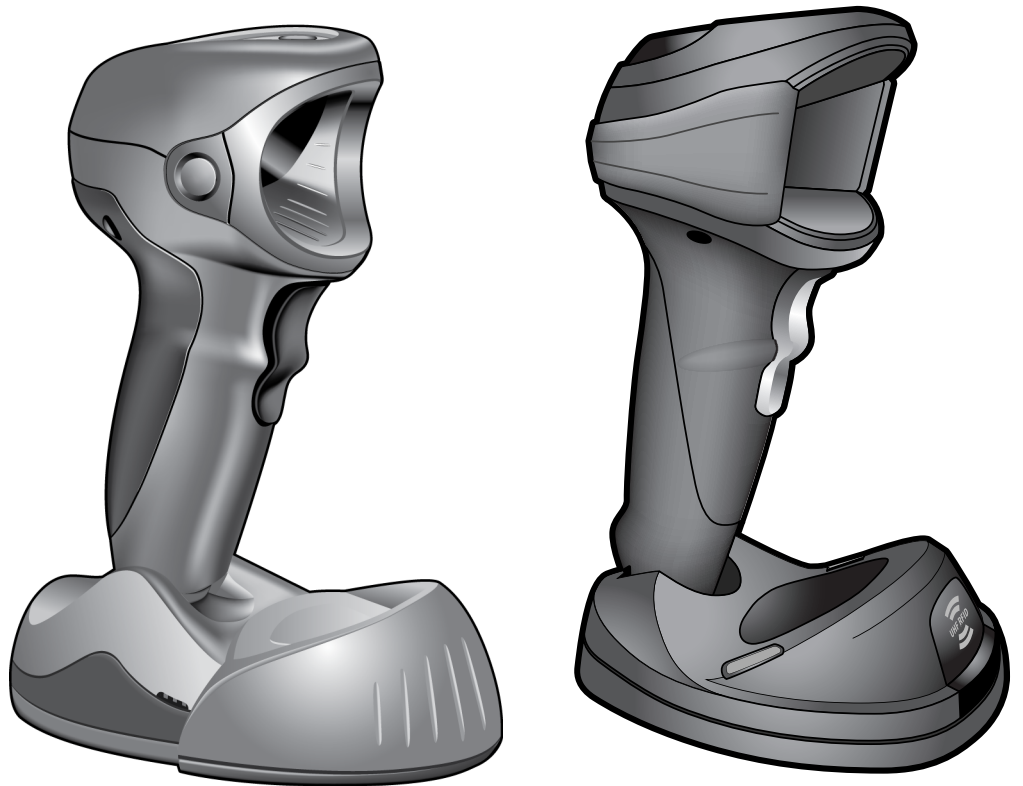


# HOST API FOR BARCODE SCANNERS WITH RFID



## Programming Interface



**ZEBRA**

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. ©2021 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: [www.zebra.com/linkoslegal](http://www.zebra.com/linkoslegal)

COPYRIGHTS: [www.zebra.com/copyright](http://www.zebra.com/copyright)

WARRANTY: [www.zebra.com/warranty](http://www.zebra.com/warranty)

END USER LICENSE AGREEMENT: [www.zebra.com/eula](http://www.zebra.com/eula)

## Terms of Use

### Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries ("Zebra Technologies"). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

### Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

### Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

### Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

## Revision History

Changes to the original guide are listed below:

Change	Date	Description
-01 Rev A	9/21/11	Initial release.
-02 Rev A	1/18/12	Update tables on pages 8, 9, 11, 13, and 14.
-03 Rev A	3/31/2015	Zebra rebranding.
-04 Rev A	2/27/2019	Add DS9908R.
-05 Rev A	8/23/2021	Remove patent-pending wording on page 8.

