Dynamag, DynaMAX, eDynamo, mDynamo, tDynamo, DynaWave, and iDynamo 6

Secure Card Reader Authenticators
Programmer's Reference (Java and Java Applet)

August 2020

Manual Part Number:
D99875724-102

REGISTERED TO ISO 9001:2015

Table 0.1 – Revisions

Rev Number	Date	Notes
1.01	09/29/2014	Initial release
1.02	10/09/2014	Add getBatteryLevel and sendData, getSwipeCount
20	1/28/2015	Update System Requirements. Fix typos on some functions. Update openDevice function to include deviceURI for BLE.
30	2/6/2015	Update instructions for section How to Set Up the Swipe Reader Control Panel.
40	05/06/2015	Add instructions on how to modify manifest and sign JAR.
50	02/17/2016	Add eDynamo. Add function getTLVPayload, getCardServiceCode.
60	05/17/2016	Added DynaPro format for EMV transaction messages.
70	10/28/2016	Added support for mDynamo.
80	08/22/2018	Updated SDK Contents and tested Java Runtime environments. Added support for tDynamo and DynaWave.
90	01/21/2019	Updated to correctly reference Bluetooth LE.

100	10/08/2019	Updated events for the event onTransactionStatus(), and result codes for the event onDeviceExtendedResponse(). Updated the function startTransaction(): cardType, option, and transactionType.
101	11/11/2019	Added the format of card data returned from GetCardData().
102	08/10/2020	Updated the Java Applet Sample and Java Sample Software information in section 1 and 2.

SOFTWARE LICENSE AGREEMENT

IMPORTANT: YOU SHOULD CAREFULLY READ ALL THE TERMS, CONDITIONS AND RESTRICTIONS OF THIS LICENSE AGREEMENT BEFORE INSTALLING THE SOFTWARE PACKAGE. YOUR INSTALLATION OF THE SOFTWARE PACKAGE PRESUMES YOUR ACCEPTANCE OF THE TERMS, CONDITIONS, AND RESTRICTIONS CONTAINED IN THIS AGREEMENT. IF YOU DO NOT AGREE WITH THESE TERMS, CONDITIONS, AND RESTRICTIONS, PROMPTLY RETURN THE SOFTWARE PACKAGE AND ASSOCIATED DOCUMENTATION TO THE ADDRESS ON THE FRONT PAGE OF THIS DOCUMENT, ATTENTION: CUSTOMER SUPPORT.

TERMS, CONDITIONS, AND RESTRICTIONS

MagTek, Incorporated (the "Licensor") owns and has the right to distribute the described software and documentation, collectively referred to as the "Software."

LICENSE: Licensor grants you (the "Licensee") the right to use the Software in conjunction with MagTek products. LICENSEE MAY NOT COPY, MODIFY, OR TRANSFER THE SOFTWARE IN WHOLE OR IN PART EXCEPT AS EXPRESSLY PROVIDED IN THIS AGREEMENT. Licensee may not decompile, disassemble, or in any other manner attempt to reverse engineer the Software. Licensee shall not tamper with, bypass, or alter any security features of the software or attempt to do so.

TRANSFER: Licensee may not transfer the Software or license to the Software to another party without the prior written authorization of the Licensor. If Licensee transfers the Software without authorization, all rights granted under this Agreement are automatically terminated.

COPYRIGHT: The Software is copyrighted. Licensee may not copy the Software except for archival purposes or to load for execution purposes. All other copies of the Software are in violation of this Agreement.

TERM: This Agreement is in effect as long as Licensee continues the use of the Software. The Licensor also reserves the right to terminate this Agreement if Licensee fails to comply with any of the terms, conditions, or restrictions contained herein. Should Licensor terminate this Agreement due to Licensee's failure to comply, Licensee agrees to return the Software to Licensor. Receipt of returned Software by the Licensor shall mark the termination.

LIMITED WARRANTY: Licensor warrants to the Licensee that the disk(s) or other media on which the Software is recorded are free from defects in material or workmanship under normal use.

THE SOFTWARE IS PROVIDED AS IS. LICENSOR MAKES NO OTHER WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Because of the diversity of conditions and PC hardware under which the Software may be used, Licensor does not warrant that the Software will meet Licensee specifications or that the operation of the Software will be uninterrupted or free of errors.

IN NO EVENT WILL LICENSOR BE LIABLE FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE, THE SOFTWARE. Licensee's sole remedy in the event of a defect in material or workmanship is expressly limited to replacement of the Software disk(s) if applicable.

GOVERNING LAW: If any provision of this Agreement is found to be unlawful, void, or unenforceable, that provision shall be removed from consideration under this Agreement and will not affect the enforceability of any of the remaining provisions. This Agreement shall be governed by the laws of the State of California and shall inure to the benefit of MagTek, Incorporated, its successors or assigns.

ACKNOWLEDGMENT: LICENSEE ACKNOWLEDGES THAT HE HAS READ THIS AGREEMENT, UNDERSTANDS ALL OF ITS TERMS, CONDITIONS, AND RESTRICTIONS, AND AGREES TO BE BOUND BY THEM. LICENSEE ALSO AGREES THAT THIS AGREEMENT SUPERSEDES ANY AND ALL VERBAL AND WRITTEN COMMUNICATIONS BETWEEN LICENSOR AND LICENSEE OR THEIR ASSIGNS RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

QUESTIONS REGARDING THIS AGREEMENT SHOULD BE ADDRESSED IN WRITING TO MAGTEK, INCORPORATED, ATTENTION: CUSTOMER SUPPORT, AT THE ADDRESS LISTED IN THIS DOCUMENT, OR E-MAILED TO SUPPORT@MAGTEK.COM.

Table of Contents

1 Introduction

This document provides instructions for software developers who want to create software solutions that include a Dynamag, DynaMAX, eDynamo, mDynamo, tDynamo, DynaWave, or iDynamo 6 connected to a Windows-based host via USB or Bluetooth LE. It is part of a larger library of documents designed to assist Dynamag, DynaMAX, eDynamo, mDynamo, tDynamo, DynaWave, and iDynamo 6 implementers, which includes the following documents available from MagTek:

- ***D99875724 Dynamag, DynaMAX, eDynamo, mDynamo, tDynamo, DynaWave, and iDynamo 6 Programmer's Manual (Java and Java Applet)***
- ***D99875725 Dynamag, DynaMAX, eDynamo, mDynamo, tDynamo, DynaWave, and iDynamo 6 Programmer’s Manual (C++)***
- ***D99875475 MagneSafe V5 Programmer’s Manual***

1.1 About the Java Sample Code

The Java Sample software, available from MagTek, provides demonstration source code and a reusable MTSCRA Java library that provides developers of custom Java software solutions with an easy-to-use interface for Dynamag, DynaMAX, eDynamo, mDynamo, tDynamo, DynaWave, and iDynamo 6. Developers can include the MTSCRA Java library in custom branded software which can be distributed to customers or distributed internally as part of an enterprise solution.

1.2 About the Read & Parse Card Data

The Java applet, available from MagTek, provides demonstration source code and a reusable Java applet that provides developers of custom HTML / JavaScript software solutions with a set of functions that parallels the functionality to the Java library, in applet form.

1.3 Nomenclature

The general terms “device” and “host” are used in different, often incompatible ways in a multitude of specifications and contexts. For example, “host” may have different meanings in the context of USB communication than it does in the context of networked financial transaction processing. In this document, “device” and “host” are used strictly as follows:

- **Device** refers to the MSR device that receives and responds to the command set specified in this document; in this case, Dynamag, DynaMAX, eDynamo, mDynamo, tDynamo, DynaWave, or iDynamo 6.
- **Host** refers to the piece of general-purpose electronic equipment the device is connected or paired to, which can send data to and receive data from the device. Host types include PC and Mac computers/laptops, tablets, smartphones, teletype terminals, and even test harnesses. In many cases the host may have custom software installed on it that communicates with the device. When “host” must be used differently, it is qualified as something specific, such as “USB host.”

The word “user” is also often used in different ways in different contexts. In this document, **user** generally refers to the **cardholder**.

