PROVIDER FOR BIZTALK RFID MOBILE

DEVELOPER GUIDE

PROVIDER FOR BIZTALK RFID MOBILE DEVELOPER GUIDE

72E-167372-02

Revision A

April 2015

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

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

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

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

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

Warranty

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

Revision History

Changes to the original guide are listed below:

Change	Date	Description
-01 Rev A	8/2012	Initial release
-02 Rev A	4/2015	Zebra Re-Branding

TABLE OF CONTENTS

About This Guide

Chapter 1: Getting Started

Chapter 2: Configuration

Chapter 3: Basic Operations

Chapter 4: Supported Commands

Index

ABOUT THIS GUIDE

Introduction

The *Zebra Provider for BizTalk RFID Mobile Developer Guide* provides software developers information on creating applications that use the Microsoft BizTalk RFID platform for the Zebra hand held MC3190Z and MC9090Z RFID Readers.

This guide assumes familiarity with the Microsoft BizTalk programming model and RFID documentation, as well as with Microsoft .NET, C#, and Visual Studio 2005 or above.

Chapter Descriptions

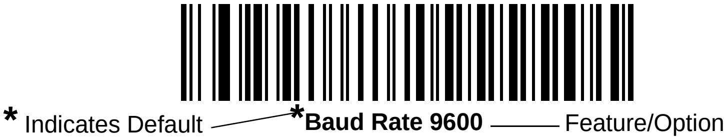
Topics covered in this guide are as follows:

- *Chapter 1, Getting Started* describes how to install Zebra Provider.
- *Chapter 2, Configuration* includes information on configuring Zebra Provider.
- *Chapter 3, Basic Operations* includes information on how to use the Zebra provider, including tag reading and device diagnostics.
- *Chapter 4, Supported Commands* includes information on the commands supported by the Zebra Provider, including general and Gen2 commands.

Notational Conventions

The following conventions are used in this document:

- *Italics* are used to highlight the following:
  - Chapters and sections in this and related documents
  - Dialog box, window and screen names
  - Drop-down list and list box names
  - Check box and radio button names
- **Bold** text is used to highlight the following:
  - Property and command names
  - Key names on a keypad
  - Button names on a screen.
- bullets (•) indicate:
  - Action items
  - Lists of alternatives
  - Lists of required steps that are not necessarily sequential
- Sequential lists (e.g., those that describe step-by-step procedures) appear as numbered lists.
- Throughout the programming bar code menus, asterisks (*) are used to denote default parameter settings.



Related Documents

Refer to the *Product Reference Guide* for the mobile RFID reader for product-specific information.

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

Service Information

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

When contacting Zebra support, please have the following information available:

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

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

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

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

CHAPTER 1 GETTING STARTED

Introduction

The Zebra DSPI Provider enables software developers to write applications using Microsoft's BizTalk RFID programming model for the Zebra hand held MC3190Z and MC9090Z RFID Readers. The provider supports all basic operations including connecting to the radio module, reading tags continuously, and reading tags on demand.

Installation

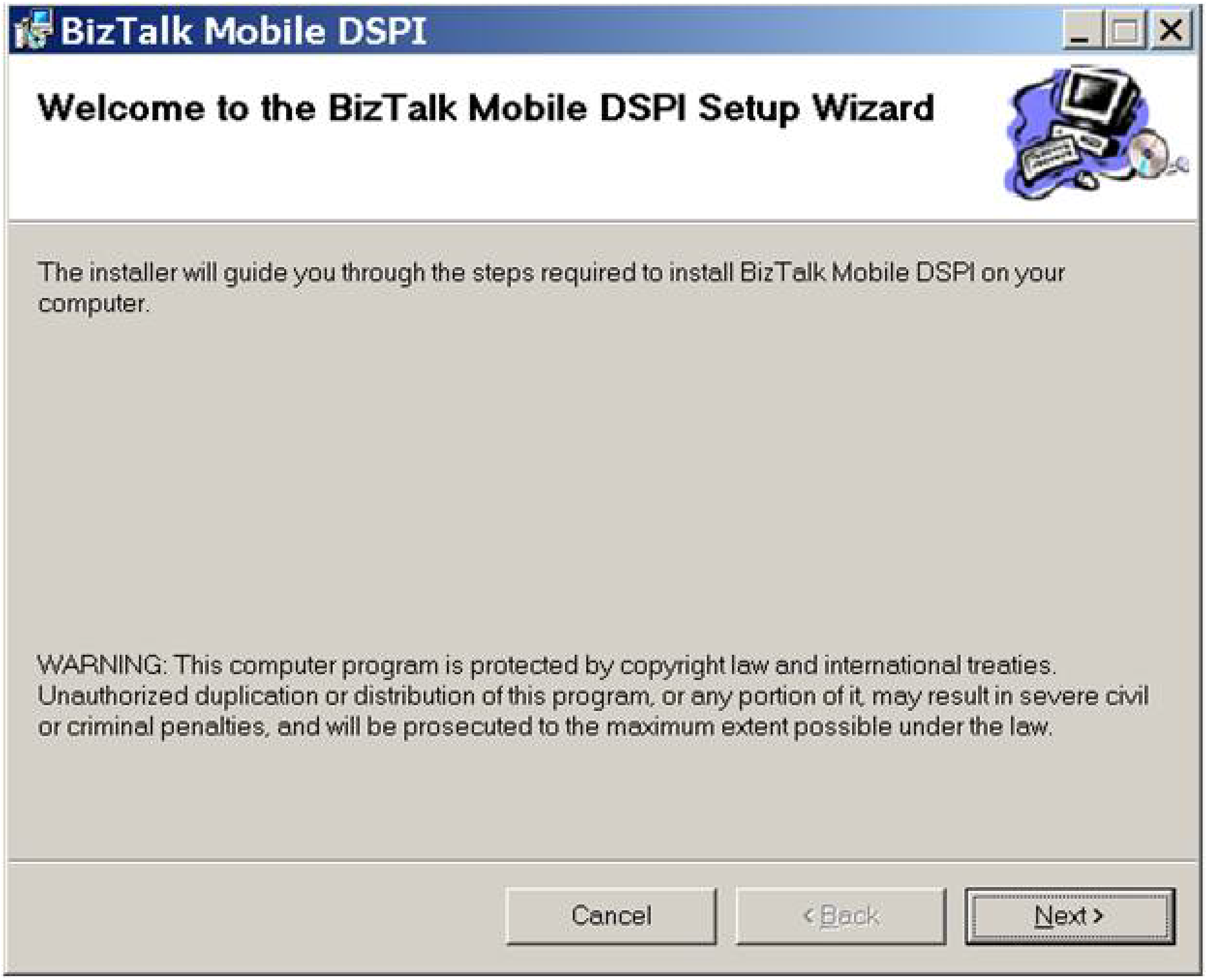
Requirements

- Windows Mobile 6.0 (and above)
- Microsoft BizTalk Mobile RFID

Install Zebra DSPI Provider to the hand held mobile device from the PC or to the device directly through .cab file.

Installing Zebra Provider From PC

1. Copy the **setup.exe** and the **MotorolaProviderInstaller.msi** to a local directory on the host PC. The file is available on Zebra support at http://www.zebra.com/support.
2. Double-click **setup.exe** to launch.
3. Run installer sequence and accept all defaults.



BizTalk Mobile DSPI

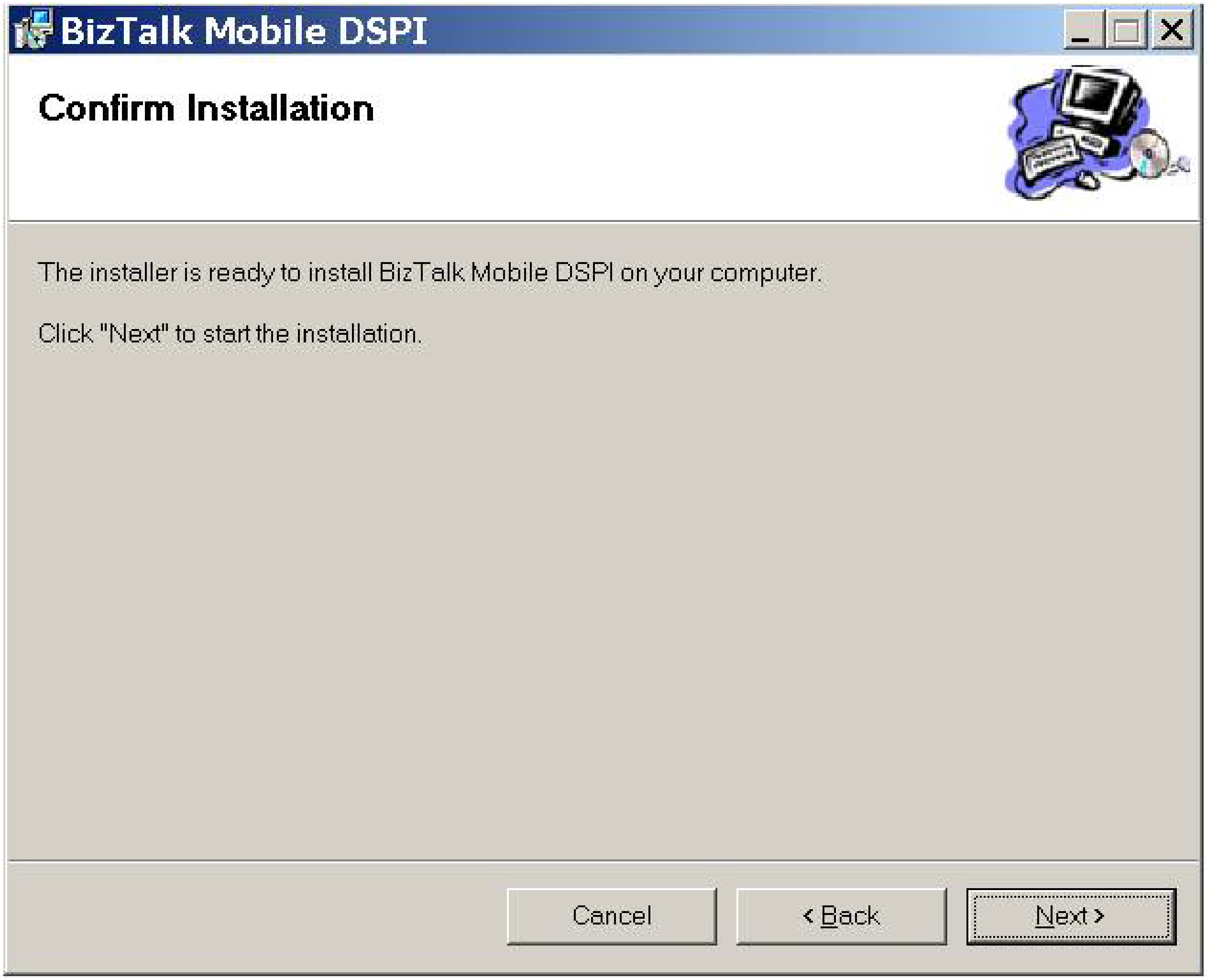
Welcome to the BizTalk Mobile DSPI Setup Wizard

The installer will guide you through the steps required to install BizTalk Mobile DSPI on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure 1-1 *Initial Installer Window*

4. .Select **Next** to start installation.