# Contents

Terms of Use .....	2
Proprietary Statement .....	2
Product Improvements .....	2
Liability Disclaimer .....	2
Limitation of Liability .....	2
Revision History .....	3
<b>About This Document .....</b>	<b>6</b>
Introduction .....	6
Notational Conventions .....	6
Related Documents and Software .....	7
Service Information .....	7
Provide Documentation Feedback .....	7
<b>Host API for RFID .....</b>	<b>8</b>
Overview .....	8
DS9808R Corded Hybrid Imager with RFID .....	8
DS9908R Next Generation Corded Hybrid Imager with RFID .....	8
Setup .....	9
Using with USB OPOS .....	9
RSM Attributes for RFID .....	9
RFID_LAST_TAG_ID – Last Reported Tag .....	10
RFID_CMD_STATUS - Common Operation Response .....	11
RFID_COMMAND .....	11
Tag Read Operation .....	11
Tag Write Operation .....	16
Example Write .....	17
Tag Lock Operation .....	18
Tag Kill Operation .....	21
RFID_TAG_CACHE – Tag Cache Operations .....	23
Example Read Cache Size .....	23
Example Clear Tag Cache .....	23

# List of Tables

<b>About This Document .....</b>	<b>6</b>
<b>Host API for RFID .....</b>	<b>8</b>
RSM Attributes .....	9
Example Read - Read Entire EPC Bank .....	12
Example Read - Read PC Word .....	14
Example Write .....	17
Example Tag Lock Operation .....	20
Example - Tag Kill Operation .....	22
Example - Read Cache Size .....	23
Example - Clear Tag Cache .....	23

# About This Document

## Introduction

This document describes the host Application Program Interface (API) for the advanced reading and writing of Radio Frequency Identification (RFID) tags using the Durango RFID Module with the DS9808 Sierra digital scanner and the DS9908R (DS9908 with RFID) imager.

## Notational Conventions

The following conventions are used in this document:

- The **Consolas** font is used to denote code.
- **Bold** text is used to highlight the following:
  - Dialog box, window and screen names
  - Drop-down list and list box names
  - Check box and radio button names
  - Icons on a screen
  - 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.

## Related Documents and Software

The following documents provide more information about the DS9X08 scanners.

- DS9808 Digital Scanner Product Reference Guide, part number 72E-112999-xx
- DS9808R Digital Scanner Product Reference Guide Supplement, part number 72E-132822-xx
- DS9908 Hands-Free Imaging Scanner Product Reference Guide, part number MN-003185--xx.
- DS9908R Hands-Free Imaging Scanner Product Reference Guide Supplement, part number MN-003377-xx.
- Zebra Scanner SDK  
[www.zebra.com/us/en/products/software/scanning-systems/scanner-drivers-and-utilities.html](http://www.zebra.com/us/en/products/software/scanning-systems/scanner-drivers-and-utilities.html)
- EPC Tag Data Standard Standard version 1.11  
[www.gs1.org/standards/epcrfid-epcis-id-keys/epc-rfid-tds/1-11](http://www.gs1.org/standards/epcrfid-epcis-id-keys/epc-rfid-tds/1-11)

For the latest version of this guide and all guides, go to: [zebra.com/support](http://zebra.com/support).

## Service Information

If you have a problem with your equipment, contact Zebra Global Customer Support for your region. Contact information is available at: [zebra.com/support](http://zebra.com/support).

When contacting support, please have the following information available:

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

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

If your problem cannot be solved by Zebra Customer 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 Zebra business product from a Zebra business partner, contact that business partner for support.

## Provide Documentation Feedback

If you have comments, questions, or suggestions about this guide, send an email to [EVM-Techdocs@zebra.com](mailto:EVM-Techdocs@zebra.com).

# Host API for RFID

## Overview

### **DS9808R Corded Hybrid Imager with RFID**

The DS9808R is the first combination 1D/2D barcode scanner and RFID reader and the first combination hand-held/hands-free UHF RFID reader. The DS9808R offers the flexibility to accommodate virtually any type of data capture application, including support for: 1D and 2D paper and mobile barcodes, mobile coupons, mobile loyalty cards, mobile boarding passes, PDF417 barcodes, and reading/writing RFID tags.

The read range for the RFID antenna can be adjusted to suit the environment. For example, the RFID read range in presentation mode can be decreased to prevent inadvertent reading of RFID tags in a nearby cash-wrap, while the range in hand-held (triggered) mode can be increased to read RFID tags that are further away.

### **DS9908R Next Generation Corded Hybrid Imager with RFID**

The DS9908R replaces the DS9808R. The DS9908R offers several enhanced features that includes the following.