1.4 SDK Contents

File name	Description
JavaSample.java	Java sample code.
JavaSample.jar	This is the JavaSample.java compiled version

File name	Description
mtscra.jar	This is the MTSCRA Java library and Java applet library
MTSCRA.dll	This is a native DLL to be copied to the system folder.
MTSCRAJ.dll	This is a native DLL to be copied to the system folder.
MTSCRABLE.dll	DLL require to interact with Bluetooth LE device. For Bluetooth LE to work, Windows 8 or above is required.
Build.bat	This .bat file builds the Java Sample software.
Test.bat	This.bat file launches the Java Sample software.
MTSCRASample.html	This sample web page demonstrates how to use the applet.

1.5 System Requirements

1.5.1 Java Library

Tested operating systems:

- Windows 7
- Windows 8, 8.1
- Windows 10

Java Build Platform: JDK 1.8 32-bit

Minimum Java Runtime requirements: Java 8

Tested Java Runtime Environments: Java 8

1.5.2 Java Applet

Tested operating systems:

- Windows 7
- Windows 8, 8.1
- Windows 10

Tested web browsers:

- Internet Explorer 11
- Firefox 52 ESR and above

Minimum Java Runtime requirements: Java 8

Tested Java Runtime Environments: Java 8

1.6 Interfaces for Operating Systems

The following table matches the device interface to operating system.

Device	Interface	Operating System
Dynamag	USB	Windows 7, Windows 8 & 8.1, Windows 10
DynaMAX	USB	Windows 7, Windows 8 & 8.1, Windows 10
	Bluetooth LE	Windows 8 & 8.1, Windows 10
eDynamo	USB	Windows 7, Windows 8 & 8.1, Windows 10
	Bluetooth LE	Windows 8 & 8.1, Windows 10
mDynamo	USB	Windows 7, Windows 8 & 8.1, Windows 10
tDynamo	USB	Windows 7, Windows 8 and 8.1, Windows 10
DynaWave	USB	Windows 7, Windows 8 and 8.1, Windows 10
iDynamo 6	USB	Windows 7, Windows 8 and 8.1, Windows 10

2 How to Set Up the Java Sample Software

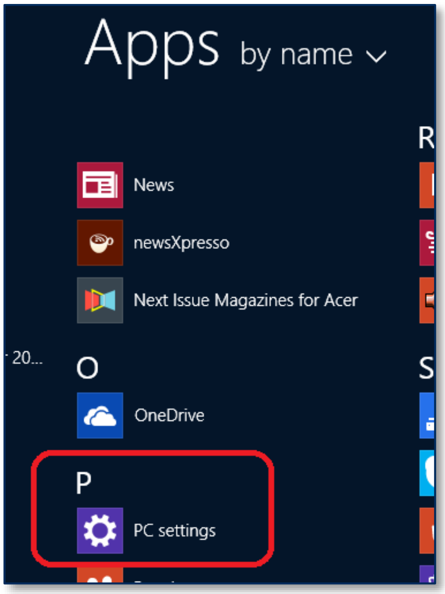
2.1 How to Download and Set Up the Java Sample Software

To set up the MTSCRA Libraries, follow these steps:
1) Download the ***DynaMag, DynaMAX, eDynamo, mDynamo, tDynamo, DynaWave, and iDynamo 6 Secure Card Reader Authenticator Windows API***, available from MagTek.com https://www.magtek.com/Content/SoftwarePackages/99510133.exe
2) Right-click **99510133.exe** and select **Run as administrator**. The installer will place all dependencies in appropriate paths.

2.2 How to Connect DynaMAX or eDynamo to a Windows Host via Bluetooth LE

To connect DynaMAX or eDynamo to a host with Windows 8.1 or higher and Bluetooth 4.0 hardware that supports Bluetooth LE, follow these steps:

1) If you are using an external Bluetooth adapter, install any required drivers and connect it to the host.
2) On the host, install and configure the software you intend to use with DynaMAX or eDynamo:
    a) Make sure the host software is configured to look for the device on the proper connection.
    b) Make sure the host software knows which device(s) it should interface with.
    c) Make sure the host software is configured to properly interpret incoming data from the device. This depends on whether the device is configured to transmit data in GATT format or streaming format emulating a keyboard.
3) Make sure the DynaMAX's batteries are installed and have adequate charge. If using eDynamo, make sure the device has an adequate charge.
4) Test the batteries by powering on the DynaMAX or eDynamo device. Provided the device is not already paired, the Bluetooth Status LED will flash blue every two seconds for up to 60 seconds until pairing is complete. If the Bluetooth Status LED is solid blue, the device is already paired with a host. Unpair from the host it is already paired with before continuing.
5) Enter app mode, scroll down to **Apps by name**, and launch the Windows **PC Settings** app.



6) In the left side navigator, select **PC and devices** > **Bluetooth**.
7) Make sure Bluetooth is turned on and close the **PC and devices** app.
8) Launch the Windows **Manage Bluetooth Devices** app by following these steps:
    a) Enter desktop mode by swiping in from the left side of the touchscreen.
    b) Touch the Bluetooth icon in the system tray and select **Add a Bluetooth Device** (see **Figure 2-1**).

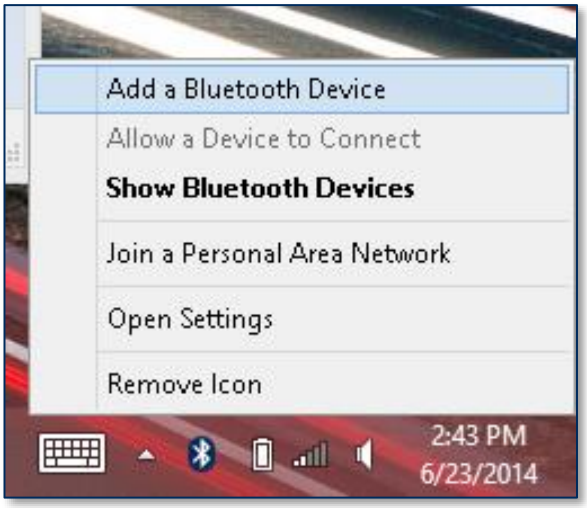
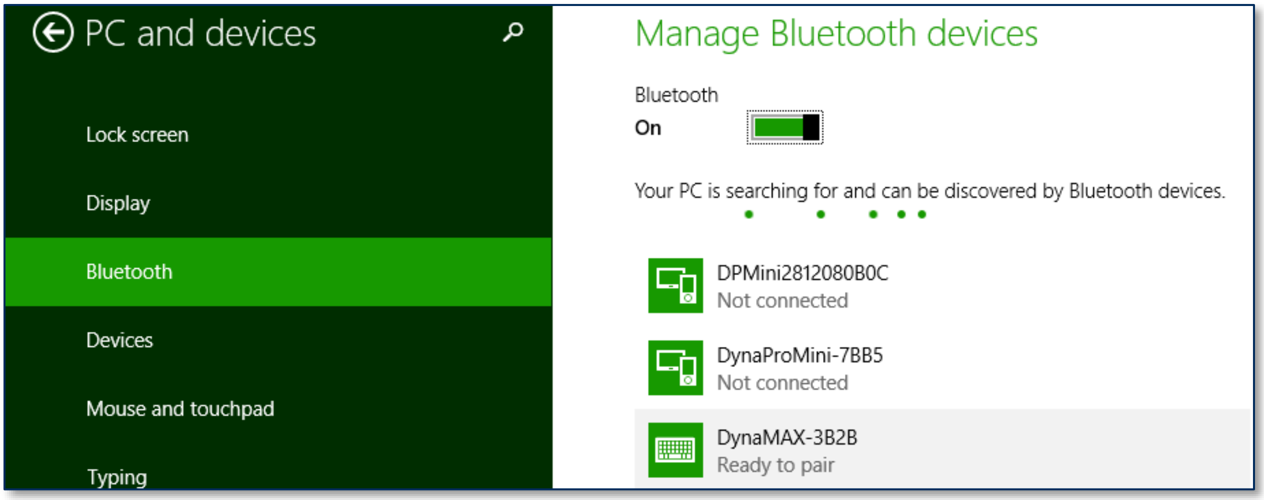


Figure 2-1 - Launch Manage Bluetooth Devices App from Desktop Mode



Bluetooth
On

Your PC is searching for and can be discovered by Bluetooth devices.