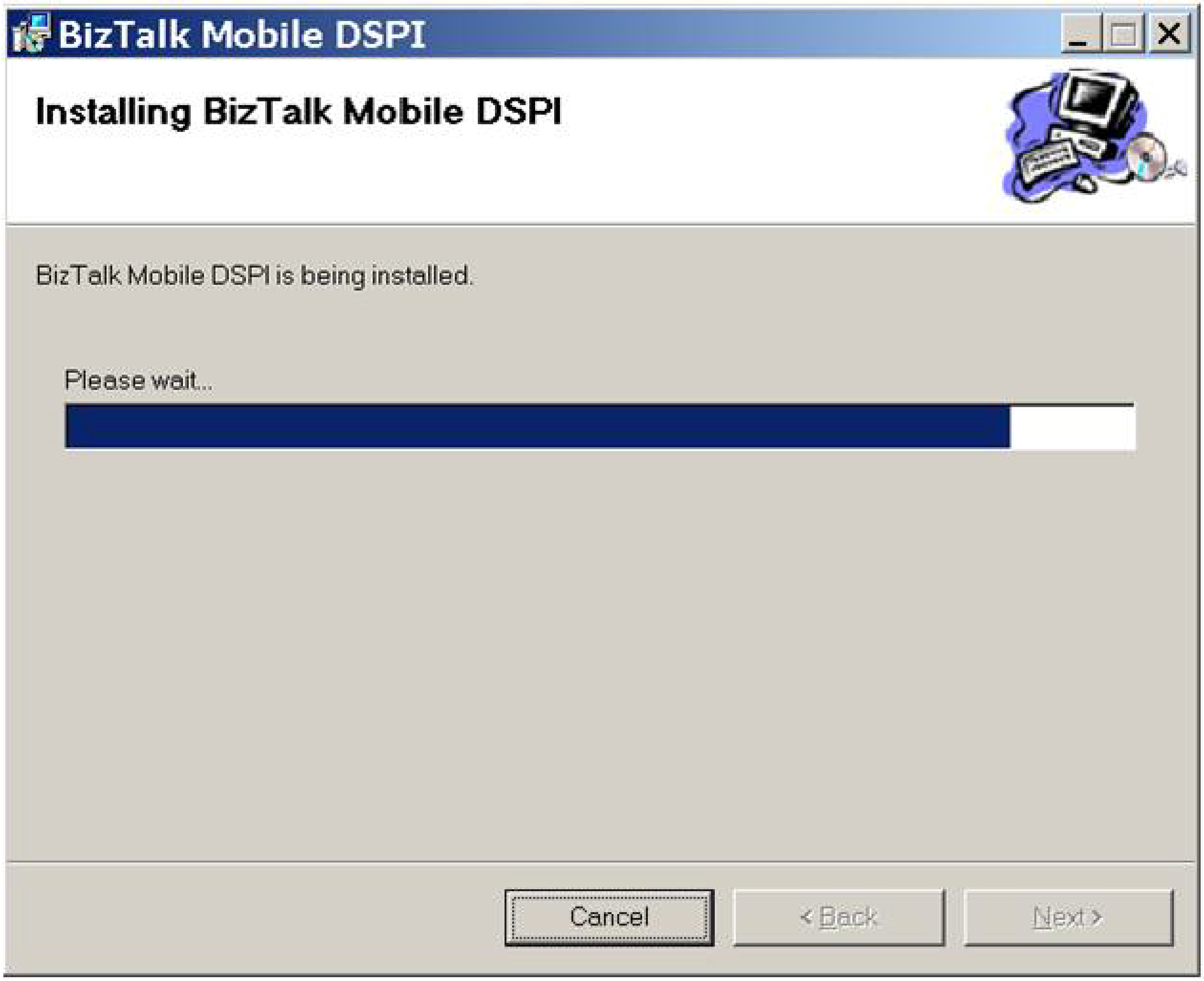


Confirm Installation

The installer is ready to install BizTalk Mobile DSPI on your computer.

Click "Next" to start the installation.

Figure 1-2 *Confirm Installation Window*



BizTalk Mobile DSPI

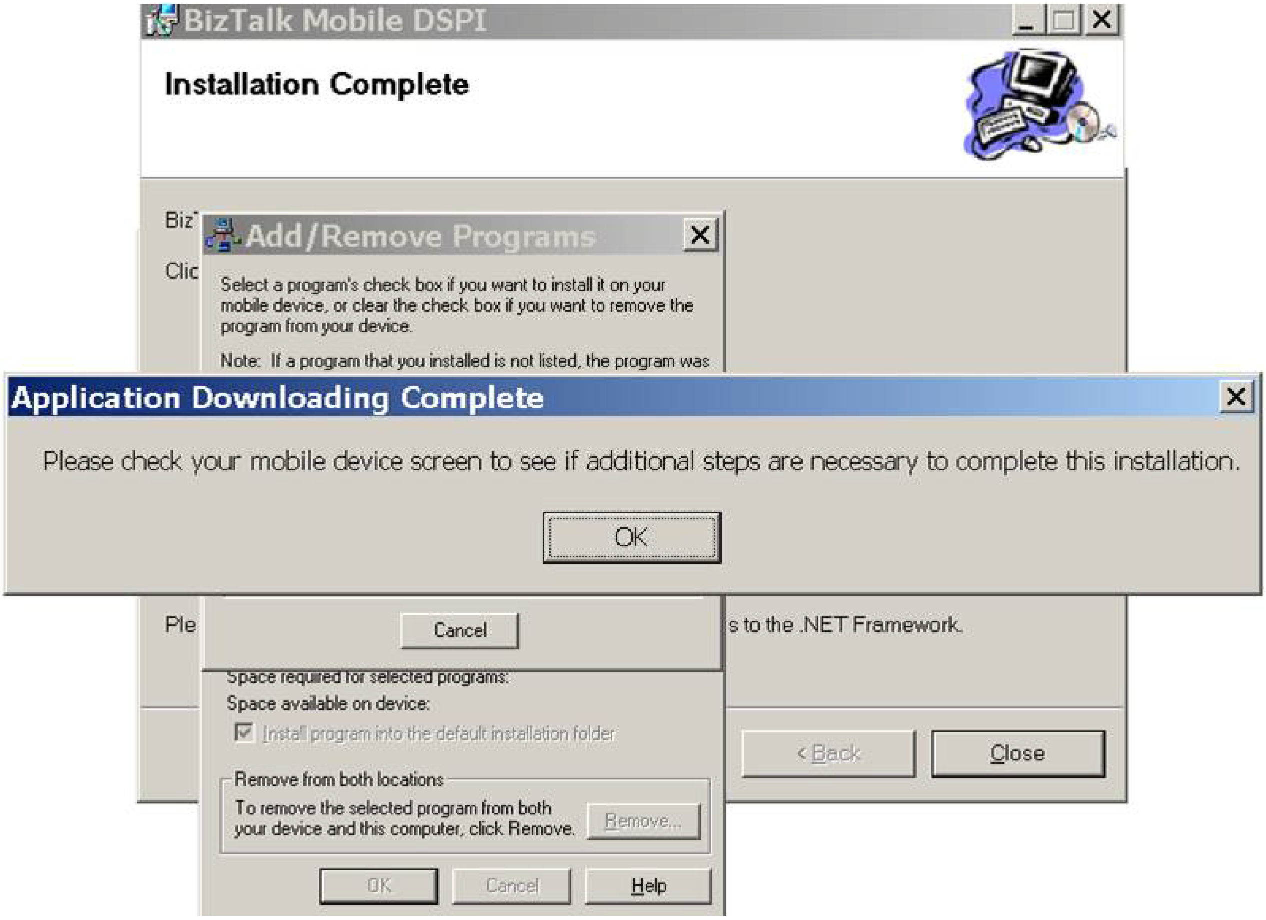
Installing BizTalk Mobile DSPI

BizTalk Mobile DSPI is being installed.

Please wait...

Figure 1-3 *Installing Window*

5. The Installer begins installing the cab onto the device automatically.



Select a program's check box if you want to install it on your mobile device, or clear the check box if you want to remove the program from your device.

Note: If a program that you installed is not listed, the program was

Please check your mobile device screen to see if additional steps are necessary to complete this installation.

s to the .NET Framework.

Space required for selected programs:

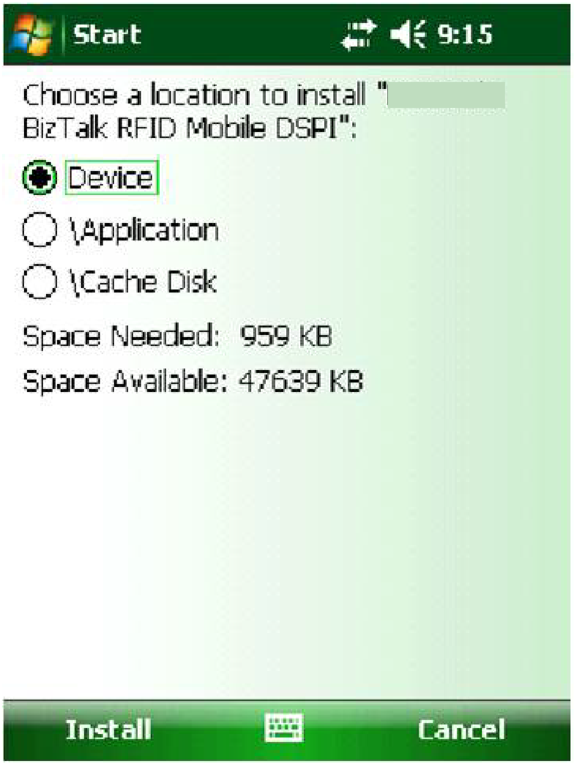
Space available on device:

☑ Install program into the default installation folder

To remove the selected program from both your device and this computer, click Remove.

Figure 1-4 *Downloading Complete Window*

6. Continue to device UI for completing installation.



Choose a location to install "[redacted] BizTalk RFID Mobile DSPI":

● Device

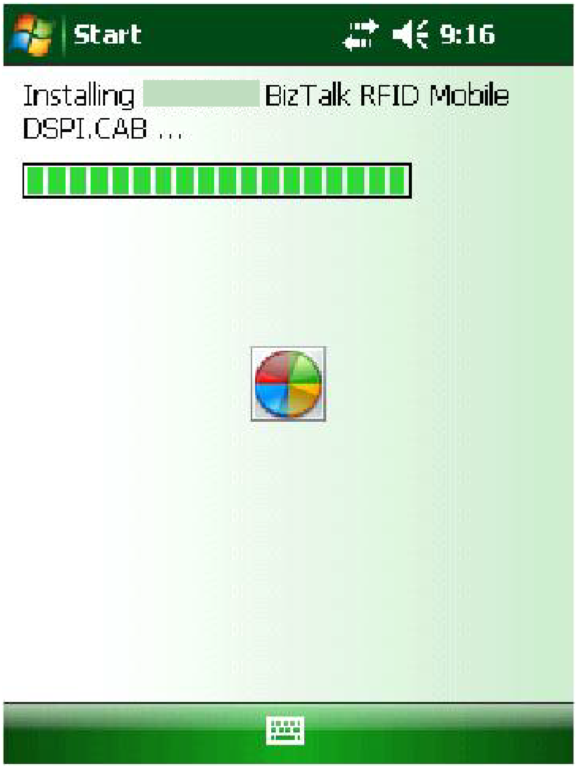
○ \Application

○ \Cache Disk

Space Needed: 959 KB

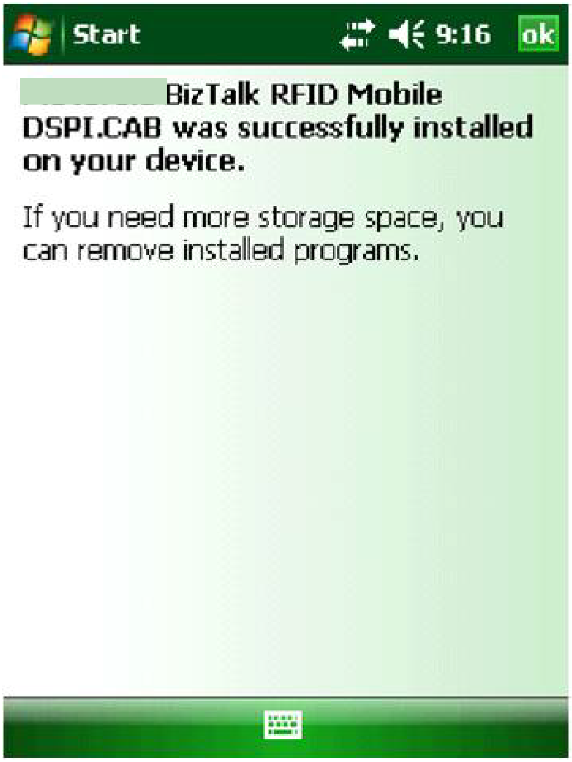
Space Available: 47639 KB

Figure 1-5 *Device Install Location Window*



Installing [redacted] BizTalk RFID Mobile DSPI.CAB ...

Figure 1-6 *Device Install Window*



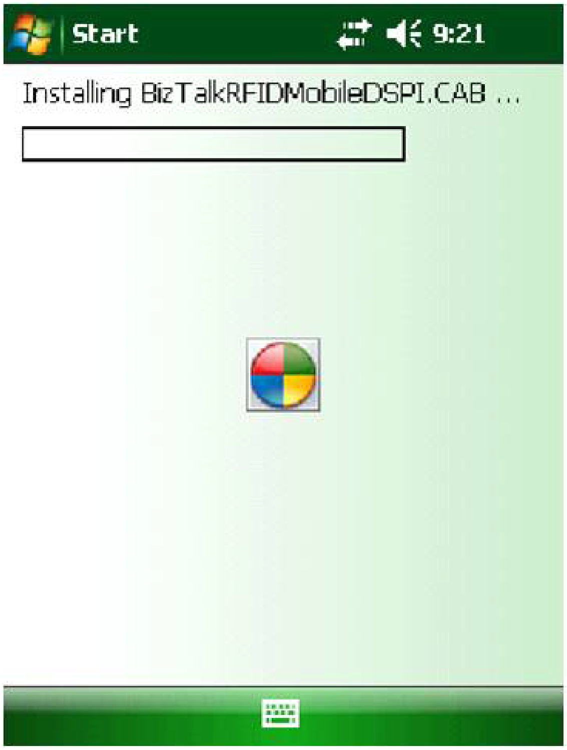
[redacted] BizTalk RFID Mobile DSPI.CAB was successfully installed on your device.

If you need more storage space, you can remove installed programs.