- A capacitive touch sensor and accelerometer that work together to instantly switch to hand-held mode as soon as an associate picks up the scanner. There are no mechanical components to wear out, providing fail-proof switching throughout the scanner's life.
- The reader operates with just 5 VDC and does not require an external power supply.
- Built-in RFID data conversion software reports the tags EPC data as:
  - Raw - the EPC buffer transmits as a hexadecimal string. This format can include the protocol control bytes.
  - GS1-128 - for GS1 encoded tags. This option converts EPC data to a GS1-128 barcode as per the GS1 EPC Tag Data Standard.
  - EPC URI - the EPC data is represented as a Universal Resource Identifier (URI) as defined in the EPC Tag Data Standard version 1.11.
- Zebra's ScanSpeed Analytics provides detailed performance metrics on each barcode captured - enabling you to identify and eliminate poor performing barcodes that slow down the checkout process. You can also view the number of decodes performed in hand-held and hands-free modes to better understand how associates are using the scanner.



## Setup

### Using with USB OPOS

To use the RFID control with the Zebra Scanner SDK you may use the Scanner WMI Sample Application (Scanner\_WMI\_test.exe). Refer to Zebra Scanner SDK for Windows at: [www.zebra.com/scannersdkforwindows](http://www.zebra.com/scannersdkforwindows).

This program offers the “SetAttributes” and “GetAttributes” methods which are needed to access the RFID attributes. All examples in this document assume you are using this sample application.

The DS9X08R scanner must have a USB cable connection and should be configured with the USB Device Type set to one of the following:

- USB OPOS Handheld
- IBM Handheld USB
- IBM Table Top USB

Refer to the scanner Product Reference Guides for details and configuration barcodes (see [Related Documents and Software on page 7](#)).

## RSM Attributes for RFID

The RFID API is expressed in a series of RSM attributes.

**Table 1** RSM Attributes

Attribute Number	Attribute Name	RSM Type	Size (Bytes)	Access	Description
35001	RFID_LAST_TAG_ID	'A'	34	R	The EPC Tag ID of the last tag reported. (size-encoded binary)
35002	RFID_TAG_ID	'A'	34	W	The EPC Tag ID of the tag to be operated upon. (size-encoded binary)
35003	RFID_BANK	'B'	1	W	Desired Tag Bank: 0 = reserved, 1 = EPC, 2 = TID, 3 = User
35004	RFID_DATA	'A'	66	RW	Buffer for read, write, and lock (size-encoded binary)
35005	RFID_OFFSET	'W'	2	W	Word offset into tag buffer
35006	RFID_LENGTH	'W'	2	W	Words of data to read from tag buffer. 0 means entire bank
35007	RFID_PASSWORD	'A'	4	W	Binary password for privileged operations

**Table 1** RSM Attributes (Continued)

Attribute Number	Attribute Name	RSM Type	Size (Bytes)	Access	Description
35008	RFID_COMMAND	'B'	1	W	Execute command: 1 = Read 2 = Write 3 = Lock 4 = Kill
35009	RFID_CMD_STATUS	'W'	2	R	Resulting status from executing a command
35010	RFID_TAG_CACHE	'W'	2	RW	Internal Tag Cache Size: Read for current cache size Write 0 to clear the cache

For size-encoded binary data, the first two bytes contain the length (MSB, LSB) for the data to be considered (needed because the RSM attributes are fixed size).

### RFID\_LAST\_TAG\_ID – Last Reported Tag

As a convenience, the `RFID_LAST_TAD_ID` may be used to get the raw EPC of the last tag reported by normal tag reading operation of the D9x08R (normally for an `RFID_READ` event, the scanner issues a 2-tone beep).

As an example, to get the `RFID_LAST_TAD_ID` attribute after the scanner reports the tag `3005FB63AC1F3681EC880469`.