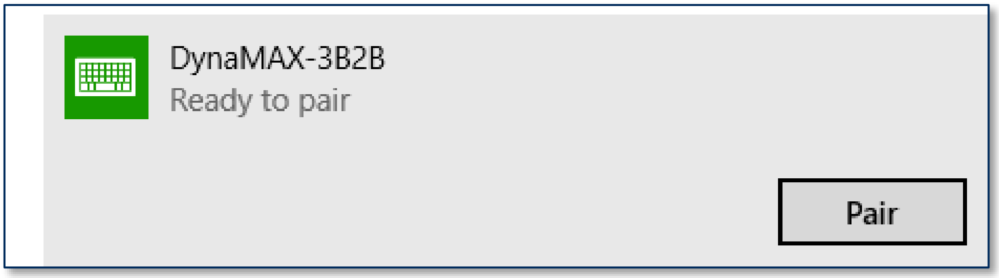
DPMini2812080B0C
Not connected

DynaProMini-7BB5
Not connected

DynaMAX-3B2B
Ready to pair

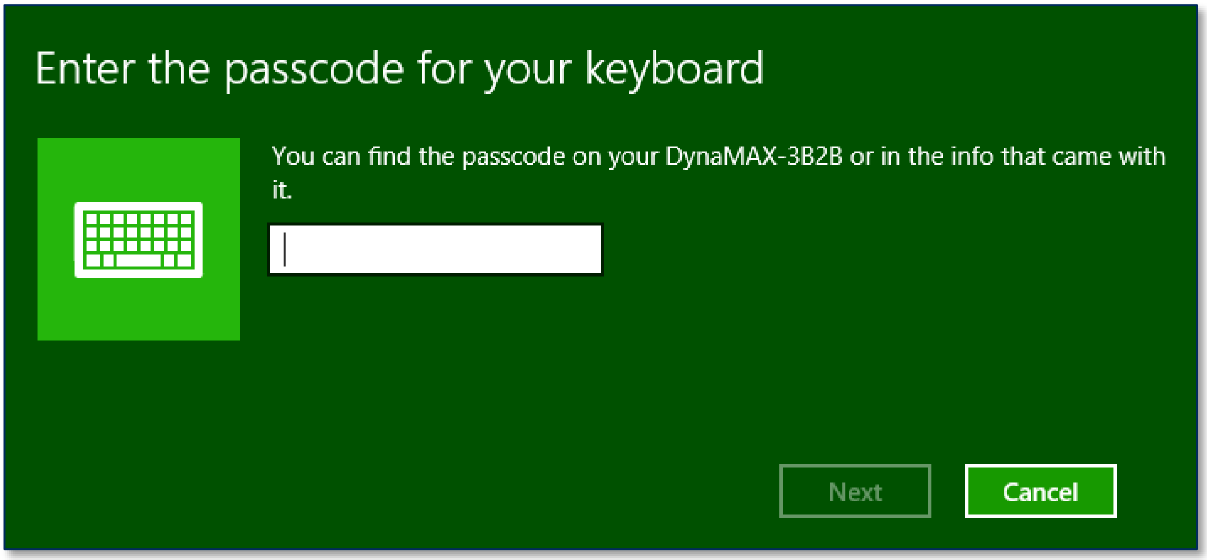
Figure 2-2 – Windows 8 Manage Bluetooth Devices App

9) Locate the serial number on the label on the bottom of the device. Note the final four digits.
10) Read through the list of pairable devices and locate the device called **DynaMAX-nnnn** or **eDynamo-nnn**, where nnnn is the last four digits of the device's serial number (if the device does not show in the list, power it off then power it back on). Below the device name you should see the text **Ready to pair**.
11) Select the device and press the **Pair** button. If the device is configured to run in KB mode, Windows will prompt you **Enter the passcode for your keyboard**.



DynaMAX-3B2B

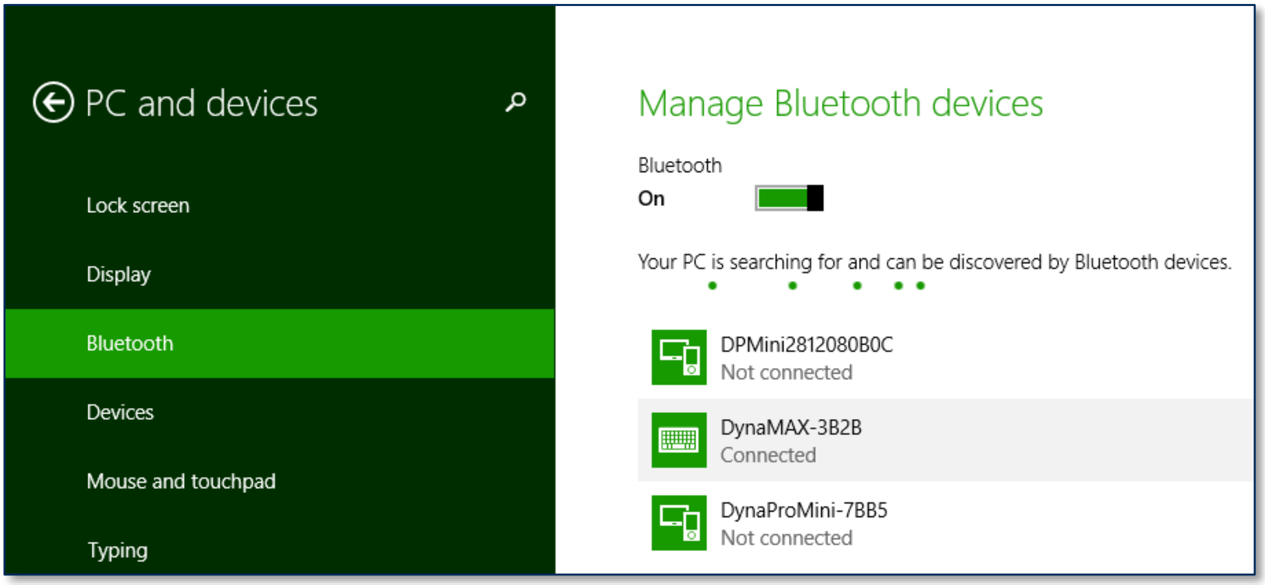
Ready to pair



Enter the passcode for your keyboard

You can find the passcode on your DynaMAX-3B2B or in the info that came with it.

12) Enter default passcode **000000** (or the device's actual password if it has been configured differently), then press the **Next** button. Windows will return you to the **Manage Bluetooth devices** page. After a short period of time, you will see the text **Connected** below the device you are pairing with. After a few seconds the device will disconnect, which is normal power-saving behavior.



PC and devices

- Lock screen
- Display
- Bluetooth
- Devices
- Mouse and touchpad
- Typing

Manage Bluetooth devices

Bluetooth
On

Your PC is searching for and can be discovered by Bluetooth devices.

DPMini2812080B0C
Not connected

DynaMAX-3B2B
Connected

DynaProMini-7BB5
Not connected

13) Use the host software to test swiping a card. If you do not yet have host software and the device is configured to run in KB mode, open any text editor and swipe a card. The card contents should appear in the text editor.
14) The device consumes very little power when not transmitting card data, so it is not necessary to power off the device to conserve power. If the device appears as **Not connected** in the Windows list of Bluetooth devices, swiping a card should cause the device to reconnect briefly, transmit the card data, then disconnect.
15) Remember to change the default password. See the DynaMAX Programmer's Reference documents for details.

To unpair from the device:

1) Locate the device in the **Manage Bluetooth devices** window.
2) Press the **Remove device** button.

2.3 How to Set Up the Java Library With the 32-bit JRE/JVM

MagTek highly recommends using the 32-bit version of Java when using the MTSCRA Java applet, regardless of whether you are using a 32-bit or 64-bit version of Windows.