Figure 1-7 *Device Installation Confirmation Window*

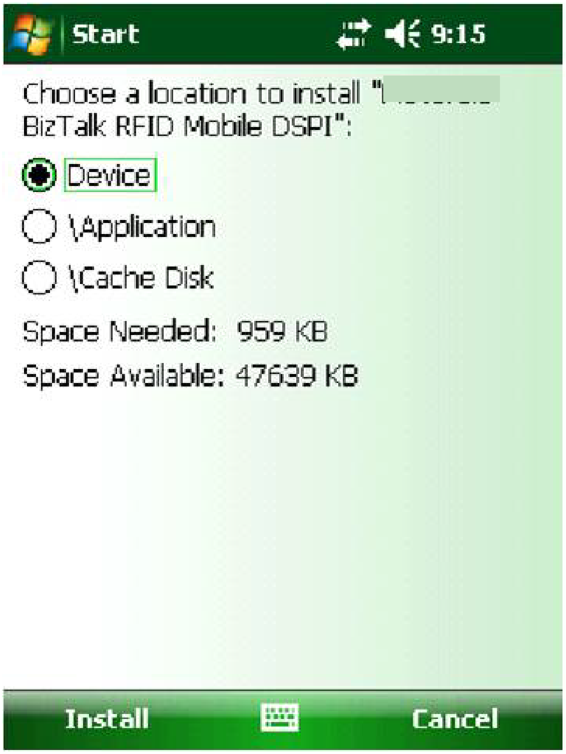
Installing Zebra Provider From Device Using .Cab File

1. Copy **BizTalkRFIDMobileDSPI.CAB** to temp folder of the device using active sync.
2. Double click the **BizTalkRFIDMobileDSPI.CAB** file to launch the installation.



Installing BizTalkRFIDMobileDSPI.CAB ...

Figure 1-8 *Device Install .CAB File Window*



Choose a location to install "[redacted] BizTalk RFID Mobile DSPI":

● Device

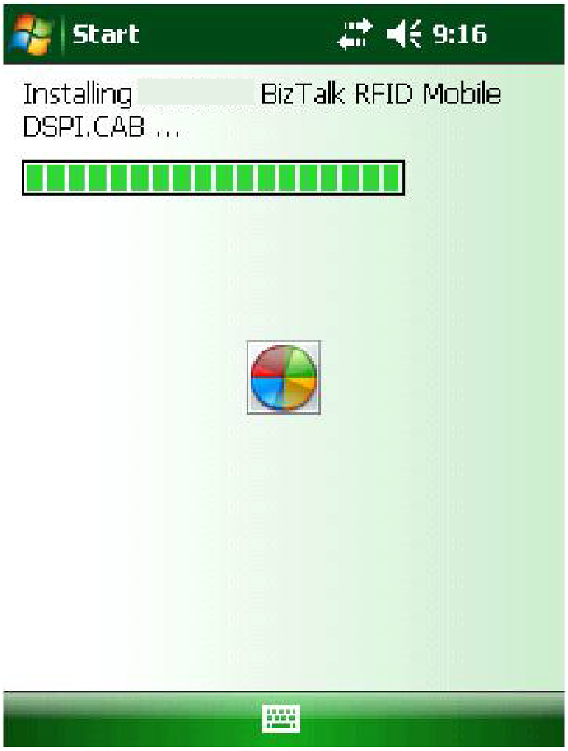
○ \Application

○ \Cache Disk

Space Needed: 959 KB

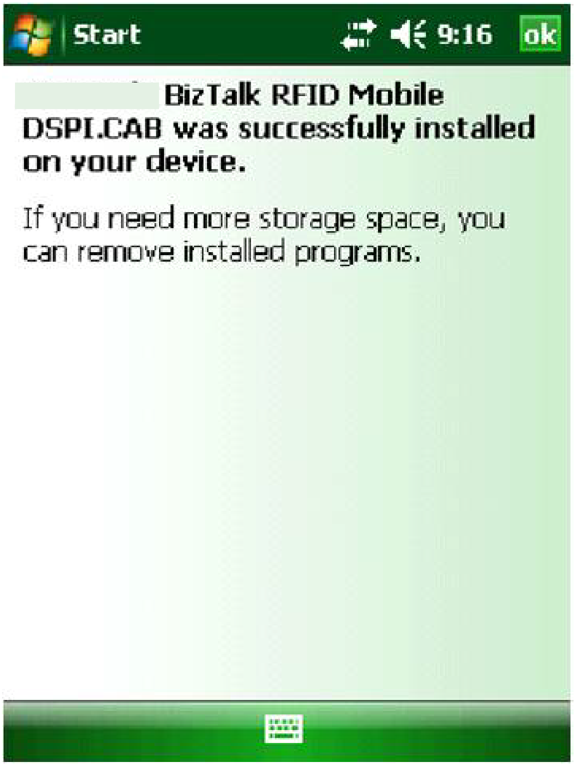
Space Available: 47639 KB

Figure 1-9 *Device Install .CAB File Location Window*



Installing [redacted] BizTalk RFID Mobile DSPI.CAB ...

Figure 1-10 *Device .CAB File Installation Window*



BizTalk RFID Mobile DSPI.CAB was successfully installed on your device.

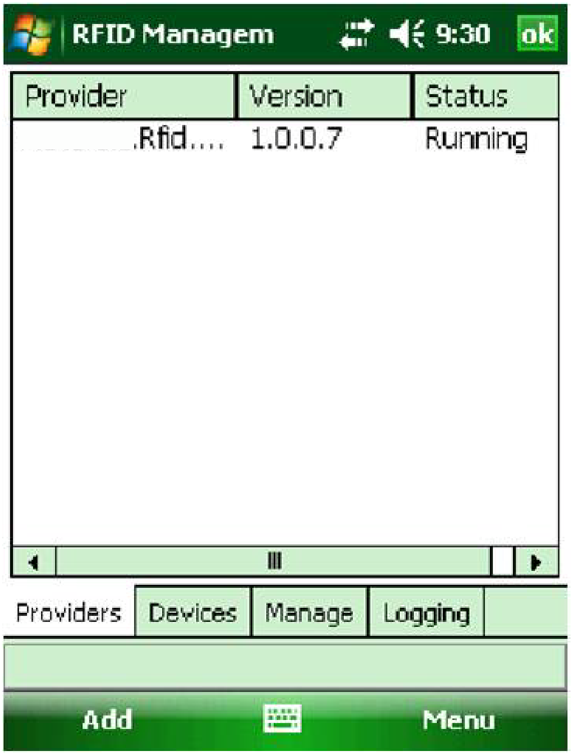
If you need more storage space, you can remove installed programs.

Figure 1-11 *Device .CAB File Confirmation Window*

Verification

To verify that the device provider installed correctly, perform the following tasks:

1. From Programs, launch RFID Manager.
2. Click on the **Providers** tab.
3. Confirm Zebra Provider is registered and starts successfully, as shown in *Figure 1-12*.



Provider	Version	Status
.Rfid....	1.0.0.7	Running

Figure 1-12 *Provider Launch Verification*

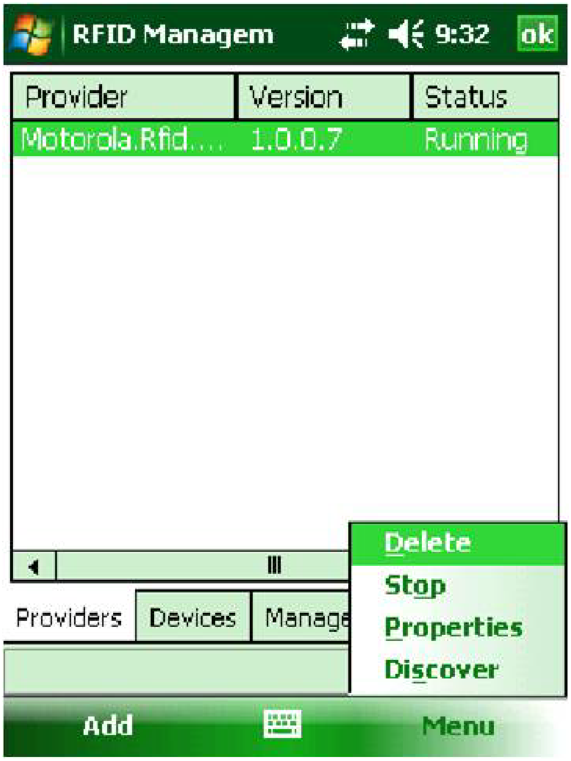
CHAPTER 2 CONFIGURATION

Introduction

This chapter includes information on configuring the Zebra provider.

Provider Properties

To see the provider properties, open RFID Manager, select the provider, and then go to **Menu** > **Properties**.

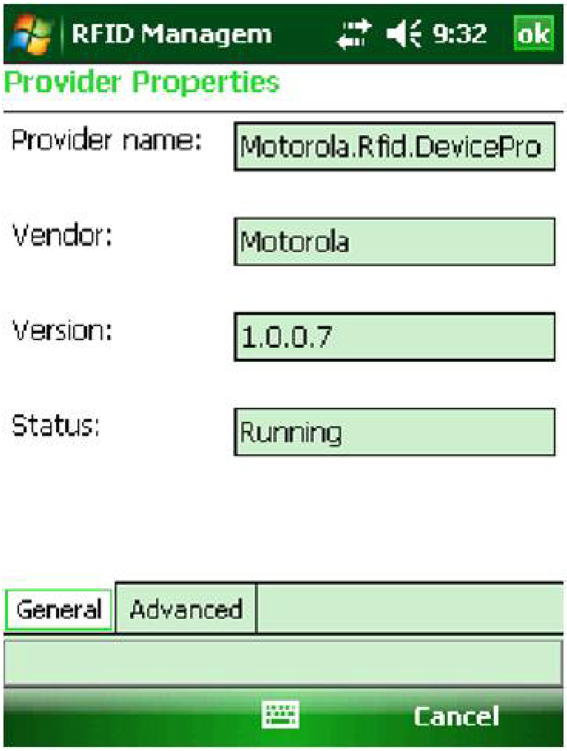


Provider	Version	Status
Motorola.Rfid....	1.0.0.7	Running

Figure 2-1 *Provider Menu*

The provider supports the following properties:

- General properties



Provider Properties

Provider name: Motorola.Rfid.DevicePro

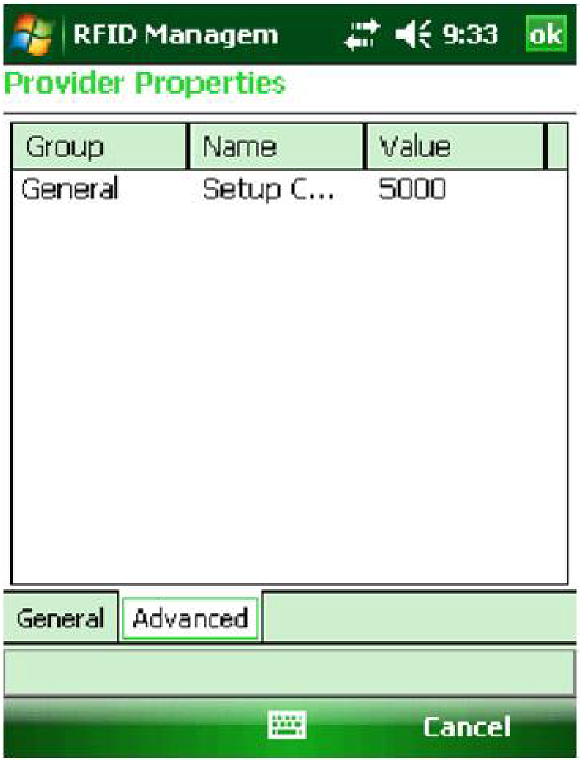
Vendor: Motorola

Version: 1.0.0.7

Status: Running

Figure 2-2 *General Properties Window*

- Advanced properties



Provider Properties

Group	Name	Value
General	Setup C...	5000

Figure 2-3 *Advanced Properties Window*

Property Descriptions

Discovery

The Zebra Mobile provider always returns local IP address for device discovery.

Setup Connection Timeout

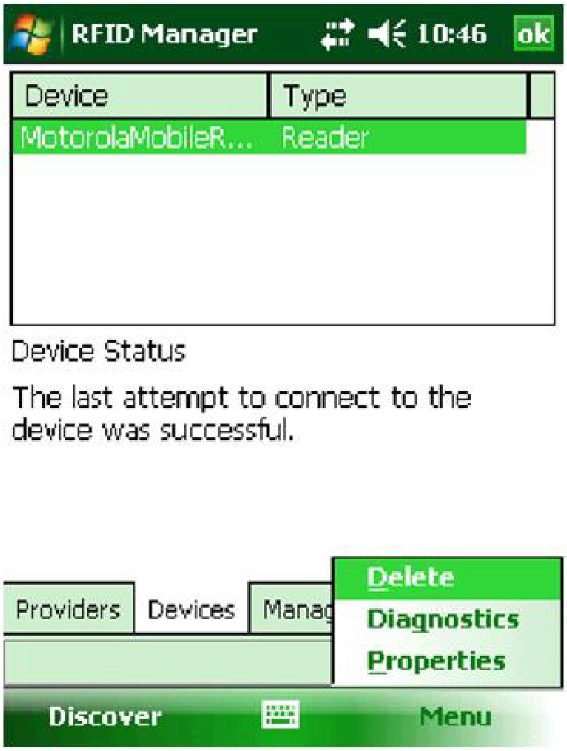
This is the time for which the provider will wait for a response from the device while setting up a connection to the device.