Method	Input	attValueList
GetAttributes	<attrib_list>35001</attrib_list>	<attrib_list> <attribute> <id>35001</id> <datatype>A</datatype> <permission>R</permission> <value>0x00 0x0c 0x30 0x05 0xfb 0x63 0xac 0x1f 0x36 0x81 0xec 0x88 0x04 0x69 0x00 </attribute> </attrib_list>

In raw RSM protocol:

Send: 00 08 02 00 88 B9 FF FF

```
Recv: 00 33 02 00 88 B9 41 01 42 00 22 00 00 00 0C 30 05 FB 63 AC 1F 36 81 EC 88 04 69  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
```

## RFID\_CMD\_STATUS - Common Operation Response

All operations (read, write, kill, lock) return a status in the RFID\_CMD\_STATUS attribute:

0x0000	Success
0x0001	No RFID module
0x0002	Tag Not Found
0x0003	Timeout
0x0004	Tag CRC Error
0x01xx	Tag Backscatter Error, LSB indicates the error_code as per EPC Protocol
0x02xx	Tag Access error. LSB indicates the error code
0x03xx	Bad Parameter, the LSB indicates which parameter: 1 = Command 2 = Tag_ID 3 = Bank 4 = Data 5 = Offset 6 = Password 7 = Length

## RFID\_COMMAND

The RFID\_COMMAND attribute is used to execute the various tag operations. Each operation has parameter attributes that should be setup prior to executing the command.

### Tag Read Operation

The Tag Read operation requires the following attributes to be set using RSM SetAttributes:

Attribute Number	Attribute	Description
35002	RFID_TAG_ID	EPC code of desired tag.
35003	RFID_BANK	Desired memory bank of tag.
35005	RFID_OFFSET	Word offset into the memory bank.
35006	RFID_LENGTH	Number of words to read.
35007	RFID_PASSWORD	Optional access password.
35008	RFID_COMMAND	1 = Read.

The Tag Read operation is activated by the SetAttribute of the RFID\_COMMAND attribute.

The result of the operation may be retrieved by the RSM GetAttributes of the following attributes. If the RFID\_CMD\_STATUS attribute indicates success, then the RFID\_DATA attribute will have the requested data.

Attribute Number	Attribute	Description
35009	RFID_CMD_STATUS	Resulting status from the read.
35004	RFID_DATA	Buffer for read.

### Example Read

Table 2 is an example to read the entire EPC bank of the tag with EPC of 3005FB63AC1F3681EC880469.

**Table 2** Example Read - Read Entire EPC Bank

Method	Input	attValueList
SetAttributes	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;!--RFIDTagID--&gt;     &lt;id&gt;35002&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x00 0x0c 0x30 0x05 0xFB 0x63 0xAC 0x1F 0x36 0x81 0xEC 0x88 0x04 0x69 &lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDBank--&gt;     &lt;id&gt;35003&lt;/id&gt;     &lt;datatype&gt;B&lt;/datatype&gt;     &lt;value&gt;1&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDOffset--&gt;     &lt;id&gt;35005&lt;/id&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;value&gt;0&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDLength--&gt;     &lt;id&gt;35006&lt;/id&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;value&gt;0&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDCommand--&gt;     &lt;id&gt;35008&lt;/id&gt;     &lt;datatype&gt;B&lt;/datatype&gt;     &lt;value&gt;1&lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>	n/a

**Table 2** Example Read - Read Entire EPC Bank (Continued)

Method	Input	attValueList
GetAttributes	<attrib_list>35009,35004</attrib_list>	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;id&gt;35009&lt;/id&gt;     &lt;name&gt;"RFIDCmdStatus"&lt;/name&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;permission&gt;R&lt;/permission&gt;     &lt;value&gt;0&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;id&gt;35004&lt;/id&gt;     &lt;name&gt;"RFIDData"&lt;/name&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;permission&gt;R W P&lt;/permission&gt;     &lt;value&gt;0x00 0x10 0xA0 0x4B 0x30 0x00 0x30 0x05 0xFB 0x63 0xAC 0x1F 0x36 0x81 0xEC 0x88 0x04 0x69 0x00 &lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>

In raw RSM protocol:

Send: 00 33 05 00 88 BA 41 00 42 00 0E 00 00 00 0C 30 05 FB 63 AC 1F 36 81 EC 88 04 69  
88 BB 42 00 01 88 BD 57 00 00 00 88 BE 57 00 00 00 88 C0 42 00 01 FF FF

Recv: 00 04 05 00

Send: 00 0A 02 00 88 C1 88 BC FF FF

Recv: 00 59 02 00 88 C1 57 01 00 00 88 BC 41 03 42 00 42 00 00 00 10 AE 57 30 00 30 05  
FB 63 AC 1F 36 81 EC 88 04 69 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00  
FF FF FF FF