To set up and run the Java Demo software using the 32-bit version of Java on either a 32-bit or 64-bit version of Windows, follow these steps:

1) Uninstall any existing instances of the 64-bit Java Runtime Environment (JRE) or Java Development Kit (JDK). Leaving them installed can cause runtime failures, as the library may fail to load.
2) Download and install the latest version of the 32-bit Java Development Kit (JDK).
3) Follow the steps in section **2.1 How to Download and Set Up the Java Sample Software** to download and install the latest Dynamag/DynaMAX/eDynamo/mDynamo SCRA Windows SDK. You may download and install it directly on the target workstation where it will be used, or you may opt to install it on a master development workstation and copy the dependencies to the target workstation manually.

4) If you opted to manually copy the Dynamag/DynaMAX/eDynamo/mDynamo SCRA Windows SDK dependencies from a master development workstation to the target workstation where it will be used, follow these steps:
   a) On the master workstation, navigate to the root of the Windows SDK. By default, it will be **C:\Program Files\MagTek\SCRA Windows SDK\Library\Java** for 32-bit operating systems, or **C:\Program Files (x86)\MagTek\SCRA Windows SDK\Library\Java** for 64-bit operating systems.
   b) Open the **\Win32** subfolder and copy all the files to the target workstation's **C:\Windows\System32** folder for x86 systems, or to the target workstation's **C:\Windows\SysWOW64** folder for x64 systems.
5) Connect the device to the workstation. Windows will install the device drivers automatically. Wait for Windows to report the driver installation is complete.
6) Launch a Windows command prompt as an Administrator.
7) **cd** to the root of the folders where the Swipe Reader Control Panel Demo is installed. By default, it will be **C:\Program Files\MagTek\SCRA Windows SDK\Sample Code\Java\Object** for 32-bit Windows, or **C:\Program Files (x86)\MagTek\SCRA Windows SDK\Sample Code\Java\Object** for 64-bit Windows.
8) Type **build.bat** and press **Enter** to build the Java Demo software.
9) Type **test.bat** and press **Enter** to launch the Java Demo software.
10) Use the Java Demo software, and / or continue to the setup steps in section **2.5 How to Set Up the Applet With the 32-bit JRE/JVM**.

2.4 How to Manually Set Up the Java Library With the 64-bit JRE/JVM

MagTek highly recommends using the 32-bit version of Java if you intend to use the MTSCRA Java applet as described in section 2.3, regardless of whether you are using a 32-bit or 64-bit version of Windows.

1) Uninstall any existing instances of the 32-bit Java Runtime Environment (JRE) or Java Development Kit (JDK). Leaving them installed can cause runtime failures, as the library may fail to load.
2) Download and install the latest version of the 64-bit Java Development Kit (JDK).
3) Follow the steps in section **2.1 How to Download and Set Up the Java Sample Software** to download and install the latest Dynamag/DynaMAX/eDynamo/mDynamo SCRA Windows SDK. You may download and install it directly on the target workstation where it will be used, or you may opt to install it on a master development workstation and copy the dependencies to the target workstation manually.
4) If you opted to manually copy the Dynamag/DynaMAX/eDynamo/mDynamo SCRA Windows SDK dependencies from a master development workstation to the target workstation where it will be used, follow these steps:
   a) On the master workstation, navigate to the root of the MTSCRA Windows SDK. By default, it will be **C:\Program Files (x86)\MagTek\SCRA Windows SDK**.
   b) Open the **\x64** subfolder and copy all the files to the target workstation's **C:\Windows\System32** folder.
5) Connect the device to the workstation. Windows will install the device drivers automatically. Wait for Windows to report the driver installation is complete.
6) Launch a Windows command prompt as an Administrator.

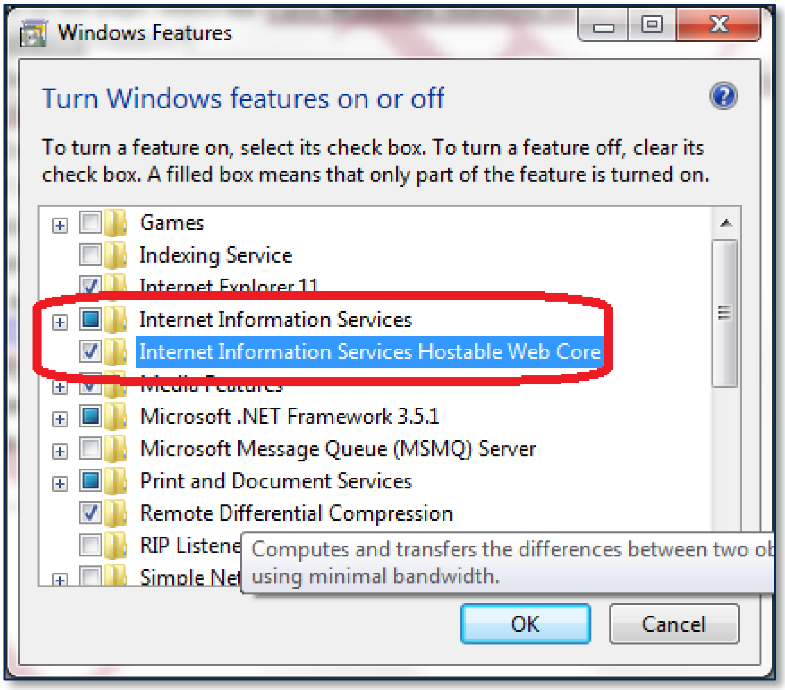
7) **cd** to the root of the folders where the Dynamag/DynaMAX/eDynamo/mDynamo SCRA Windows SDK is installed. By default, it will be **C:\Program Files (x86)\MagTek\SCRA Windows SDK\Sample Code\Java\Object** for 64-bit Windows.
8) Type **build.bat** and press **Enter** to build the Java Demo software.
9) Type **test.bat** and press **Enter** to launch the Java Demo software.
10) Use the Java Demo software.

2.5 How to Set Up the Applet With the 32-bit JRE/JVM

MagTek highly recommends using the 32-bit version of Java when using the MTSCRA Java applet, regardless of whether you are using a 32-bit or 64-bit version of Windows.

To set up the Java applet using the 32-bit version of Java on either a 32-bit or 64-bit version of Windows, follow these steps:

1) Follow the steps in section **2.3 How to Set Up the Java Library With the 32-bit JRE/JVM**. Having a working JVM, working Java library, working drivers, and working DLLs are prerequisites for using the applet.
2) Verify Java is installed, and that the Internet Explorer Java plugin is working correctly by using Oracle's Java applet test page, usually provided as a link or auto-launch at the end of installation.
3) On the Windows 7 workstation you will use for development, enable Internet Information Services 7 (IIS) as follows:
   a) Log in to a Windows 7 workstation using an administrator account.
   b) Launch the Windows **Control Panel**.
   c) Select the **Programs and Features** item to open the **Programs and Features** page.
   d) On the left side of the page, select the **Turn Windows features on or off** link to launch the **Windows Features** window.
   e) Turn on the checkboxes for **Internet Information Services** and **Internet Information Services Hostable Web Core**.



Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ☐ Games
- ☐ Indexing Service
- ☑ Internet Explorer 11
- ■ Internet Information Services
- ☑ Internet Information Services Hostable Web Core
- ☑ Media Features
- ■ Microsoft .NET Framework 3.5.1
- ☐ Microsoft Message Queue (MSMQ) Server
- ■ Print and Document Services
- ☑ Remote Differential Compression
- ☐ RIP Listene
- ☐ Simple Net

Computes and transfers the differences between two ol using minimal bandwidth.

f) Press the **OK** button to launch a progress window. Wait for Windows to install IIS.

4) Launch a web browser and navigate to **//localhost**. Verify the IIS default page appears as shown in **Figure 2-3**.

