=1,0,0,0,0,0,0,0,0,0,0
CODE128=1,0,1,1,1,0,0,1,0,0,0
TRIOPTIC39=0,1,0,0,0,0,0,0,0,0,0
```

Copy this into notepad for a base scanner.ini configuration or rename the appropriate attached file.

Parameters for each symbology are defined as follows:

Symbology	PARAM1	PARAM2	PARAM3	PARAM4	PARAM5	PARAM6	PARAM7	PARAM8	PARAM9
UPCE0	Report Check Digit	Preamble	Convert To UPCA						
UPCE1	Report Check Digit	Preamble	Convert To UPCA						
UPCA	Report Check Digit	Preamble							
MSI	Redundancy	Check Digits	Check Digit Scheme	Report Check Digit	Min Length	Max Length			
EAN8	Convert To EAN13								
CODABAR	Redundancy	Clsi Editing	Not-is Editing	Min Length	Max Length				
CODE39	Verify Check Digit	Report Check Digit	Concatenation	Full ASCII	Redundancy	Convert To Code32	Code32 Prefix	Min Length	Max Length
D25	Redundancy	Min Length	Max Length						
I25	Redundancy	Verify Check Digit	Report Check Digit	Convert To EAN13	Min Length	Max Length			
CODE11	Redundancy	Check Digit Count	Report Check Digit	Min Length	Max Length				
CODE93	Redundancy	Min Length	Max Length						
CODE128	Redundancy	EAN128	ISBT128	Other128	ISBT128 Concatenate Mode	Check ISBT Table	Security Level	Min Length	Max Length
TRIOPTIC39	Redundancy								

10.1.1 UPCE0 Options

This symbology has options identical to the options in UPCE1 below.

10.1.2 UPCE1 Options

Report Check Digit - Defines if the engine will return the check digit as part of the data string after a successful decode. If 1, the engine returns the check digit. If 0, the check digit is not returned. The default value is 0.

Preamble - Defines if the engine will return the numeric system digit of the UPC label. If 1, the engine returns the number system digit. If 0, the number system digit is not returned. The default value is 0.

Convert To UPCA - Defines if the engine will expand UPC-E codes to the 12 digit UPC-A format after a successful decode. If 1, the engine expands the code. If 0, the engine does not expand the UPC-E code. The default value is 0.

10.1.3 UPCA Options

Report Check Digit - Defines if the engine will return the check digit as part of the data string after a successful decode. If 1, the engine returns the check digit. If 0, the check digit is not returned. The default value is 0.

Preamble - Defines if the engine will return the numeric system digit of the UPC label. If 1, the engine returns the number system digit. If 0, the number system digit is not returned. The default value is 1.

10.1.4 MSI Options

Redundancy - The minimum length decoded MSI message the engine should return. MSI messages smaller than this minimum length are not reported by the engine. The minimum allowable value (as well as the default) is 4.

Check Digits - The maximum length decoded MSI message the engine should return. MSI messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 48.

Check Digit Scheme - Defines if the engine will return the check character as part of the data string after a successful decode. If 1, the engine returns the check character. If 0, the check character is not returned. The default value is 0.

Report Check Digit - Defines if the engine will return the check character as part of the data string after a successful decode. If 1, the engine returns the check character. If 0, the check character is not returned. The default value is 0.

Min Length - The minimum length decoded Codabar message the engine should return. Codabar messages smaller than this minimum length are not reported by the engine. The minimum allowable value (as well as the default) is 2.

Max Length - The maximum length decoded Codabar message the engine should return. Codabar messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 60.

10.1.5 EAN8 Options

Convert To EAN13 - Defines if the engine will return the check character as part of the data string after a successful decode. If 1, the engine returns the check character. If 0, the check character is not returned. The default value is 0.

10.1.6 EAN13 Options

This symbology is not currently adjustable.

10.1.7 CODABAR Options

Redundancy - Defines if the start and stop characters are returned in the data string after a successful Codabar decode. If SSXmit is 1, the start and stop characters are included. If 0, they are not included. The default value is 0.

Cls Editing - Defines if the engine will read Codabar barcodes with or without check characters. If 1, the engine only decodes Codabar barcodes with a check character. If 0, the decoder decodes codes with or without a check character. The default value is 0.

Not-is Editing - Defines if the engine will return the check character as part of the data string after a successful decode. If 1, the engine returns the check character. If 0 the check character is not returned. The default value is 0.

Min Length - The minimum length decoded Codabar message the engine should return. Codabar messages smaller than this minimum length are not reported by the engine. The minimum allowable value (as well as the default) is 2.

Max Length - The maximum length decoded Codabar message the engine should return. Codabar messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 60.

10.1.8 CODE39 Options

Verify Check Digit - Defines if the start and stop characters are returned in the data string after a successful Code 39 decode. If SSXmit is 1, the start and stop characters are included. If 0, they are not included. The default value is 0.

Report Check Digit - Defines if the engine will read Code 39 barcodes with or without check characters. If 1, the engine only decodes Code 39 codes with a check character. If 0, the decoder decodes codes with or without a check character. The default value is 0.

Concatenation - Defines if the engine will return the check character as part of the data string after a successful decode. If 1, the engine returns the check character. If 0, the check character is not returned. The default value is 0. *Note: this parameter is only used when CheckCharOn is set to 1. If CheckCharOn is set to 0, this parameter is ignored.*

Full ASCII - Defines if certain character pairs within the barcode symbol are interpreted and returned as a single character. If Full ASCII is 1, interpretation is enabled. If 0, no interpretation is attempted. The default value is 0.

Redundancy - This parameter not supported - must be set to 0.