Table 3 is a second example to read the PC word (second word, whose value is 0x3000) from the EPC buffer of the same tag:

**Table 3** Example Read - Read PC Word

Method	Input	attValueList
SetAttributes	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;!--RFIDTagID--&gt;     &lt;id&gt;35002&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x00 0x0c 0x30 0x05 0xFB 0x63 0xAC 0x1F 0x36 0x81 0xEC 0x88 0x04 0x69 &lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDBank--&gt;     &lt;id&gt;35003&lt;/id&gt;     &lt;datatype&gt;B&lt;/datatype&gt;     &lt;value&gt;1&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDOffset--&gt;     &lt;id&gt;35005&lt;/id&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;value&gt;1&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDLength--&gt;     &lt;id&gt;35006&lt;/id&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;value&gt;1&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDCommand--&gt;     &lt;id&gt;35008&lt;/id&gt;     &lt;datatype&gt;B&lt;/datatype&gt;     &lt;value&gt;1&lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>	n/a

**Table 3** Example Read - Read PC Word (Continued)

Method	Input	attValueList
GetAttributes	<attrib_list>35009,35004</attrib_list>	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;id&gt;35009&lt;/id&gt;      &lt;name&gt;"RFIDCmdStatus"&lt;/name&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;permission&gt;R&lt;/permission&gt;     &lt;value&gt;0&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;id&gt;35004&lt;/id&gt;     &lt;name&gt;"RFIDData"&lt;/name&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;permission&gt;R W P&lt;/permission&gt;     &lt;value&gt;0x00 0x02 0x30 0x00 &lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>

In raw RSM protocol:

Send: 00 33 05 00 88 BA 41 00 42 00 0E 00 00 00 0C 30 05 FB 63 AC 1F 36 81 EC 88 04 69  
88 BB 42 00 01 88 BD 57 00 00 01 88 BE 57 00 00 01 88 C0 42 00 01 FF FF

Recv: 00 04 05 00

Send: 00 0A 02 00 88 C1 88 BC FF FF

Recv: 00 59 02 00 88 C1 57 01 00 00 88 BC 41 03 42 00 42 00 00 00 02 30 00 00 00 00 00  
00  
00  
FF FF FF FF

## Tag Write Operation

The Tag Write operation requires the following attributes to be set using RSM SetAttributes:

Attribute Number	Attribute	Description
35002	RFID_TAG_ID	EPC code of desired tag
35003	RFID_BANK	Desired memory bank of tag
35004	RFID_DATA	data to write
35005	RFID_OFFSET	Word offset into the memory bank
35007	RFID_PASSWORD	Optional access password
35008	RFID_COMMAND	2 = "write"

The Tag Write operation is activated by the SetAttribute of the RFID\_COMMAND attribute.

The result of the operation may be retrieved by the RSM GetAttributes of the RFID\_CMD\_STATUS attribute.

Attribute Number	Attribute	Description
35009	RFID_CMD_STATUS	Resulting status from the Write



## Example Write

Table 4 is an example to write 0x1234 into the second word of the user bank of the tag with EPC of 3005FB63AC1F3681EC880469.

**Table 4** Example Write

Method	Input	attValueList
SetAttributes	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;!--RFIDTagID--&gt;     &lt;id&gt;35002&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x00 0x0c 0x30 0x05 0xFB 0x63 0xAC 0x1F 0x36 0x81 0xEC 0x88 0x04 0x69 &lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDBank--&gt;     &lt;id&gt;35003&lt;/id&gt;     &lt;datatype&gt;B&lt;/datatype&gt;     &lt;value&gt;3&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDOffset--&gt;     &lt;id&gt;35005&lt;/id&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;value&gt;1&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDData--&gt;     &lt;id&gt;35004&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x00 0x02 0x12 0x34 &lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDCommand--&gt;     &lt;id&gt;35008&lt;/id&gt;     &lt;datatype&gt;B&lt;/datatype&gt;     &lt;value&gt;2&lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>	n/a