✓ ***NOTE*** *CheckFirmwareCompatibility*, *RebootDevice*, *RenameDevice*, and *UpdateDevice* methods are NOT supported in DeviceManagerProxy.

Device Properties

To see the device properties, open RFID Manager, select the devices tab, click on the device and **Menu > Properties.** These properties are typically not specific to each RF source.

.



Device	Type
MotorolaMobileR...	Reader

Device Status

The last attempt to connect to the device was successful.

Figure 2-4 *Device Properties Window*

General Properties

Name

This is a read-only property hardcoded to *ZebraMobileRFIDReader*.

Vendor

A read-only property hardcoded to Zebra Technologies Corporation.

Firmware Version

The version string points to Mobile RFID version. For example: *Firmware:2.2.4*.

Location

This is a read-only property hardcoded to *Inproc Device*.

Device ID

A read-only property, and corresponds to the MAC address/Serial number of the device.

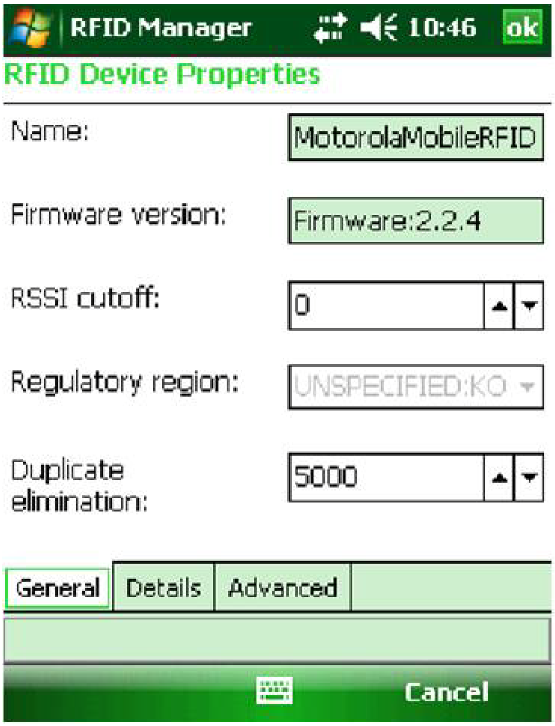
Regulatory Region

This is a read-only property. The reader capabilities of the RFID Reader object are used to form the Regulatory Region by concatenating the Communication Standard with the Country Name e.g. "ETSI_302_208 : UNITED KINGDOM"

Description

Read-only property, which provides model name in the description.

.



RFID Manager 10:46 ok

RFID Device Properties

Name: MotorolaMobileRFID

Firmware version: Firmware:2.2.4

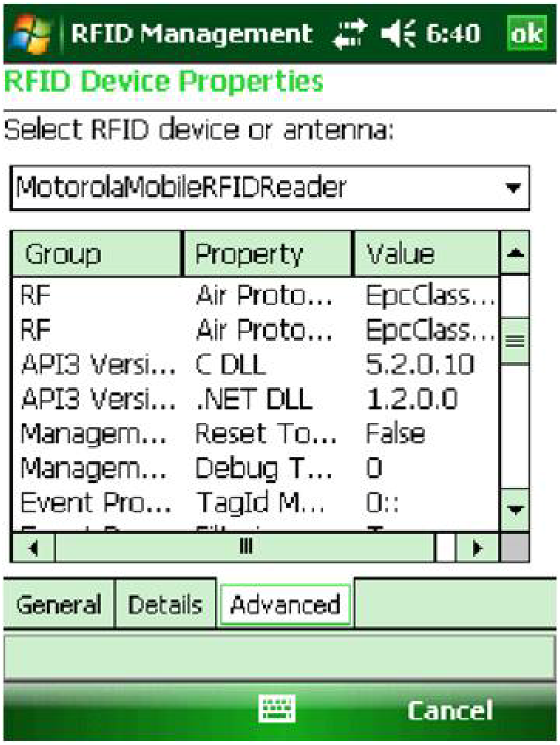
RSSI cutoff: 0

Regulatory region: UNSPECIFIED:KO

Duplicate elimination: 5000

Figure 2-5 *General Tab Window*

.



Select RFID device or antenna:

MotorolaMobileRFIDReader

Group	Property	Value
RF	Air Proto...	EpcClass...
RF	Air Proto...	EpcClass...
API3 Versi...	C DLL	5.2.0.10
API3 Versi...	.NET DLL	1.2.0.0
Managem...	Reset To...	False
Managem...	Debug T...	0
Event Pro...	TagId M...	0::

Figure 2-6 *Advanced Tab Window*

RF Properties

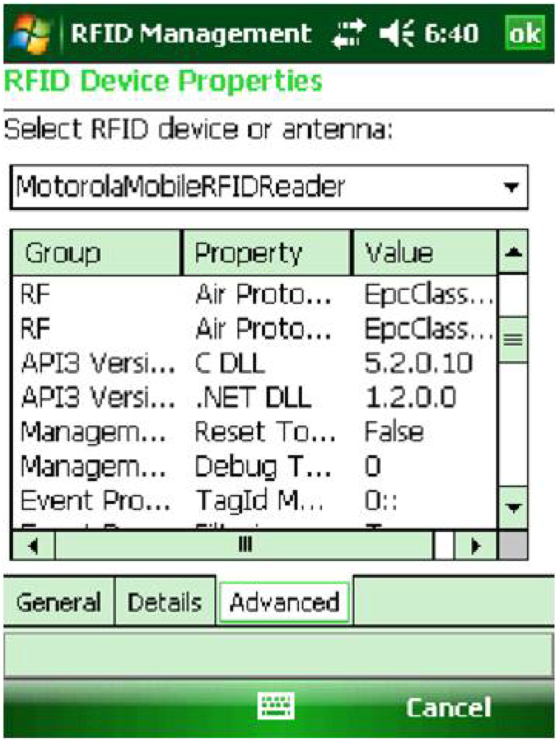
Air Protocol

Read-only properties with hardcoded value "EpcClass1Gen2". This can be accessed using Air Protocols in Use and Air protocols supported property.

RSSI CutOff

This property specifies RSSI cutoff as a percentage value, above which a tag will be considered as valid.

.



Select RFID device or antenna:

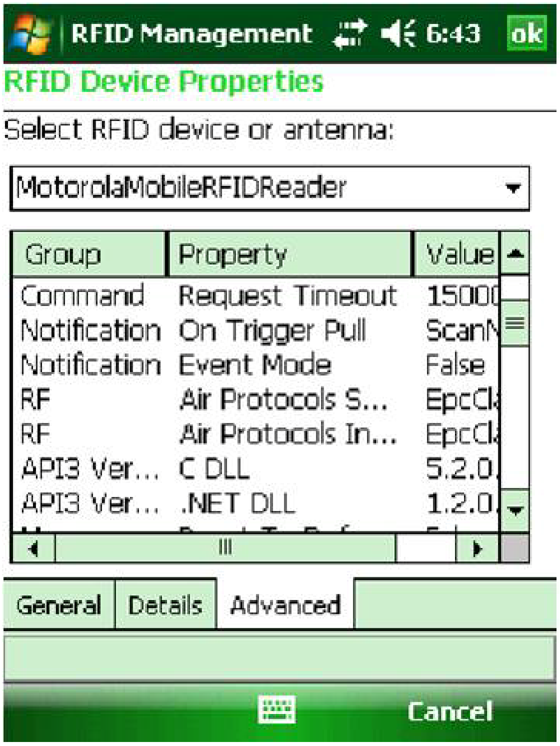
MotorolaMobileRFIDReader

Group	Property	Value
RF	Air Proto...	EpcClass...
RF	Air Proto...	EpcClass...
API3 Versi...	C DLL	5.2.0.10
API3 Versi...	.NET DLL	1.2.0.0
Managem...	Reset To...	False
Managem...	Debug T...	0
Event Pro...	TagId M...	0::

Figure 2-7 *Advanced Tab Window*

Command Properties

.



Select RFID device or antenna:

MotorolaMobileRFIDReader

Group	Property	Value
Command	Request Timeout	1500
Notification	On Trigger Pull	ScanN
Notification	Event Mode	False
RF	Air Protocols S...	EpcCl
RF	Air Protocols In...	EpcCl
API3 Ver...	C DLL	5.2.0.
API3 Ver...	.NET DLL	1.2.0.

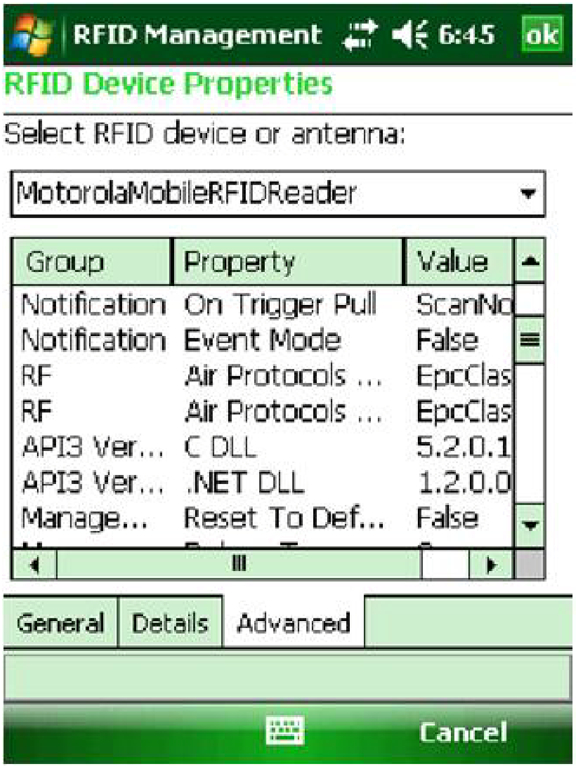
Figure 2-8 *Command Properties Window*

Request Timeout (ms)

Specifies the timeout period (in milliseconds) for request messages, the Zebra device provider sends a timeout response after this period has expired.

Notification Properties

.



RFID Device Properties

Select RFID device or antenna:

MotorolaMobileRFIDReader

Group	Property	Value
Notification	On Trigger Pull	ScanNo
Notification	Event Mode	False
RF	Air Protocols ...	EpcClas
RF	Air Protocols ...	EpcClas
API3 Ver...	C DLL	5.2.0.1
API3 Ver...	.NET DLL	1.2.0.0
Manage...	Reset To Def...	False

Figure 2-9 *Notification Properties Window*

Event Mode

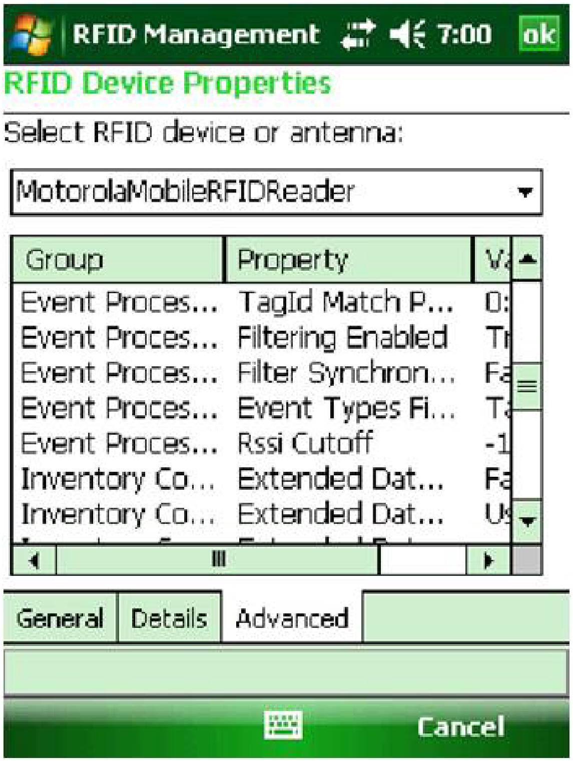
This is a read-only property which is set to false and added for backward compatibility.

OnTriggerPull

This indicates the action that should be taken when a trigger is pressed. Using OnTriggerPull, you can also specify what the platform should read when the user presses the trigger (RFID source/barcode/both/None).

Custom Device Properties

Custom Properties are generally divided in to reader device specific properties and Antenna Source properties. The number of antenna source properties depends on the reader model.



RFID Device Properties

Select RFID device or antenna:

MotorolaMobileRFIDReader

Group	Property	Va
Event Proces...	TagId Match P...	0:
Event Proces...	Filtering Enabled	Tr
Event Proces...	Filter Synchron...	Fa
Event Proces...	Event Types Fi...	Ta
Event Proces...	Rssi Cutoff	-1
Inventory Co...	Extended Dat...	Fa
Inventory Co...	Extended Dat...	Us

Figure 2-10 *Custom Device Properties Window*

Event Processing Service Properties

Filtering Enabled

Master switch which determines whether filtering is enabled in the Zebra device provider or not.

Filter Synchronous Tags

Used to determine whether the filtering parameters should apply to tags, read synchronously as well (e.g. when doing a GetTags operation on a device).

✓ ***NOTE*** The filtering parameters always apply to asynchronous tags (e.g. during inventory using *ContinuousRead* and *OnTiggerPull*).