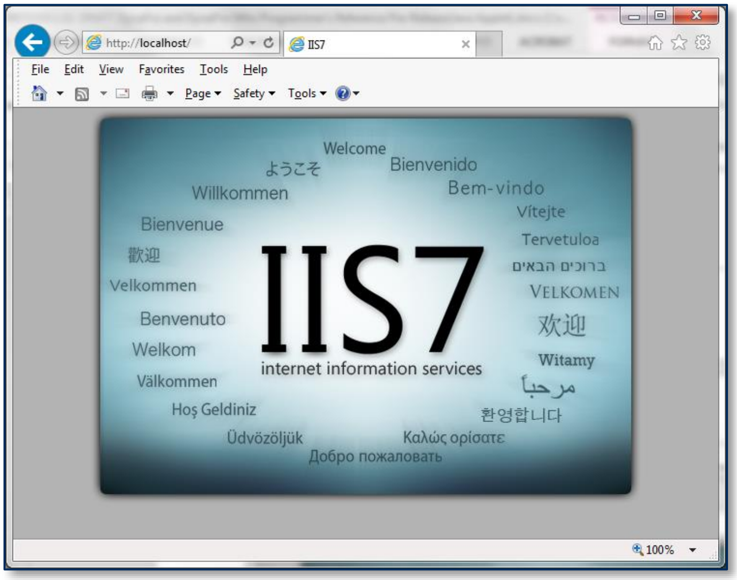


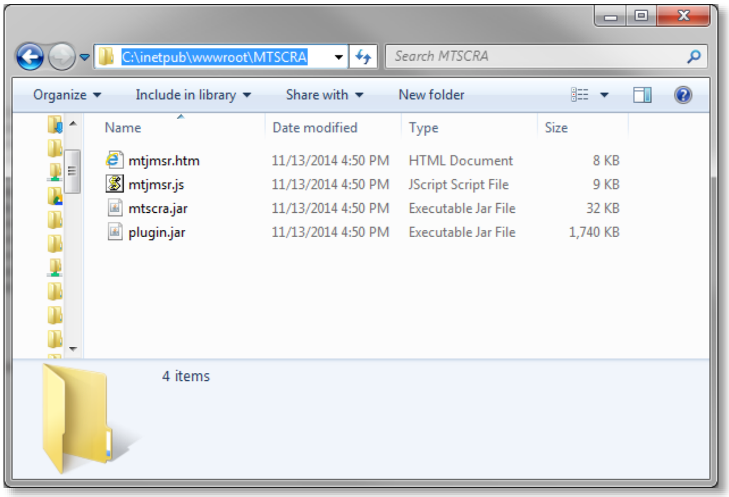
Figure 2-3 - IIS Default Page

5) If it does not already exist, create a **MTSCRA** folder in **C:\inetpub\wwwroot\**. If it does exist, delete its contents.

6) On the workstation where the Dynamag/DynaMAX/eDynamo/mDynamo SCRA Windows SDK is installed, navigate to the folder where it is installed. By default, it will be **C:\Program Files\MagTek\SCRA Windows SDK\Sample Code\Java Applet\Object** for 32-bit operating

systems, or **C:\Program Files (x86)\MagTek\SCRA Windows SDK\Sample Code\Java Applet\Object** for 64-bit operating systems.

7) Copy the contents to **C:\inetpub\wwwroot\MTSCRA**.



Name	Date modified	Type	Size
mtjmsr.htm	11/13/2014 4:50 PM	HTML Document	8 KB
mtjmsr.js	11/13/2014 4:50 PM	JScript Script File	9 KB
mtscra.jar	11/13/2014 4:50 PM	Executable Jar File	32 KB
plugin.jar	11/13/2014 4:50 PM	Executable Jar File	1,740 KB

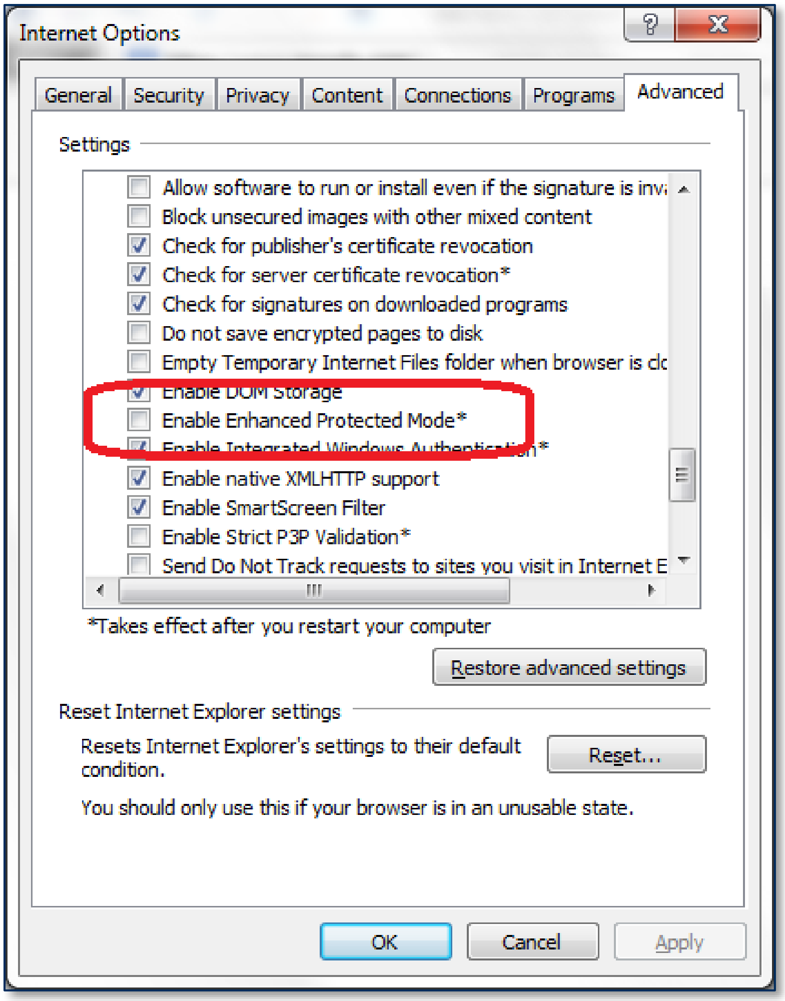
Figure 2-4 - inetpub Structure

8) Connect the device to the workstation. Windows will install the device drivers automatically. Wait for Windows to report the driver installation is complete.

9) Open Internet Explorer as an administrator.

10) If you are using a 64-bit version of Windows with IE8 or IE9, make sure to launch directly in 32-bit mode using the iexplore.exe found in **C:\Program Files (x86)**. Verify you are running in 32-bit mode using the **Help** > **About** menu.

11) If you are running a 64-bit version of Windows with IE10 or higher, choose the **Internet options** that enable 32-bit mode / disable **Protected Mode** for the zone you are accessing. Also turn **OFF** the checkbox for **Enhanced Protected Mode** in the **Internet Options** > **Advanced** tab.



Internet Options

Settings

- ☐ Allow software to run or install even if the signature is inv
- ☐ Block unsecured images with other mixed content
- ☑ Check for publisher's certificate revocation
- ☑ Check for server certificate revocation*
- ☑ Check for signatures on downloaded programs
- ☐ Do not save encrypted pages to disk
- ☐ Empty Temporary Internet Files folder when browser is cl
- ☑ Enable DOM Storage
- ☐ Enable Enhanced Protected Mode*
- ☑ Enable Integrated Windows Authentication*
- ☑ Enable native XMLHTTP support
- ☑ Enable SmartScreen Filter
- ☐ Enable Strict P3P Validation*
- ☐ Send Do Not Track requests to sites you visit in Internet E

*Takes effect after you restart your computer

Reset Internet Explorer settings

Resets Internet Explorer's settings to their default condition.

You should only use this if your browser is in an unusable state.

12) If you changed the value of the **Enable Enhanced Protected Mode** checkbox, restart Windows.