**Table 4** Example Write

Method	Input	attValueList
GetAttributes	<attrib_list>35009</attrib_list>	<attrib_list> <attribute> <id>35009</id> <name>"RFIDCmdStatus"</name> <datatype>W</datatype> <permission>R</permission> <value>0</value> </attribute> </attrib_list>

In raw RSM Protocol:

Send: 00 33 05 00 88 BA 41 00 42 00 0E 00 00 00 0C 30 05 FB 63 AC 1F 36 81 EC 88 04 69  
88 BB 42 00 03 88 BD 57 00 00 01 88 BC 41 00 42 00 04 00 00 00 02 12 34 88 C0 42 00 02  
FF FF

Recv: 00 04 05 00

Send: 00 08 02 00 88 C1 FF FF

Recv: 00 0E 02 00 88 C1 57 01 00 00 FF FF FF FF

## Tag Lock Operation

The Tag Lock operation requires the following attributes to be set using RSM SetAttributes:

Attribute Number	Attribute	Description
35002	RFID_TAG_ID	EPC code of desired tag
35004	RFID_DATA	Lock configuration
35007	RFID_PASSWORD	Access password (required)
35008	RFID_COMMAND	3 = "Lock"

The Lock Configuration is 4 bytes, defined in the EPC Protocol Spec as:

### Lock-Command Payload

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Kill Mask		Access Mask		EPC Mask		TID Mask		User Mask		Kill Action		Access Action		EPC Action		TID Action		User Action	

### Masks and Associated Action Fields

	Kill pwd		Access pwd		EPC memory		TID memory		User memory	
	19	18	17	16	15	14	13	12	11	10
Mask	skip/ write	skip/ write	skip/ write	skip/ write	skip/ write	skip/ write	skip/ write	skip/ write	skip/ write	skip/ write
	9	8	7	6	5	4	3	2	1	0
Action	pwd read/ write	perma lock	pwd read/ write	perma lock	pwd write	perma lock	pwd write	perma lock	pwd write	perma lock

pwd-write	permalock	Description
0	0	Associated memory bank is writeable from either the <b>open</b> or <b>secured</b> states.
0	1	Associated memory bank is permanently writeable from either the <b>open</b> or <b>secured</b> states and may never be locked.
1	0	Associated memory bank is writeable from the <b>secured</b> state but not from the <b>open</b> state.
1	1	Associated memory bank is not writeable from any state.
pwd-read/write	permalock	Description
0	0	Associated password location is readable and writeable from either the <b>open</b> or <b>secured</b> states.
0	1	Associated password location is permanently readable and writeable from either the <b>open</b> or <b>secured</b> states and may never be locked.
1	0	Associated password location is readable and writeable from the <b>secured</b> state but not from the <b>open</b> state.
1	1	Associated password location is not readable or writeable from any state.

The password is required for the Tag Lock operation and must match the access password of the tag (bytes 4-7 of the reserved bank).

The Tag Lock operation is activated by the SetAttribute of the RFID\_COMMAND attribute.

The result of the operation may be retrieved by the RSM GetAttributes of the RFID\_CMD\_STATUS attribute.

Attribute Number	Attribute	Description
35009	RFID_CMD_STATUS	Resulting status from the Kill

## Example Tag Lock Operation

Table 5 is an example to hide the kill password (bytes 0-3 of the reserved bank) of the tag with EPC of 3005FB63AC1F3681EC880469, assuming the access password (bytes 4-7 of the reserved bank) is 0x87654321.

To setup the lock configuration, set:

“kill pwd” mask = 11

“kill pwd” action = 10

Therefore the lock configuration is 0x000c0200.

**Table 5** Example Tag Lock Operation

Method	Input	attValueList
SetAttributes	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;!--RFIDTagID--&gt;     &lt;id&gt;35002&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x00 0x0c 0x30 0x05 0xFB 0x63 0xAC 0x1F 0x36 0x81 0xEC 0x88 0x04 0x69 &lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDData--&gt;     &lt;id&gt;35004&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x00 0x04 0x00 0x0c 0x02 0x00 &lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDPassword--&gt;     &lt;id&gt;35007&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x87 0x65 0x43 0x21&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDCommand--&gt;     &lt;id&gt;35008&lt;/id&gt;     &lt;datatype&gt;B&lt;/datatype&gt;     &lt;value&gt;3&lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>	n/a
GetAttributes	<pre> &lt;attrib_list&gt;35009&lt;/attrib_list&gt; </pre>	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;id&gt;35009&lt;/id&gt;      &lt;name&gt;"RFIDCmdStatus"&lt;/name&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;permission&gt;R&lt;/permission&gt;     &lt;value&gt;0&lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>