Event Types Filter

Used to filter events based on the type (e.g. *TagReadEvent* or None).

TagReadEvent is the default value which specifies that user is interested in *TagReadEvent*.

None indicates that the user is interested in no events and thus no events are reported.

TagId Match Pattern

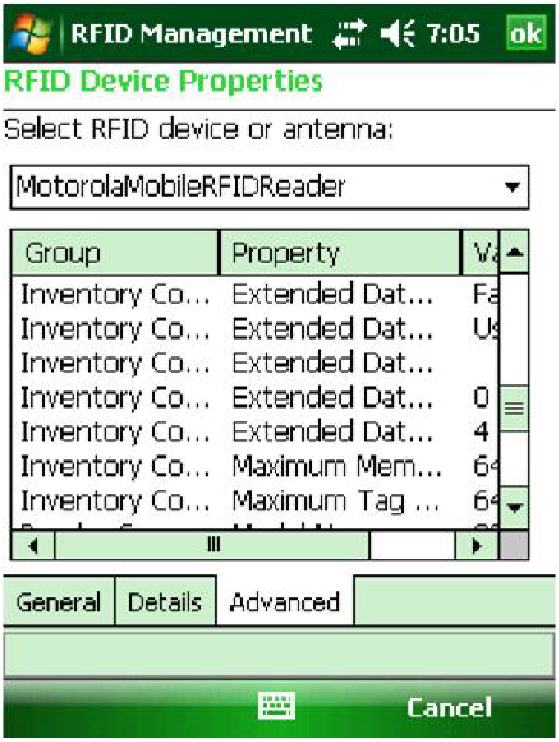
Specifies the TagID pattern which should be matched if the tag is surfaced to the application.

RSSI CutOff

The application can set an *RSSI Cutoff* on the device, so that only tags with an RSSI higher than a specified value will be surfaced to the application. This is used to filter stray tag reads.

Inventory Control Properties

.



RFID Device Properties

Select RFID device or antenna:

MotorolaMobileRFIDReader

Group	Property	Va
Inventory Co...	Extended Dat...	Fa
Inventory Co...	Extended Dat...	U
Inventory Co...	Extended Dat...	
Inventory Co...	Extended Dat...	0
Inventory Co...	Extended Dat...	4
Inventory Co...	Maximum Mem...	6
Inventory Co...	Maximum Tag ...	6

Figure 2-11 *Inventory Control Properties Window*

Inventory can be enabled to either return only the EPC (TagId) of the tags in front of the reader, or to return the EPC along with data from some other memory bank of the tag. The following information explains when each of these is used, and how they operate:

Extended Data Enabled, Memory Bank Id, Memory Bank Passcode, Extended Data Byte Offset, Extended Data Byte Count

Extended Data Enabled is the switch which determines whether to execute *Perform Inventory* on the reader (which reads only the EPC), or whether to execute *Perform Sequence*, which is used to read extended data from some other memory bank also on the reader.

The other parameters are used only when Extended Data Enabled is true, and are used to specify which bank should be read - TID, User or Reserved (using the Memory Bank Id property), from where to start reading (Extended Data Byte Offset), how many bytes to read (Extended Data Byte Count) and the passcode in case it is required (Memory Bank Passcode).

Maximum TagId Byte Count, Maximum Memory Bank Byte Count

These properties are set in the server side RFID3 API DLL, and is used to configure the maximum length of the EPC that will be read, and the maximum length of the extended data that will be read.

API3 version related properties

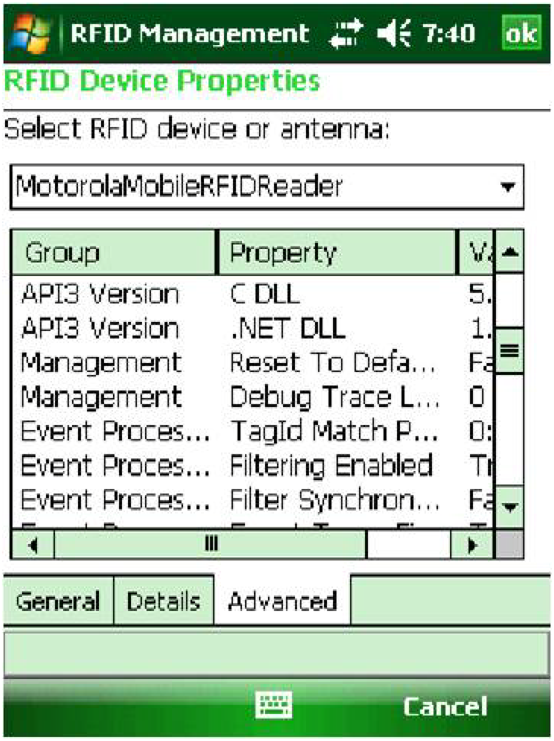
C DLL

Read only property indicates the Version of the RFID3 API C DLL used internally.

.NET DLL

Read only property indicates the Version of the RFID3 API .NET DLL used internally.

.



RFID Device Properties

Select RFID device or antenna:

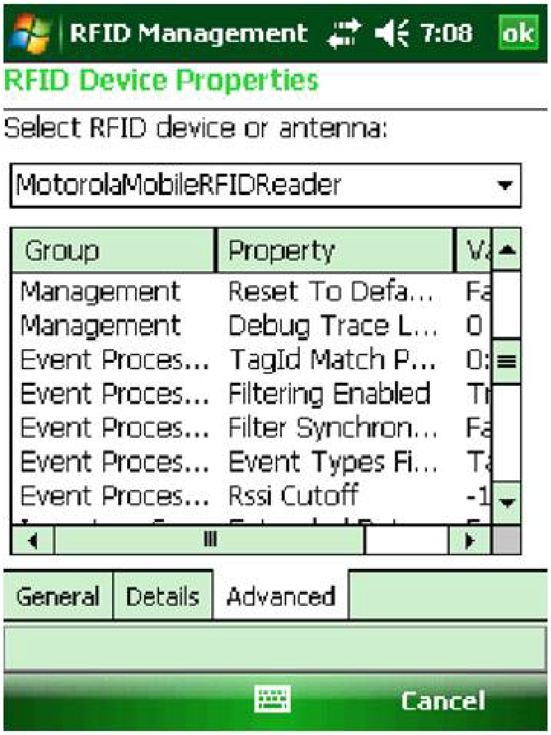
MotorolaMobileRFIDReader

Group	Property	V
API3 Version	C DLL	5.
API3 Version	.NET DLL	1.
Management	Reset To Defa...	Fa
Management	Debug Trace L...	0
Event Proces...	TagId Match P...	0:
Event Proces...	Filtering Enabled	Tr
Event Proces...	Filter Synchron...	Fa

Figure 2-12 *API3 version Properties Window*

Management Related Properties

.



Select RFID device or antenna:

MotorolaMobileRFIDReader

Group	Property	Va
Management	Reset To Defa...	Fa
Management	Debug Trace L...	0
Event Proces...	TagId Match P...	0:
Event Proces...	Filtering Enabled	Tr
Event Proces...	Filter Synchron...	Fa
Event Proces...	Event Types Fi...	T
Event Proces...	Rssi Cutoff	-1

Figure 2-13 *Management Properties Window*

Reset to Factory Defaults

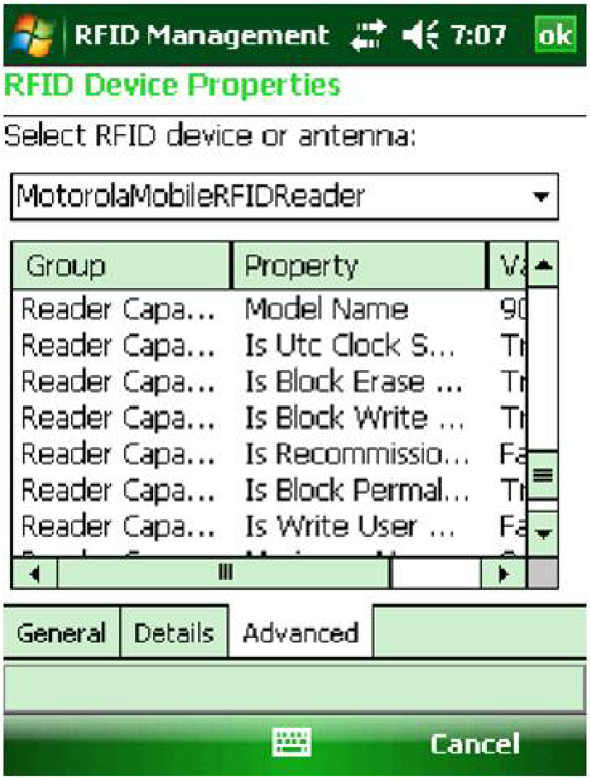
This property can be set to true, which resets the reader to factory defaults.

Debug Trace Level

This sets the trace level on the RFID3 host side DLL, and is used for debugging purposes. It is always recommended to set this to 0 (default) for normal operations.

Reader Capabilities Properties

.



RFID Device Properties

Select RFID device or antenna:

MotorolaMobileRFIDReader

Group	Property	V
Reader Capa...	Model Name	90
Reader Capa...	Is Utc Clock S...	Tr
Reader Capa...	Is Block Erase ...	Tr
Reader Capa...	Is Block Write ...	Tr
Reader Capa...	Is Recommissio...	Fa
Reader Capa...	Is Block Permal...	Tr
Reader Capa...	Is Write User ...	Fa

Figure 2-14 *Reader Capabilities* *Window*

The following properties are read-only and reflect the reader capabilities:

- Model Name
- Is Utc Clock Supported
- Is Block Erase Supported
- Is Block Write Supported
- Is Recommission Supported
- Is Block Permalock Supported
- Is Write User Memory Indicator Supported
- Maximum Number of Operations in Access Sequence
- Maximum number of pre-filters

Source Properties

This section covers properties of the device sources (i.e. antennas and bar code). All standard properties indicated below except *PowerLevel* will also be applicable to bar code source.

System Enabled

Read-only property. Specifies whether the source is enabled by the device.

Enabled

Read only property which always set to true.

Source Type

Read only property which specifies the type of source (i.e Antenna).

ContinuousRead

Indicates whether the inventory on source is continuously on.

Name

This is a read-only property hardcoded as *RFID Antenna* for RFID source and *BarcodeReader* for bar code source.

Location

Read-only property hardcoded as *Inproc Antenna Port* for RFID source and *Inproc Barcode Port* for bar code source.

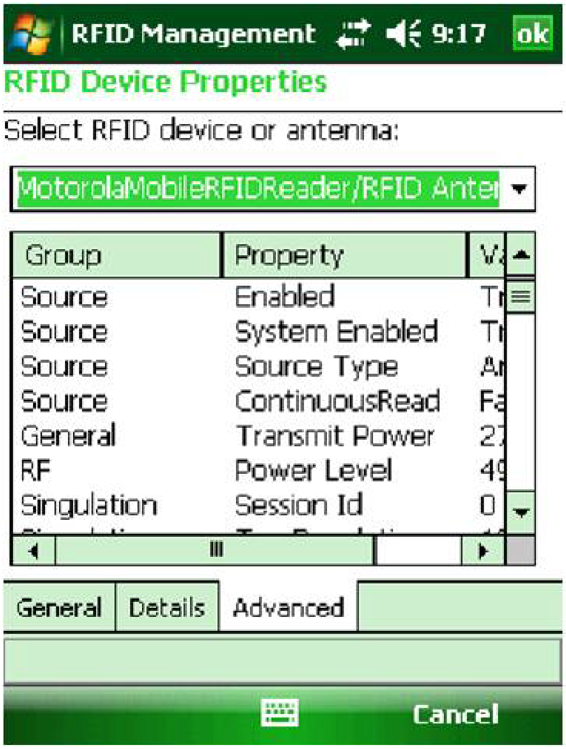
Eventmode

Read-only property which is set to false and added for backward compatibility.

Power Level

This is exposed as a percentage (as mandated by the BizTalk RFID standard properties).

.



RFID Device Properties

Select RFID device or antenna:

MotorolaMobileRFIDReader/RFID Ante

Group	Property	Va
Source	Enabled	Tr
Source	System Enabled	Tr
Source	Source Type	Ar
Source	ContinuousRead	Fa
General	Transmit Power	27
RF	Power Level	49
Singulation	Session Id	0

Figure 2-15 *Source Properties* *Window*

Antenna Configuration Custom Properties

Gain

Specifies the gain of the antenna. This is a read-only property.

Receive Sensitivity

This is exposed as an absolute number (in dBm), and corresponds to the sensitivity of the antennas. It is mapped to the receive sensitivity table in the reader capabilities.

RF Mode

Used to set the RF Mode which the reader operates in.