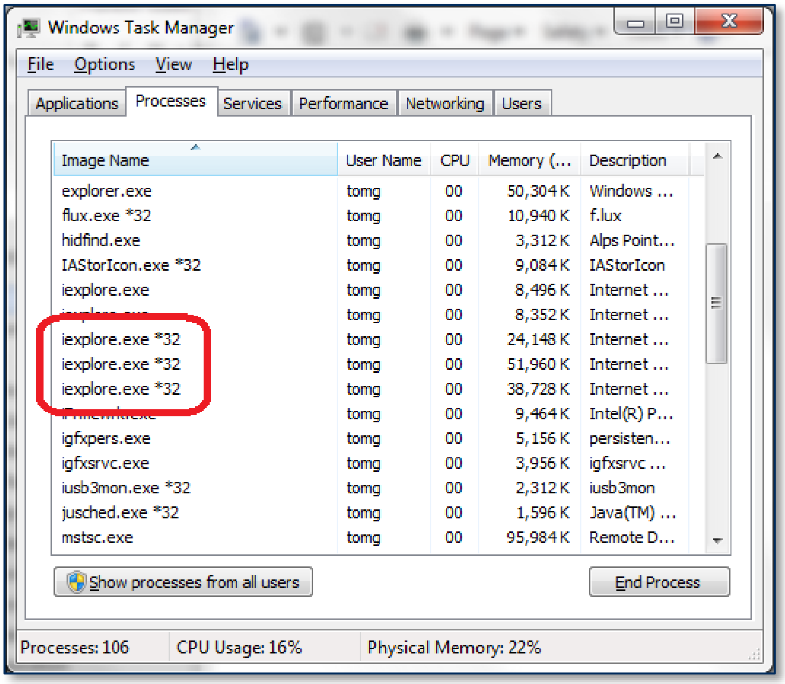
13) Open Windows Task Manager (**Ctrl-Alt-Del** > **Start Task Manager**).

14) Open the **Processes** tab and sort by **Image Name**.

15) Note the number and location of all **iexplore.exe *32** processes.

16) In Internet Explorer, navigate to http://localhost/MTSCRA/MTSCRASample.html.

17) In the Windows Task Manager **Processes** tab, find the new process for the Internet Explorer tab you just opened and make sure it is running in 32-bit mode (**iexplore.exe *32** instead of **iexplore.exe**).



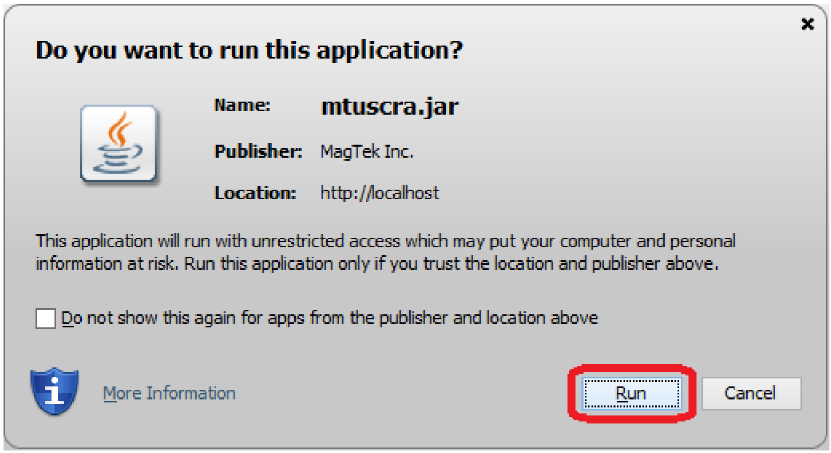
Windows Task Manager
File Options View Help
Applications Processes Services Performance Networking Users
Show processes from all users
End Process

Image Name	User Name	CPU	Memory (...	Description
explorer.exe	tomg	00	50,304 K	Windows ...
flux.exe *32	tomg	00	10,940 K	f.lux
hidfind.exe	tomg	00	3,312 K	Alps Point...
IAStorIcon.exe *32	tomg	00	9,084 K	IAStorIcon
iexplore.exe	tomg	00	8,496 K	Internet ...
iexplore.exe	tomg	00	8,352 K	Internet ...
iexplore.exe *32	tomg	00	24,148 K	Internet ...
iexplore.exe *32	tomg	00	51,960 K	Internet ...
iexplore.exe *32	tomg	00	38,728 K	Internet ...
IFrmewrk.exe	tomg	00	9,464 K	Intel(R) P...
igfxpers.exe	tomg	00	5,156 K	persisten...
igfxsrvc.exe	tomg	00	3,956 K	igfxsrvc ...
iusb3mon.exe *32	tomg	00	2,312 K	iusb3mon
jusched.exe *32	tomg	00	1,596 K	Java(TM) ...
mstsc.exe	tomg	00	95,984 K	Remote D...

Processes: 106 CPU Usage: 16% Physical Memory: 22%

18) Close the **Windows Task Manager** window.

19) Internet Explorer will display a welcome page and will pop up a **Do you want to run this application?** window. Press the **Run** button to run the Java applet.



Do you want to run this application?

Name: mtuscra.jar

Publisher: MagTek Inc.

Location: http://localhost

This application will run with unrestricted access which may put your computer and personal information at risk. Run this application only if you trust the location and publisher above.

□ Do not show this again for apps from the publisher and location above

20) On the Read & Parse Card Data page, select the device to open, then press the **Open Device** button. Command/Response/Status text box in the browser will display the text **Reader Connected**.

21) Use the buttons and fields on the welcome page to test the connection to the device.

2.6 How to Modify Manifest

The **Caller-Allowable-Codebase** attribute is used to identify the domains from which JavaScript code can make calls to your RIA without security prompts. Set this attribute to the domain that hosts the

JavaScript code. If a call is made from JavaScript code that is not located in a domain specified by the **Caller-Allowable-Codebase** attribute, the call is blocked. To specify more than one domain, separate the domains by a space, for example:

```
Caller-Allowable-Codebase: *.yahoo.com *.google.com *.magtek.com *
```

The **Application-Library-Allowable-Codebase** attribute identifies the locations where your signed RIA is expected to be found. This attribute is used to determine what is listed in the Location field for the security prompt that is shown to users when the JAR file for your RIA is in a different location than the JNLP file or HTML page that starts your RIA. If the files are not in the locations identified, the RIA is blocked. Set this attribute to the domains where the JAR file, JNLP file, and HTML page are located. To specify more than one domain, separate the domains by a space, for example:

```
Application-Library-Allowable-Codebase: *.yahoo.com *.google.com
*.magtek.com *
```

For more information regarding the JAR File Manifest Attributes for Security, please visit this website
http://docs.oracle.com/javase/7/docs/technotes/guides/jweb/security/manifest.html

In order to modify the Manifest file, please follow these steps.

1) Find installation folder by default, the installation folder is: **\Sample Code\Java Applet\Object\Unsigned**
2) Launch the command prompt and extract the META-INF/MANIFEST.MF from the jar file.
   ```
   jar xf mtscra.jar  META-INF/MANIFEST.MF
   ```
3) Open **MANIFEST.MF** and look for the **Caller-Allowable-Codebase** and **Application-Library-Allowable-Codebase** and add your website URL to the list like the example above.
4) Update the manifest to the jar file.
   ```
   jar umf META-INF/MANIFEST.MF mtscra.jar
   ```

2.7 How to Sign JAR

These instructions provide an overview of obtaining and using Sun Java signing and a digital certificate.

1) Make sure your machine has the latest Java JDK installed.
2) Generate a public/private key pair by entering the following command, specifying an alias for your keystore:
   ```
   keytool -genkey -keyalg rsa -alias MyCert
   ```
3) Generate a certificate signing request (CSR) by entering the following command:
   ```
   keytool -certreq -alias MyCert
   ```
   After prompting you to enter the password for your keystore, keytool will generate a CSR.
4) Save the certificate received from the Certificate provider as Certname.p7b.
5) Import your Digital Certificate by entering the following command:
   ```
   keytool -import -alias MyCert -file Certname.p7b
   ```

In this string, keytool is requested to import the Digital ID “Certname.cer” into the keystore MyCert.