Min Length - The minimum length decoded Code 39 message the engine should return. Code 39 messages smaller than this minimum length are not reported by the engine. The default value is 2, and the minimum allowable value is 0.

Max Length - The maximum length decoded Code 39 message the engine should return. Code 39 messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 48.

10.1.9 D25 Options

Redundancy - Defines if the start and stop characters are returned in the data string after a successful Codabar decode. If SSXmit is 1, the start and stop characters are included. If 0, they are not included. The default value is 0.

Min Length - The minimum length decoded message the engine should return. Messages smaller than this minimum length are not reported by the engine. The minimum allowable value (as well as the default) is 4.

Max Length - The maximum length decoded message the engine should return. Messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 80.

10.1.10 I25 Options

Redundancy - Defines if the engine will read Interleaved 2 of 5 barcodes with or without check characters. If 1, the engine only decodes Interleaved 2 of 5 codes with a check digit. If 0, the decoder decodes codes with or without a check digit. The default value is 0.

Min Length - The minimum length decoded Interleaved 2 of 5 message the engine should return. Interleaved 2 of 5 messages smaller than this minimum length are not reported by the engine. The default value is 6, and the minimum allowable value is 4.

Max Length - The maximum length decoded Interleaved 2 of 5 message the engine should return. Interleaved 2 of 5 messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 80.

10.1.11 CODE11 Options

Redundancy - If 1, the engine only decodes Code 11 barcodes printed with two check digits. Otherwise, the engine decodes Code 11 barcodes as if they were printed with only one check digit. The default value is 1.

Check Digit Count - If 1, the engine only decodes Code 11 barcodes printed with two check digits. Otherwise, the engine decodes Code 11 barcodes as if they were printed with only one check digit. The default value is 1.

Report Check Digit - If 1, the engine only decodes Code 11 barcodes printed with two check digits. Otherwise, the engine decodes Code 11 barcodes as if they were printed with only one check digit. The default value is 1.

Min Length - The minimum length decoded Code 11 message the engine should return. Code 11 messages smaller than this minimum length are not reported by the engine. The default value is 4, and the minimum allowable value is 1.

Max Length - The maximum length decoded Code 11 message the engine should return. Code 11 messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 80.

10.1.12 CODE93 Options

Redundancy - Defines if the start and stop characters are returned in the data string after a successful Codabar decode. If SSXmit is 1, the start and stop characters are included. If 0, they are not included. The default value is 0.

Min Length - The minimum length decoded Code 93 message the engine should return. Code 93 messages smaller than this minimum length are not reported by the engine. The minimum allowable value (as well as the default) is 0.

Max Length - The maximum length decoded Code 93 message the engine should return. Code 93 messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 80.

10.1.13 CODE128 Options

Redundancy - Flag to enable redundancy. If this flag is set, the bar code must be decoded twice before being accepted.

Note: This flag is only considered if scan_security_level=0.

EAN128 - Flag to enable EAN128 subtype.

ISBT128 - Flag to enable ISBT128 subtype.

Other128 - Flag to enable other (non EAN or ISBT) 128 subtype.

ISBT128 Concatenate Mode - The ISBT128 concatenation feature allows a pair of barcodes which meet certain criteria defined in the ISBT128 spec to be reported as a single barcode. This parameter describes the different concatenation modes available for ISBT128. Possible values are:

- 0 – (ISBT128_CONCAT_MODE_NEVER) Will ignore the barcode pair and only output decode data for only one of the barcodes.
- 1 – (ISBT128_CONCAT_MODE_ALWAYS) Will not decode if both the barcodes are not present or if one of them cannot be decoded.
- 2 – (ISBT128_CONCAT_MODE_AUTO) Auto-discriminate.

Check ISBT Table - Will decode concatenated output only if the pair belongs to one of the commonly concatenated pair as defined by the standard.

Security Level - Describes the CODE 128 security level. Possible values are:

- 0 – This setting allows the digital scanner to operate in its most aggressive state, while providing sufficient security in decoding most “in-spec” bar codes;
- 1 – This default setting eliminates most misdecodes;
- 2 – Select this option if Security level 1 fails to eliminate misdecodes;
- 3 – If you selected Security Level 2 and misdecodes still occur, select this security level.

Note: Be advised, selecting SECURITY_LEVEL3 is an extreme measure against mis-decoding severely out of spec bar codes. Selecting this level of security significantly impairs the decoding ability of the digital scanner. If you need this level of security, try to improve the quality of the bar codes

Min Length - The minimum length decoded message the engine should return. Messages smaller than this minimum length are not reported by the engine.

Max Length - The maximum length decoded message the engine should return. Messages larger than this maximum length are not reported by the engine.

10.1.14 TRIOPTIC39 Options

Redundancy - Defines if the start and stop characters are returned in the data string after a successful Codabar decode. If SSXmit is 1, the start and stop characters are included. If 0, they are not included. The default value is 0.

Min Length - The minimum length decoded message the engine should return. Messages smaller than this minimum length are not reported by the engine. The minimum allowable value (as well as the default) is 4.

Max Length - The maximum length decoded message the engine should return. Messages larger than this maximum length are not reported by the engine. The maximum allowable value (as well as the default) is 48.

10.1.15 RSS14 Options

This symbology is not currently adjustable.

11 Contact Information

The features are included as part of the StayLinked solution. Many of these features require specific Server, Administrator and/or Client versions for access to the complete feature set. With proper configuration, these tools create many new options for improving the operation and presentation of emulation to your audience.

For further information or technical assistance, please contact your StayLinked Certified Partner or our technical support team.



13700 Alton Parkway
Suite 160
Irvine, CA 92618

714.918.7817 Support
714.918.7700 Main
support@staylinked.com

<http://www.StayLinked.com/>