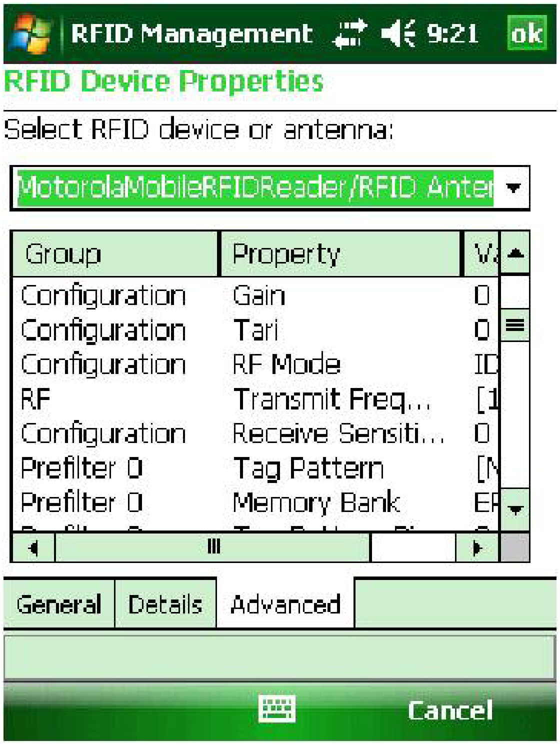
Tari

Used to set the Tari on the reader.

Transmit Power

Used to set the Transmit Power on the reader in dbm.

.



RFID Device Properties

Select RFID device or antenna:

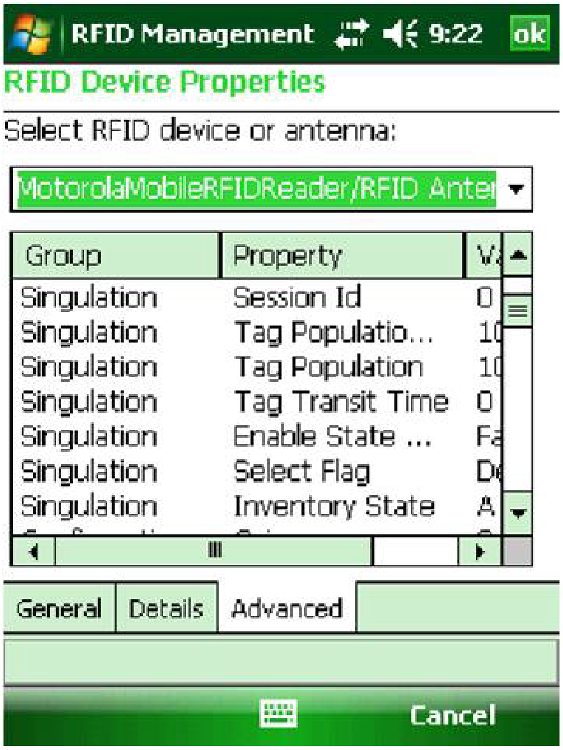
MotorolaMobileRFIDReader/RFID Anter

Group	Property	Va
Configuration	Gain	0
Configuration	Tari	0
Configuration	RF Mode	ID
RF	Transmit Freq...	[1
Configuration	Receive Sensiti...	0
Prefilter 0	Tag Pattern	[N
Prefilter 0	Memory Bank	EP

Figure 2-16 *Antenna Configuration Properties Window*

Singulation and Pre-filter Properties

.



RFID Device Properties

Select RFID device or antenna:

MotorolaMobileRFIDReader/RFID Anter

Group	Property	V:
Singulation	Session Id	0
Singulation	Tag Populatio...	1(
Singulation	Tag Population	1(
Singulation	Tag Transit Time	0
Singulation	Enable State ...	Fa
Singulation	Select Flag	De
Singulation	Inventory State	A

Figure 2-17 *Singulation and Pre-filter Properties Window*

Singulation properties: Tag Transit Time, Tag Population Estimation, Session Id

These properties specify the Gen2 specific properties of the reader to be used during singulation (e.g. what is the expected tag population when the application is running, and in what session should the tags be singulated).:

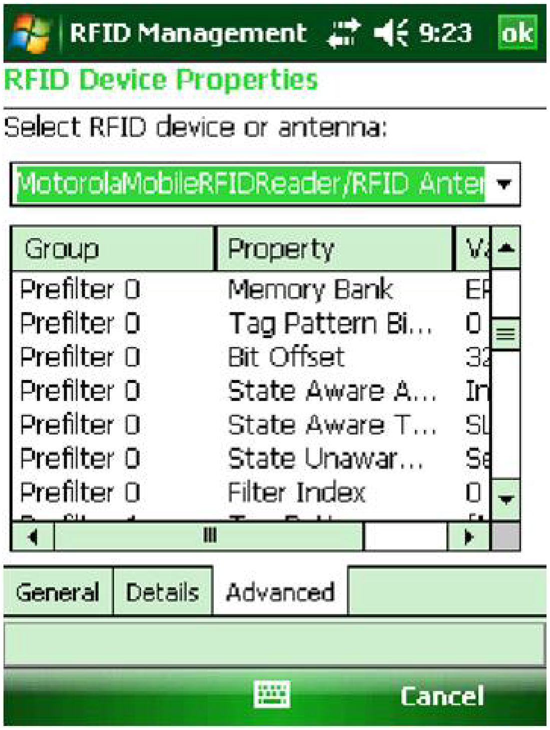
✓ ***NOTE*** Property *Tag Population* which does the same function as *Tag Population Estimation* (defined in v1.0.0.3) is also retained.

Singulation properties: Enable State Aware, Inventory State, Select Flag

Enabled State Aware is the master switch which controls these properties. These properties should be changed in advanced scenarios only when the user wants complete control over how the tags are read and processed. When *Enable State Aware* is true, then the application has control over which session/state it singulates the tag into (e.g. it can choose to assert/deassert tags through the *Select Flag* property into *Inventory State A* or *B* through the *Inventory State* property). This also decides the Filter Action of the pre-filters for that antenna.

Inventory State can have values *A* or *B*.

Select Flag can have values *Asserted* or *Deasserted*.



RFID Device Properties

Select RFID device or antenna:

MotorolaMobileRFIDReader/RFID Ante

Group	Property	Va
Prefilter 0	Memory Bank	EF
Prefilter 0	Tag Pattern Bi...	0
Prefilter 0	Bit Offset	32
Prefilter 0	State Aware A...	In
Prefilter 0	State Aware T...	Sl
Prefilter 0	State Unawar...	Se
Prefilter 0	Filter Index	0

Figure 2-18 *Pre-filter Properties Window*

Pre-filter properties: Tag Pattern, Tag Pattern Bit Count, State Aware Action, State Aware Target, State Unaware Action, Bit Offset, Memory Bank

These properties describe a single pre-filter.

Tag Pattern is the master switch for this property group. When this is set to *None*, this corresponds to *no pre-filter*, and none of the other parameters are considered.

Tag Pattern Bit Count is the number of bits in the *Tag Pattern* to be taken for comparison.

Bit Offset is the first (msb) bit location of the specified memory bank against which to compare the *Tag Pattern*.

Memory Bank is the memory bank against which the pre-filter must operate.

State Aware Action and *State Aware Target* specify the behavior of the pre-filter when the Singulation is *State Aware* (see the [*Singulation, Enable State Aware*] property).

State Aware Action can have the following values:

- Inv A Not Inv B
- Inv A
- Not Inv B
- Inv A2BB2A Not Inv A
- Inv B Not Inv A
- Inv B
- Not Inv A
- Not Inv A2BB2A

State Aware Target can have the following values (specifies which session the singulated tags should be put into):

- SL
- S0
- S1
- S2
- S3

State Unaware Action can have the following values (corresponds to the behavior of the pre-filter when the singulation is *State Unaware):*

- Select Not Unselect
- Select
- Not Unselect
- Unselect
- Unselect Not Select
- Not Select

Pre-filter Poperties: Filter Index

This property decides the order in which the pre-filters are added onto the device. The value set by the user is only indicative of the relative order in which the pre-filters should be added to the device. The actual value of the index stored on the device (and returned in a call to *GetProperty* or *GetPropoertyProfile*) could be different.

RF Properties

Transmit Frequency

Used to set the transmit frequency of the reader:

- If *Frequency Hopping* is enabled, the user can select from a drop down of hop tables (where each hop table is represented as a semi-colon separated string of individual frequency values in that hop table).
- If *Frequency Hopping* is not enabled, the user can select from a drop down of fixed frequency values available on the reader.

CHAPTER 3 BASIC OPERATIONS

Introduction

This chapter includes information on how to use the Zebra provider.

Reading Tags

The tags can be read asynchronously by setting *ContinuousRead* or *OnTriggerPull* property. For additional information, see *Device Properties on page 2-4* and *Source Properties on page 2-16*.

Additionally, use the *GetTags synchronous* command to read tags.This command runs inventory for a configurable period of time (default = 3 seconds) and then returns all the tags that were observed during this period.

The following vendor extensions are supported for each tag:

- Channel index
- First Seen Timestamp UTC
- Last Seen Timestamp UTC
- First Seen Timestamp UTC Uptime
- Last Seen Timestamp UTC Uptime
- PC
- XPC
- CRC

CHAPTER 4 SUPPORTED COMMANDS

Introduction

This chapter includes information on the commands supported by the Zebra Provider, including:

- General - commands pertaining to properties or diagnostics.
- Gen2 - commands that operate on a tag and comply with the Gen2 standard.
- Other - commands to get tags from the device synchronously, as well as advanced commands.

General Commands

Following are commands supported by the device.

GetCurrentPropertyProfileCommand

Return the current property profile on the device. Refer to *Chapter 2, Configuration* section for more information on the properties supported by the provider.

GetPropertyCommand

Return a specific property to the application.

GetDefaultPropertyProfileCommand

Returns the default values for all properties supported by the provider.

SetPropertyCommand

Set a property on the reader module.

ApplyPropertyProfileCommand

Apply the properties specified by the application on the device.

SetReadFilter

This can be used (as an alternative to the Event Processing Services Tag Id Match Pattern property) to filter tags from the device.

NOTE Tags can be filtered based on *Tagid* only

GetReadFilter

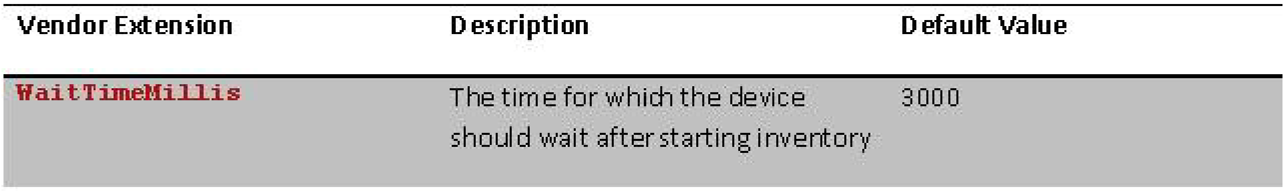
This retrieves the filters that have been set using *SetReadFilter*.

GetTags

This retrieves the tags that are currently in the Field Of View of the device.

The device starts inventory and then waits for a configurable amount of time before returning with the tags that were read.

Vendor extensions supported:



Vendor Extension	Description	Default Value
WaitTimeMillis	The time for which the device should wait after starting inventory	3000

Figure 4-1 *GetTags Vendor Extensions*

NOTE Passcode field in this command is not applicable for Zebra provider.

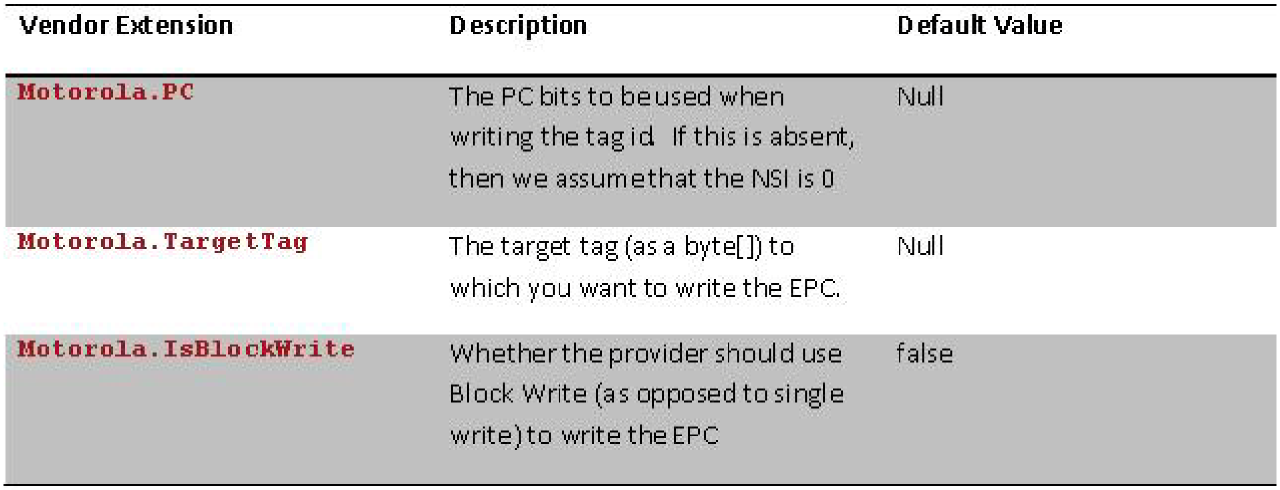
Gen2 Commands

Following is the support for the tag-based commands, including Gen2 commands:

WriteID

Here, potentially writing 3 pieces of data to the tag is needed: the EPC, the access code and / or the kill code. Internally, this calls the *TagAccess.WriteWait() API* for targeted write, and calls the *PerformSequence() API* for non-targeted write (single blind write), in which the appropriate access spec is set up with *TagObservation.N = 1* (i.e. execute the operation on exactly 1 tag).