6) Bundle your applet into a Java Application Resource (JAR) file by entering the following command:

```
jar cvf C:\mtscra.jar
```

7) Sign your applet by using jarsigner to sign the JAR file, using the private key you saved in your keystore:

```
jarsigner C:\mtscra.jar MyCert
```

8) Verify the output of your signed JAR file by entering the following command:

```
jarsigner -verify -verbose -certs C:\mtscra.jar
```

Please visit this website https://docs.oracle.com/javase/tutorial/deployment/jar/signing.html for more information regarding signing JAR files.

3 MTSCRA Functions

If you are developing Java software, follow the setup steps in section **2.1 How to Download and Set Up the Java Sample Software**, then create an instance of the `MTSCRA` object in your software project, then use Java method calls to invoke the functions described in this chapter to communicate with the device. For sample code that demonstrates how to use these functions, see `JavaSample.java` in the SDK files.

If you are developing HTML / JavaScript software using the Java applet, follow the setup steps in section **2.5 How to Set Up the Applet With the 32-bit JRE/JVM**, create an instance of the applet in your HTML, then use JavaScript to invoke the functions described in this chapter to communicate with the device. For sample code that demonstrates how to use these functions, see the `MTSCRASample.html` sample code in the SDK files.

Generally, these functions will run in one of two modes:

- **Asynchronous** functions will return data using the event handlers (callback functions) defined in section **4 MTSCRAEvent**.
- **Synchronous** functions will return requested data immediately in the function's return value. If the requested data is not available immediately, synchronous calls will generally block until a specified wait time has elapsed.

Most calls that wait for input from the user will run in the asynchronous mode.

3.1 getSDKVersion

This function retrieves the Java library version information.

```
String getSDKVersion();
```

Return Value: String containing the Version of the Java library.

3.2 openDevice

This function opens a connection to the device. The event associated with this command is **onDeviceConnectionStateChanged**.

```
long openDevice(String deviceURI);
```

Parameter	Description
deviceURI	URI of the device. For USB devices, deviceURI should be an empty string. For Bluetooth LE devices, deviceURI should be BLE://XXXXXXXX, where XXXXXXXX is variable length string indicate device friendly name.

Return Value:
0 = Success
Non-Zero = Error

3.3 closeDevice

This function closes the connection to the device. The event associated with this command is **onDeviceConnectionStateChanged**.

```
long closeDevice();
```

Return Value:
0 = Success
Non-Zero = Error

3.4 init

This function registers a listener for callback and applies for Java only not Java Applet. See section **4 MTSCRAEvent** for details about implementing the listener object.

```
Void init(MTSCRAEvent e);
```

Return Value: None

3.5 getDeviceList

This function enumerates all SCRA devices connected to the host.

```
String getDeviceList();
```

Return Value:
Returns a string which contains zero or more device paths separated by ','.

3.6 isDeviceConnected

This function retrieves the connection status of the device.

```
boolean isDeviceConnected();
```

Return Value:
`True` if the host is connected to the device, otherwise `False`.

3.7 isDeviceEMV

This function indicates whether the device supports EMV or not.

```
boolean isDeviceEMV();
```

Return Value:
`True` = EMV supported by the device.

3.8 getFirmware

This function retrieves firmware revision number.

```
String getFirmware();
```

Return Value: String containing the Firmware revision number.

3.9 clearBuffer

This function clears the library's local cache of card swipe data.

```
void clearBuffer();
```

3.10 getCardData

This function retrieves card data after a cardholder has swiped a card.

```
String getCardData();
```

Output	Description
String	String containing the card data. Fields are delimited by the pipe character "|". Fields: Device ID (USB Vendor ID) |Device Serial Number |Card Swipe Status |CardEncode Type |Track 1 Decode Status |Track 2 Decode Status |Track 3 Decode Status |MagnePrint Status |Track 1 Length |Track 2 Length |Track 3 Length |Masked Track 1 Length |Masked Track 2 Length |Masked Track 3 Length |MagnePrint Length |Card Data |Masked Card Data |DUKPT Session ID |DUKPT Key Serial Number |First Name |Last Name |PAN |Month |Year |Track 1 Data |Track 2 Data |Track 3 Data |Masked Track 1 Data |Masked Track 2 Data |Masked Track 3 Data |MagnePrint Data |Battery Level

Return Value: String containing the card data.

Example:

```
11|B354C81061217AA|0|0|0|0|0|61403000|60|39|0|60|39|0|54|2542393939393
93935353535353535353535305E4D43445320352F535743502054455354 5E31393034323
0313030303031303032333330303030303F3B39393939393935353535353535353535353
03D313930343230313030303030303032333330303F|%B9999995555555550^LAST/
FIRST
M^1904201000010023300000?;9999995555555550=19042010000000233000?|0000
000000000000|0000000000000000000000|FIRST|LAST|9999995555555550|04|19|25
4239393939393935353535353535353535305E4D43445320352F5357435020544553545E
31393034323031303030303130303233333030303030303F|3B39393939393935353353
53535353535303D3139303432303130303030303030323333303030303F||%B999999555
5555550^LAST/FIRST
M^1904201000010023300000?|;9999995555555550=19042010000000233000?||02
002A8069912E8BAB88EA89082AA4D701473EC3C1A0C8E1DF0075F40A167713CF6AE4DF
442DC4ED8831377E98AFA88AE422536CEF2F|100
```

3.11 getCardName

This function retrieves the card name after a cardholder has swiped a card.

```
String getCardName();
```

Return Value: String containing the card name.

3.12 getFirstName

This function retrieves the card first name after a cardholder has swiped a card.

```
String getFirstName();
```

Return Value: String containing the card first name.

3.13 getLastName

This function retrieves the card last name after a cardholder has swiped a card.

```
String getLastName();
```

Return Value: String containing the card last name.

3.14 getMiddleName

This function retrieves the card middle name after a cardholder has swiped a card.

```
String getMiddleName();
```

Return Value: String containing the card middle name.

3.15 getEncodeType

This function returns a string value for a card type after a cardholder has swiped a card.

```
String getEncodeType();
```

Return Value:
0 = Other
1 = Financial
2 = AAMVA
3 = Manual
4 = Unknown
5 = ICC
6 = Contactless ICC

3.16 getPAN

This function retrieves the card PAN value after a cardholder has swiped a card.

```
String getPAN();
```

Return Value: PAN

3.17 getPANLength

This function retrieves the length of the card PAN value after a cardholder has swiped a card.

```
long getPANLength();
```

Return Value: long containing the PAN length after a cardholder has swiped a card.

3.18 getProductID

This function returns the device's product identifier after a cardholder has swiped a card.

```
String getProductID();
```

Return Value:
Returns a null terminated string. For example - "2"

3.19 getDeviceName

This function retrieves the device name after a cardholder has swiped a card.

```
String getDeviceName();
```

Return Value: String containing the device name.

3.20 getCapMagneSafe20Encryption.

This function retrieves MagneSafe 2.0 encryption information after a cardholder has swiped a card.

```
String getCapMagneSafe20Encryption();
```

Return Values:
"1" = Device uses MagneSafe 2.0 Encryption
"0" = Device does not use MagneSafe 2.0 Encryption

3.21 getCapMagStripeEncryption

This function retrieves device capability of track encryption after a cardholder has swiped a card.

```
String getCapMagStripeEncryption();
```

Return Value:
“1” = Device can encrypt the track
“0” = Device cannot encrypt the track

3.22 getExpDate

This function retrieves the card expiration date after a cardholder swipes a card. This function is deprecated. Use **getCardExpDate** instead.

```
String getExpDate();
```

Return Value: String containing the card expiration date

3.23 getCardExpDate

This function retrieves the card expiration date after a cardholder swipes a card.

```
String getCardExpDate();
```

Return Value: String containing the card expiration date.

3.24 getExpDateMonth

This function retrieves the card expiration month after a cardholder swipes a card.

```
String getExpDateMonth();
```

Return Value: String containing the card expiration month.

3.25 getExpDateYear

This function retrieves the card expiration year after a cardholder swipes a card.

```
String getExpDateYear();
```

Return Value: String containing the card expiration year.

3.26 getCardIIN

This function retrieves card issuer identification number (IIN) after a cardholder swipes a card.

```
String getCardIIN();
```

Return Value: String containing the card issuer identification number.