## Tag Kill Operation

The Tag Kill operation requires the following attributes to be set using RSM SetAttributes:

Attribute Number	Attribute	Description
35002	RFID_TAG_ID	EPC code of desired tag
35007	RFID_PASSWORD	Kill password (required non-zero)
35008	RFID_COMMAND	4 = "kill"

The password is required for the Tag Kill operation and must match the kill password of the tag (bytes 0-3 of the reserved bank). Note that, as per the EPC protocol spec, if the kill password is zero the tag cannot be killed.

The Tag Kill operation is activated by the SetAttribute of the RFID\_COMMAND attribute.

The result of the operation may be retrieved by the RSM GetAttributes of the RFID\_CMD\_STATUS attribute.

Attribute Number	Attribute	Description
35009	RFID_CMD_STATUS	Resulting status from the Kill

### Example Tag Kill Operation

[Table 6](#) is an example to kill the tag with EPC of 3005FB63AC1F3681EC880469, assuming the kill password (bytes 0-3 of the reserved bank) is 0x12345678.

**Table 6** Example - Tag Kill Operation

Method	Input	attValueList
SetAttributes	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;!--RFIDTagID--&gt;     &lt;id&gt;35002&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x00 0x0c 0x30 0x05 0xFB 0x63 0xAC 0x1F 0x36 0x81 0xEC 0x88 0x04 0x69 &lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDData--&gt;     &lt;id&gt;35004&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x00 0x04 0x00 0x0c 0x02 0x00 &lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDPassword--&gt;     &lt;id&gt;35007&lt;/id&gt;     &lt;datatype&gt;A&lt;/datatype&gt;     &lt;value&gt;0x12 0x34 0x56 0x78&lt;/value&gt;   &lt;/attribute&gt;   &lt;attribute&gt;     &lt;!--RFIDCommand--&gt;     &lt;id&gt;35008&lt;/id&gt;     &lt;datatype&gt;B&lt;/datatype&gt;     &lt;value&gt;4&lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>	n/a
GetAttributes	<pre> &lt;attrib_list&gt;35009&lt;/attrib_list&gt; </pre>	<pre> &lt;attrib_list&gt;   &lt;attribute&gt;     &lt;id&gt;35009&lt;/id&gt;     &lt;name&gt;"RFIDCmdStatus"&lt;/name&gt;     &lt;datatype&gt;W&lt;/datatype&gt;     &lt;permission&gt;R&lt;/permission&gt;     &lt;value&gt;0&lt;/value&gt;   &lt;/attribute&gt; &lt;/attrib_list&gt; </pre>

## RFID\_TAG\_CACHE – Tag Cache Operations

As a diagnostic tool, the RFID\_TAG\_CACHE may be used to read the current tag cache size or to flush the tag cache. Reading this attribute returns the current number of the unique RFID tags in the cache. Writing any value to this attribute causes the cache to be cleared (flushed).



**NOTE:** Use caution when clearing the tag cache as all tags in range will be read on the very next inventory. If automatic reading is enabled, this will be immediately.

### Example Read Cache Size

**Table 7** Example - Read Cache Size

Method	Input	attValueList
GetAttributes	<attrib_list>35010</attrib_list>	<attrib_list> <attribute> <id>35010</id> <name>""</name> <datatype>W</datatype> <permission>R W</permission> <value>2</value> </attribute> </attrib_list>

In raw RSM protocol:

Send: 00 08 02 00 88 C2 FF FF

Recv: 00 0E 02 00 88 C2 57 03 00 02 FF FF FF FF

### Example Clear Tag Cache

**Table 8** Example - Clear Tag Cache

Method	Input	attValueList
SetAttributes	<attrib_list> <attribute> <id>35010</id> <datatype>W</datatype> <value>0</value> </attribute> </attrib_list>	

In raw RSM protocol:

Send: 00 0A 05 00 88 C2 57 03 00 00

Recv: 00 04 05 00