Vendor extensions supported:



Vendor Extension	Description	Default Value
Motorola.PC	The PC bits to beused when writing the tag id. If this is absent, then we assumethat the NSI is 0	Null
Motorola.TargetTag	The target tag (as a byte[]) to which you want to write the EPC.	Null
Motorola.IsBlockWrite	Whether the provider should use Block Write (as opposed to single write) to write the EPC	false

Figure 4-2 *WriteID Vendor Extensions*

✓ ***NOTE*** Vendor extension called *TargetId* which does the same function as *Zebra.TargetTag* is retained to provide backward compatibility.

Writing the Tag Id

This is written onto the EPC memory bank of the tag along with the PC bits i.e. calculate the PC bits (if the PC bits are not supplied), and start writing the tag id at word pointer 1.

Kill Code

This is written onto the Reserved bank of the tag starting at word pointer 0, and for a word length of 2.

Access Code

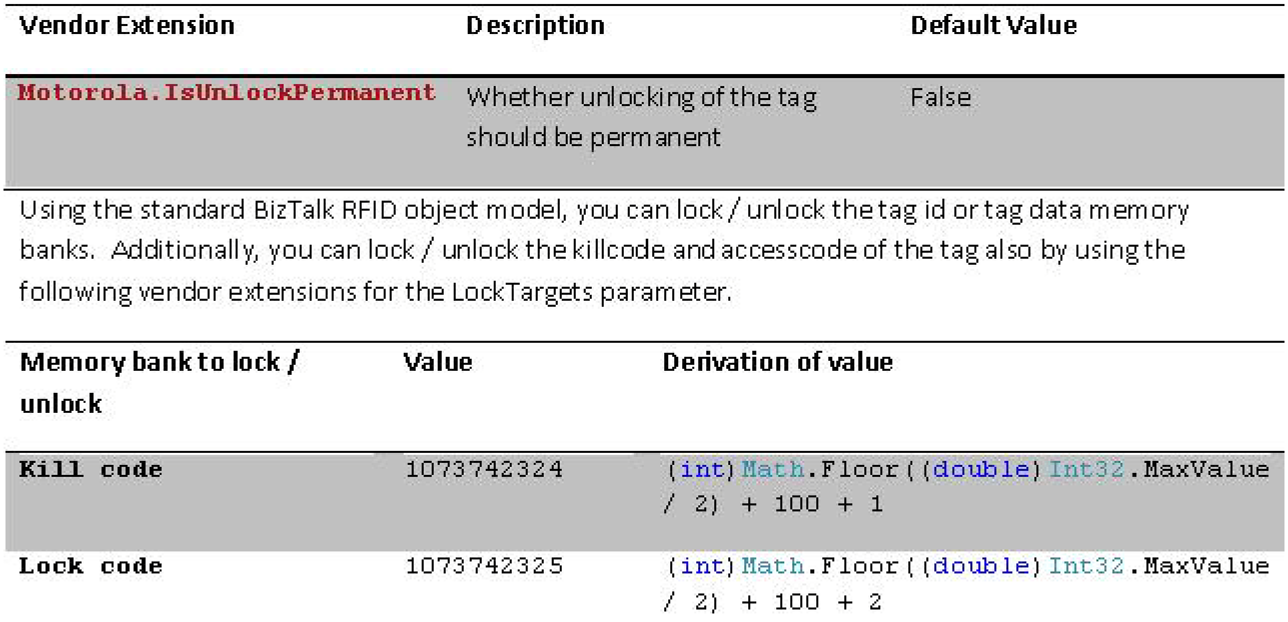
This is written onto the Reserved bank of the tag starting at word pointer 2, for a word length of 2.

LockTag, UnlockTag

Used to lock/unlock the EPC bank (standard functionality) or any of the other memory banks (vendor extension). Internally, this calls the *TagAccess.LockWait() API*.

The lock privilege for the operation is set based on whether the lock/unlocking is permanent or not.

The following vendor extensions are supported:



Vendor Extension	Description	Default Value
`Motorola.IsUnlockPermanent`	Whether unlocking of the tag should be permanent	False

Using the standard BizTalk RFID object model, you can lock / unlock the tag id or tag data memory banks. Additionally, you can lock / unlock the killcode and accesscode of the tag also by using the following vendor extensions for the LockTargets parameter.

Memory bank to lock / unlock	Value	Derivation of value
`Kill code`	1073742324	`(int)Math.Floor((double)Int32.MaxValue / 2) + 100 + 1`
`Lock code`	1073742325	`(int)Math.Floor((double)Int32.MaxValue / 2) + 100 + 2`

Figure 4-3 *LockTag / Unlock Tag Vendor Extensions*

Per Microsoft guidelines, vendor extensions to the LockTargets structure must begin at *(int)Math.Floor((double)Int32.MaxValue / 2)*, which is the basis for the above assigned values.

e.g. *LockTarget* for the kill code can be created as follows (which can be passed as parameter to *LockTagCommand* or *UnlockTagCommand*):

```
LockTargets killCodeLockTarget = new LockTargets(1073742324, "Kill code lock target");
```

✓ ***NOTE*** Only one target can be locked at a time.

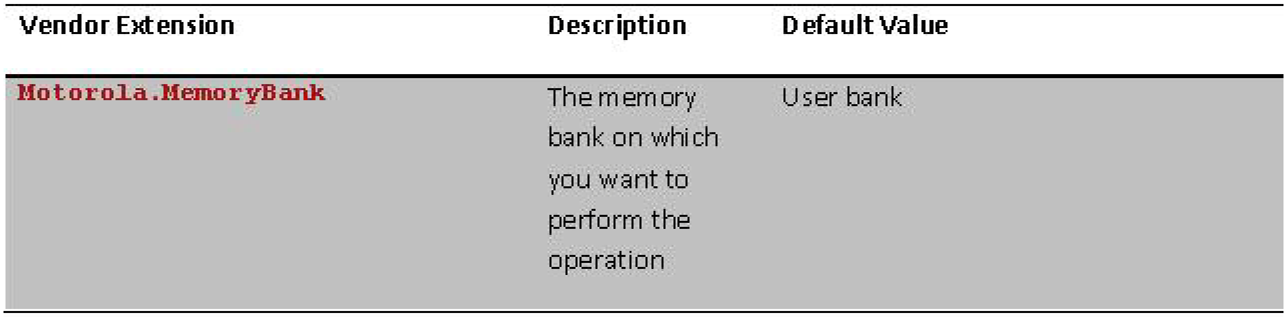
Kill

Kills the tag specified.

GetTagData

Can be used to read all the data from a memory bank.

Vendor extensions supported:



Vendor Extension	Description	Default Value
Motorola.MemoryBank	The memory bank on which you want to perform the operation	User bank

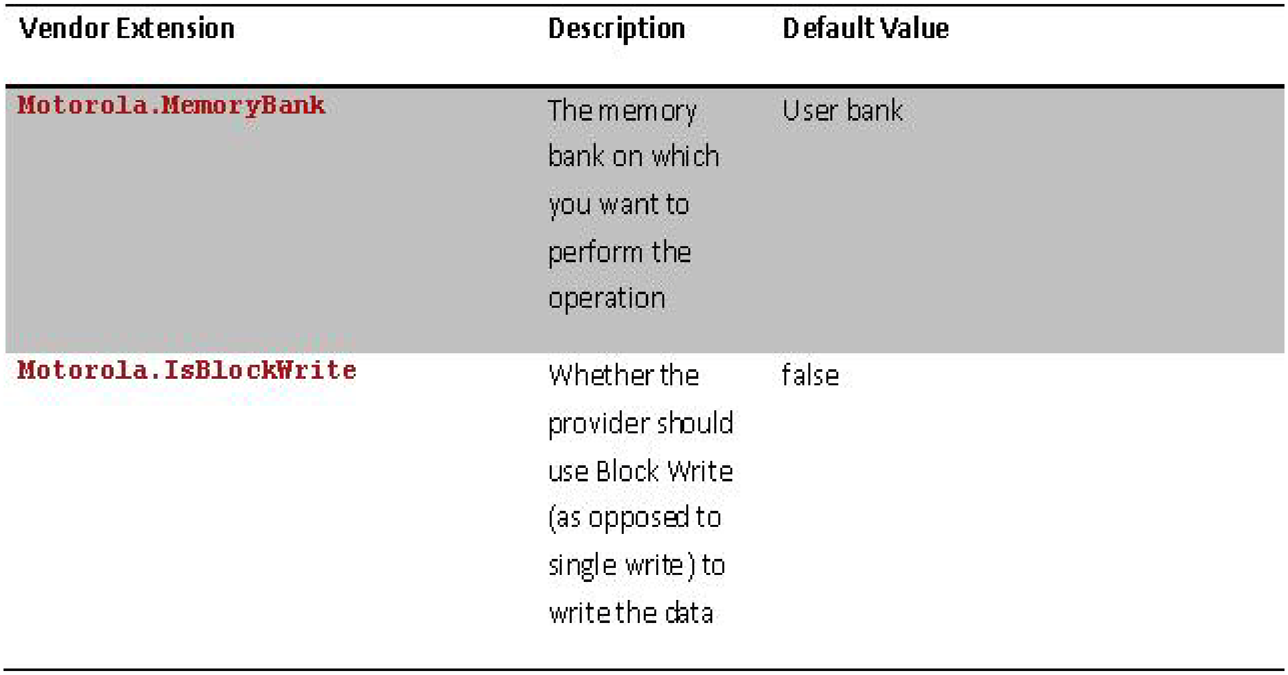
Figure 4-4 *GetTag Data Vendor Extensions*

WriteTagData

Used to write data to a memory bank.

NOTE The provider does not zero out the data in the remainder of the bank.

Vendor extensions supported:



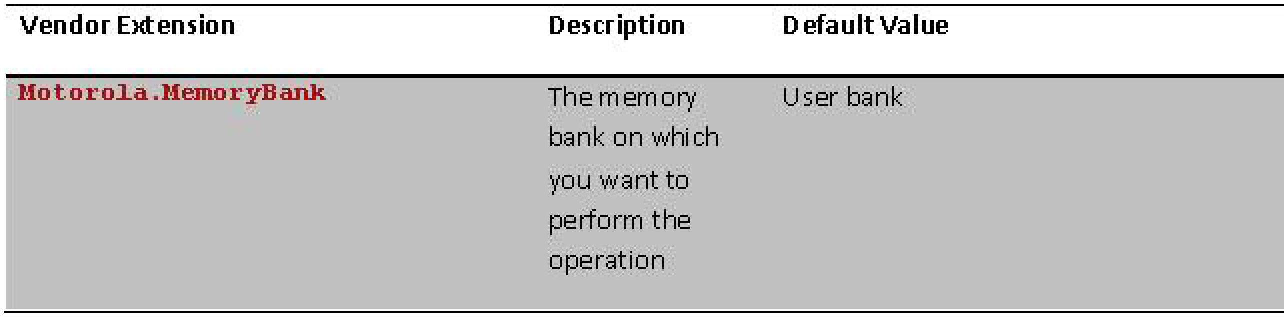
Vendor Extension	Description	Default Value
Motorola.MemoryBank	The memory bank on which you want to perform the operation	User bank
Motorola.IsBlockWrite	Whether the provider should use Block Write (as opposed to single write) to write the data	false

Figure 4-5 *WriteTagData Vendor Extensions*

GetPartialTagData

Used to read partial data from a memory bank. The provider only supports *SeekOrigin = SeekOrigin.Begin***.**

Vendor extensions supported:



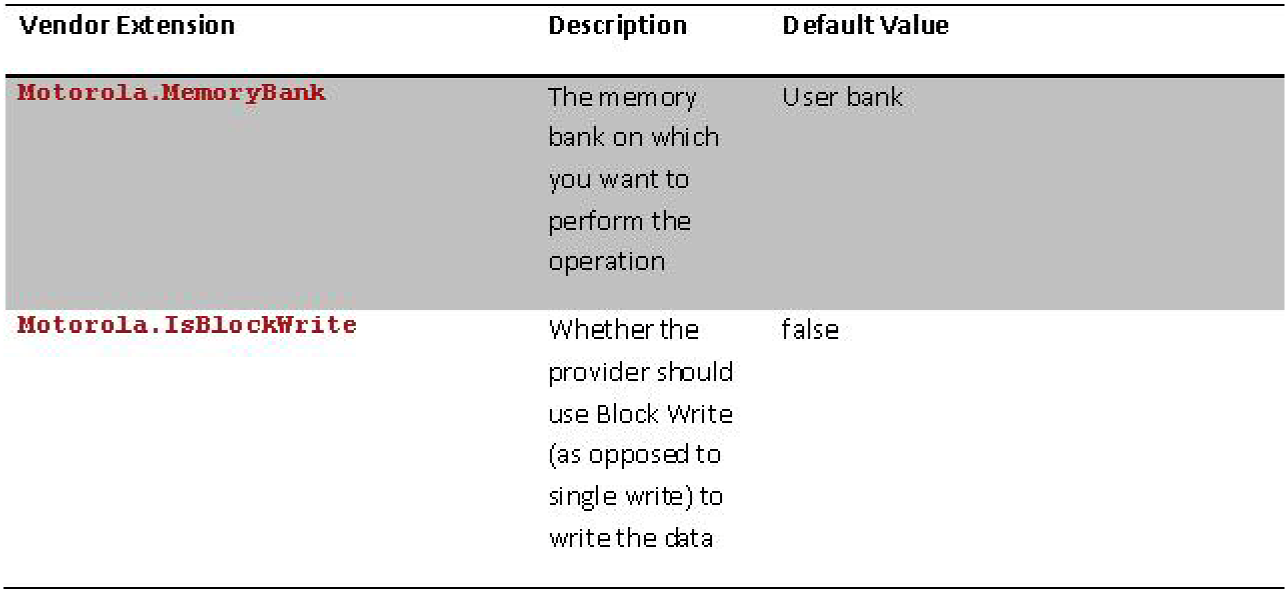
Vendor Extension	Description	Default Value
Motorola.MemoryBank	The memory bank on which you want to perform the operation	User bank

Figure 4-6 *GetPartialTagData Vendor Extensions*

WritePartialTagData

The provider writes to a part of the specified data bank. It only supports *SeekOrigin* = *SeekOrigin.Begin*.

Vendor extensions supported:



Vendor Extension	Description	Default Value
Motorola.MemoryBank	The memory bank on which you want to perform the operation	User bank
Motorola.IsBlockWrite	Whether the provider should use Block Write (as opposed to single write) to write the data	false

Figure 4-7 *WritePartialTagData Vendor Extensions*

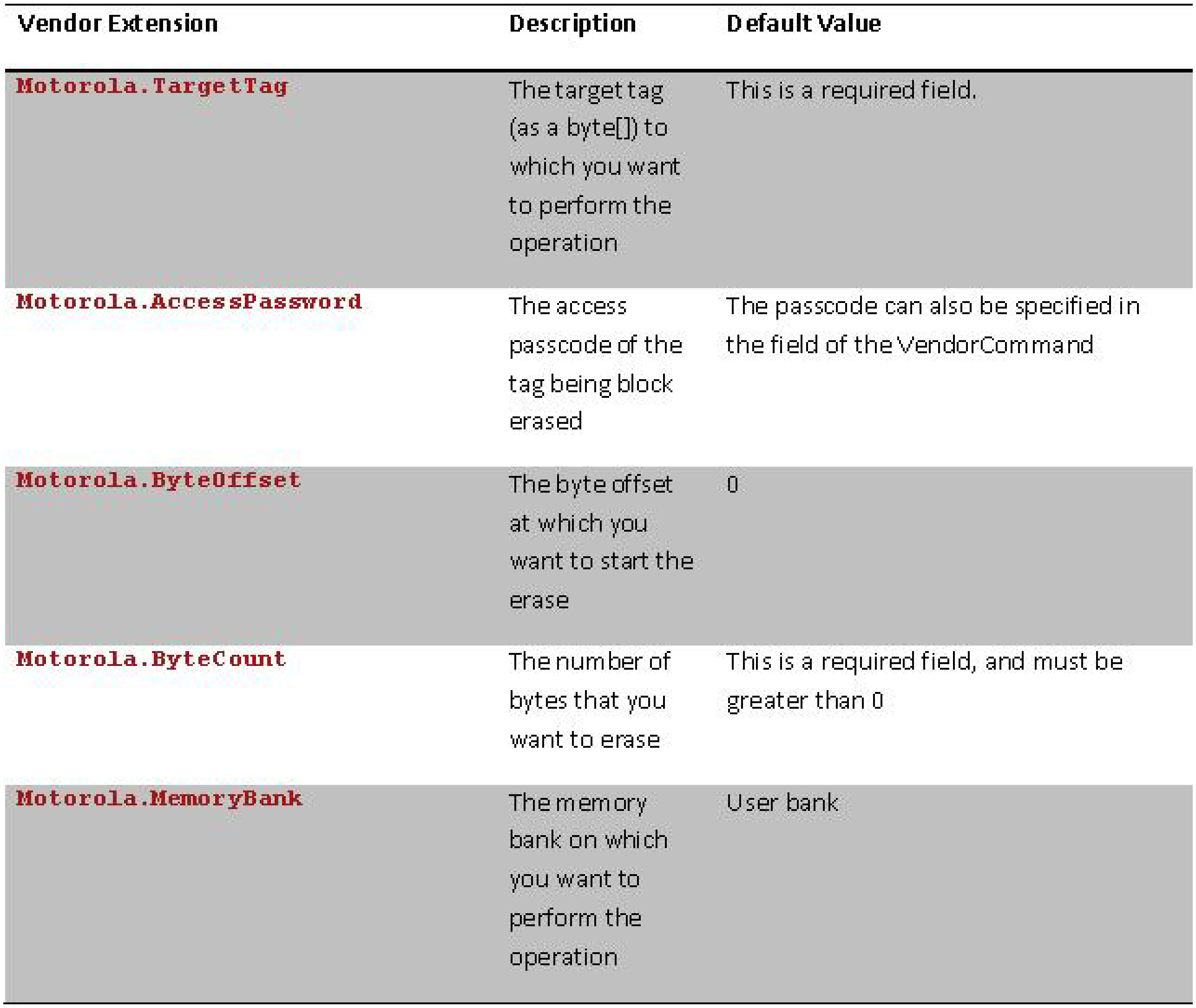
Vendor Commands

The Zebra provider exposes advanced device functionality through the following vendor commands:

BlockErase

This command erases the memory from the specified bank. To use this functionality, the key *VendorDefinedCommand.Name* must be set to *Zebra.BlockErase* in the *VendorDefinedCommand* object.

Vendor parameters expected for this command are:



Vendor Extension	Description	Default Value
Motorola.TargetTag	The target tag (as a byte[]) to which you want to perform the operation	This is a required field.
Motorola.AccessPassword	The access passcode of the tag being block erased	The passcode can also be specified in the field of the VendorCommand
Motorola.ByteOffset	The byte offset at which you want to start the erase	0
Motorola.ByteCount	The number of bytes that you want to erase	This is a required field, and must be greater than 0
Motorola.MemoryBank	The memory bank on which you want to perform the operation	User bank

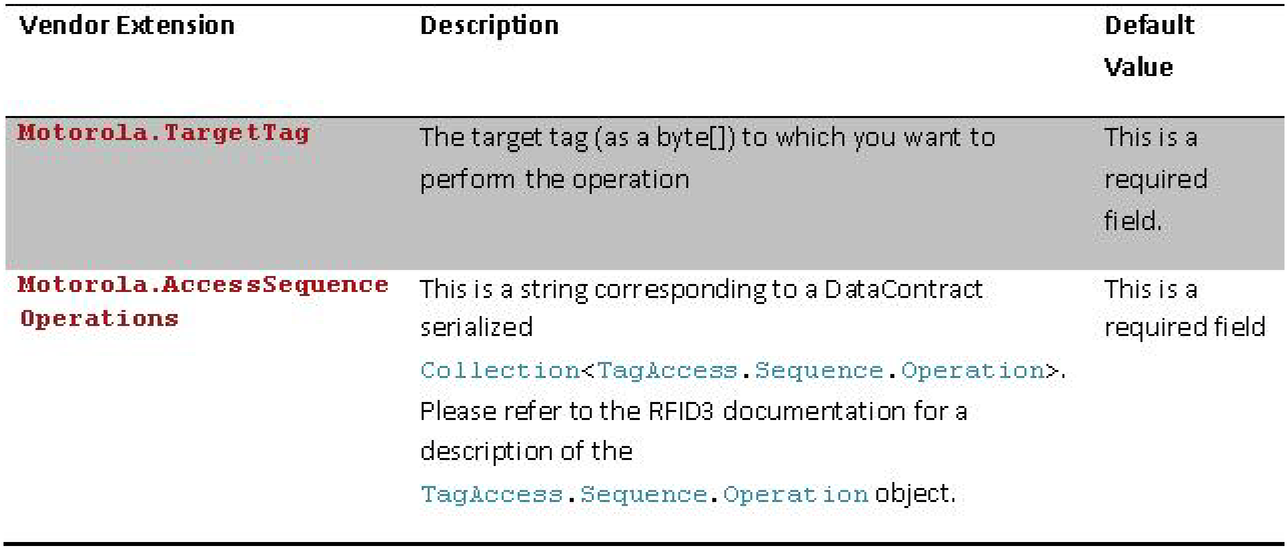
Figure 4-8 *Block Erase Vendor Extensions*

AccessSequence

This command is used to batch operations at the device layer e.g. if the user wants to write the passcode and killcode, lock the tag, and then write the EPC by supplying the passcode in a single operation instead of making multiple round-trips to the device, that can be achieved using the *AccessSequence* vendor extension.

To use this functionality, the key *VendorDefinedCommand.Name* must be set to *Zebra.AccessSequence* in the *VendorDefinedCommand* object.

Vendor parameters expected for this command are:



Vendor Extension	Description	Default Value
Motorola.TargetTag	The target tag (as a byte[]) to which you want to perform the operation	This is a required field.
Motorola.AccessSequence Operations	This is a string corresponding to a DataContract serialized Collection<TagAccess.Sequence.Operation>. Please refer to the RFID3 documentation for a description of the TagAccess.Sequence.Operation object.	This is a required field

Figure 4-9 *Access Sequence Vendor Extensions.*

✓ ***NOTE*** If the *Access Sequence* contains an operation to write the EPC of a tag, this must be the last operation in the sequence. *Read Access* Operation is not allowed as part of *Access Sequence* Operations.

In order to get the access operation results, the following are a added part of the *TagReadEvent* / *TagListEvent* vendor extension's:

- Operation code
- Operation Status
- Operation Result
- MemoryBank Data

✓ ***NOTE*** *TagReadEvent* should be registered to view the above results and property of *deviceconnection* object *DuplicateEliminationInterval* should be set to 0 (zero).

Below is an example of how to unlock a tag and then write to it in a single Access Sequence:

```
Collection<TagAccess.Sequence.Operation> UnlockAndWrite =
        new Collection<TagAccess.Sequence.Operation>();

//First unlock the tag
TagAccess.Sequence.Operation unlockTag = new TagAccess.Sequence.Operation();

unlockTag.AccessOperationCode = ACCESS_OPERATION_CODE.ACCESS_OPERATION_LOCK;
unlockTag.LockAccessParams = new TagAccess.LockAccessParams();


//Assume that 860116326 is the current passcode for the tag
unlockTag.LockAccessParams.AccessPassword = 860116326;
      unlockTag.LockAccessParams.LockPrivilege[unlockTag.LockAccessParams.EPCMemory] =
LOCK_PRIVILEGE.LOCK_PRIVILEGE_UNLOCK;
UnlockAndWrite.Add(unlockTag);

//Now write to the tag again without an access code
TagAccess.Sequence.Operation writeEpcWithNoPasscode = new TagAccess.Sequence.Operation();
writeEpcWithNoPasscode.AccessOperationCode = ACCESS_OPERATION_CODE.ACCESS_OPERATION_WRITE;

//Get the PC + EPC to write to the tag
byte[] pcAndEpc = …
writeEpcWithNoPasscode.WriteAccessParams = new TagAccess.WriteAccessParams();
writeEpcWithNoPasscode.WriteAccessParams.MemoryBank = MEMORY_BANK.MEMORY_BANK_EPC;
writeEpcWithNoPasscode.WriteAccessParams.ByteOffset = 2;
writeEpcWithNoPasscode.WriteAccessParams.WriteData = pcAndEpc;
writeEpcWithNoPasscode.WriteAccessParams.WriteDataLength = (uint)pcAndEpc.Length;
writeEpcWithNoPasscode.WriteAccessParams.AccessPassword = 0; //no access pwd

UnlockAndWrite.Add(writeEpcWithNoPasscode);

VendorDefinedParameters vendorParams = new VendorDefinedParameters();

vendorParams.InputParameters = new VendorSpecificInformation();

vendorParams.InputParameters["Motorola.AccessSequenceOperations"] =
        RfidCommon.SerializeToXmlDataContract(UnlockAndWrite, true);

vendorParams.InputParameters["Motorola.TargetTag"] = targetTagId;
dc.ExecuteVendorDefinedCommand(null, "Motorola", "Motorola.AccessSequence", "Motorola.AccessSequence", null,
vendorParams);
```

INDEX

Tell Us What You Think...

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

Zebra Technologies Corporation
Lincolnshire, IL U.S.A.
http://www.zebra.com
Attention: Advanced Data Capture
Technical Publications Manager

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

Manual Title:____
(please include revision level)

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

□ Very familiar □ Slightly familiar □ Not at all familiar

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

____
____
____
____

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

____
____
____
____

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

____
____
____
____

What can we do to further improve our manuals?

____
____
____
____

Thank you for your input—We value your comments.

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