3.27 getCardLast4

This function retrieves the last 4 digits of the card number (PAN) after a cardholder swipes a card.

```
String getCardLast4();
```

Return Value: String containing the last 4 digits.

3.28 getCapTracks

This function retrieves track capability. See commands `0x00` and `0x01`, and property `0x05` in ***D99875475 MagneSafe V5 Communication Reference Manual*** for detail.

```
String getCapTracks();
```

Return Value: Hex String containing track capability.

3.29 getTrackDecodeStatus

This function retrieves track decode status.

```
String getTrackDecodeStatus();
```

Return Value:
Track Decode Status. Consists of three 2-byte hex values representing the decode status for tracks 1, 2, and 3 (respectively from left to right). Values are:
00 = Track OK
01 = Track read Error
02 = Track is Blank

3.30 getTrack1DecodeStatus

This function retrieves track 1 decode status.

```
String getTrack1DecodeStatus();
```

Return Value: String containing track 1 decode status.
00 = Track OK
01 = Track read Error
02 = Track is Blank

3.31 getTrack2DecodeStatus

This function retrieves the track 2 decode status.

```
String getTrack2DecodeStatus();
```

Return Value: String containing track 2 decode status.
00 = Track OK
01 = Track read Error
02 = Track is Blank

3.32 getTrack3DecodeStatus

This function retrieves the track 3 decode status.

```
String getTrack3DecodeStatus();
```

Return Value: String containing track 3 decode status.
00 = Track OK
01 = Track read Error
02 = Track is Blank

3.33 getMaskedTracks

This function retrieves masked track data.

```
String getMaskedTrack();
```

Return Value: String containing masked track 1, track 2, and track3 data.

3.34 getTrack1Masked

This function retrieves masked track 1 data.

```
String getTrack1Masked();
```

Return Value: String containing masked track 1 data.

3.35 getTrack2Masked

This function retrieves masked track 2 data.

```
String getTrack2Masked();
```

Return Value: String containing masked track 2 data.

3.36 getTrack3Masked

This function retrieves masked track 3 data.

```
String getTrack3Masked();
```

Return Value: String containing masked track 3 data.

3.37 getTrack1

This function retrieves track 1 data.

```
String getTrack1();
```

Return Value: String containing track 1 data.

3.38 getTrack2

This function retrieves track 2 data.

```
String getTrack2();
```

Return Value: String containing track 2 data.

3.39 getTrack3

This function retrieves track 3 data.

```
String getTrack3();
```

Return Value: String containing track 3 data.

3.40 getMagnePrint

This function retrieves MagnePrint data

```
String getMagnePrint();
```

Return Value: String containing MagnePrint data.

3.41 getMagnePrintLength

This function retrieves MagnePrint data length.

```
long getMagnePrintLength();
```

Return Value: long value indicating the length of MagnePrint data.

3.42 getMagnePrintStatus

This function retrieves MagnePrint status. See ***D99875475 MagneSafe V5 Communication Reference Manual*** for detail.

```
String getMagnePrintStatus();
```

Return Value: String containing the MagnePrint status.

3.43 getEncryptionStatus

This function retrieves encryption status. See ***D99875475 MagneSafe V5 Communication Reference Manual*** for detail.

```
String getEncryptionStatus();
```

Return Value: String containing encryption status.

3.44 getDeviceSerial

This function returns the device's serial number.

```
String getDeviceSerial();
```

Return Value: Returns a string. For example - "12345678"

3.45 getSessionID

This function retrieves the device session ID, which the host can use to uniquely identify a transaction to prevent replay. See ***D99875475 MagneSafe V5 Communication Reference Manual*** for detail.

```
String getSessionID();
```

Return Value: Returns a string containing the session id.

3.46 getKSN

This function retrieves the device's key serial number (KSN) after a card swipe.

```
String getKSN();
```

Return Value: Returns a string containing the key serial number.

3.47 sendCommand

This function sends a direct “SET” byte command to the device. For information about direct commands, see ***D99875475 MagneSafe V5 Communication Reference Manual***.

```
long sendCommand(String lpCommand);
```

Parameter	Description
lpCommand	A hexadecimal command string to send to the device. For example, command “0003” (where “00” is the command number and “03” is the property ID) will retrieve device serial number.

Return Value: Null terminated hex string for the return result. NULL value for failed.

3.48 sendCommandWithLength

This function sends a direct “SET” byte command to the device. For information about direct commands, see ***D99875475 MagneSafe V5 Communication Reference Manual***.

```
long sendCommandWithLength(String lpCommand);
```

Parameter	Description
lpCommand	A hexadecimal command string to send to the device. For example, command “000103” (where “00” is command number, "01" is the length, and “03” is property ID) will retrieve the device serial number.

Return Value: Null terminated hex string for the return result. NULL value for failed.

3.49 sendData

This function sends a direct “SET” byte command to the device. For information about direct commands, see ***D99875475 MagneSafe V5 Communication Reference Manual***. The event associated with this command is **onDeviceResponse**.

```
long sendData(String lpCommand);
```

Parameter	Description
lpCommand	A hexadecimal command string to send to the device. For example, command “000103” (where “00” is command number, "01" is the length, and “03” is property ID) will retrieve the device serial number.

Return Value: 0 for no error.

3.50 getBatteryLevel

This function retrieves battery level between 0% and 100%.

```
long getBatteryLevel();
```

Return Value: percentage of battery level. 100 means 100%.

3.51 getSwipeCount

This function is reserved for future use.

```
long getSwipeCount();
```

Return Value: long containing swipe count.

3.52 getResultCode

This function will return the result code of V5 command or V5 extend command.

```
long getResultCode();
```

Return Value: 0 for no error.

The following are Result Codes when a command is an EMV command from an EMV device.

EMV Command Result Code Description
• 0x0000 = Success, the transaction process has been started • 0x0381 = Failure, DUKPT scheme is not loaded • 0x0382 = Failure, DUKPT scheme is loaded but all of its keys have been used • 0x0383 = Failure, DUKPT scheme is not loaded (Security Level not 3 or 4) • 0x0384 = Invalid Total Transaction Time field • 0x0385 = Invalid Card Type field • 0x0386 = Invalid Options field • 0x0387 = Invalid Amount Authorized field • 0x0388 = Invalid Transaction Type field • 0x0389 = Invalid Cash Back field • 0x038A = Invalid Transaction Currency Code field • 0x038B = Invalid Selection Status • 0x038C = Invalid Selection Result • 0x038D = Failure, no transaction currently in progress • 0x038E = Invalid Reporting Option • 0x038F = Failure, transaction in progress, card already inserted • 0x0390 = Device Has No Keys • 0x0391 = Invalid Device Serial Number • 0x0396 = Invalid System Date and Time

3.53 getCardServiceCode

This function retrieves the card service code and should be called after a cardholder swipes a card and before calling clearBuffer.

```
String getCardServiceCode();
```

Return Value:
String representing the card service code after a cardholder swipes a card.

3.54 getTLVPayload

This function will return the TLV payload after a card swipe.

```
String getTLVPayload();
```

Return Value:
A hex string representing the payload as follows.

```
FA <len> (Container for Generic Data)
        DFDF25 <len> <val> (Device Serial Number)
        F4 <len> (Container for MSR Data)
                DFDF37 <len> <val> (Encrypted Track 1)
                DFDF39 <len> <val> (Encrypted Track 2)
                DFDF3B <len> <val> (Encrypted Track 3)
                DFDF3C <len> <val> (Encrypted MagnePrint)
                DFDF3D <len> <val> (Encrypted MagnePrint Status)
                DFDF50 <len> <val> (KSN)
```

Tag	Description
FA	Container for generic data
DFDF25	IFD Serial Number
F4	Container for MSR data
DFDF37	Encrypted T1
DFDF39	Encrypted T2
DFDF3B	Encrypted T3
DFDF3C	Encrypted MagnePrint
DFDF3D	Encrypted MagnePrint Status
DFDF50	MSR KSN

3.55 startTransaction (EMV Only)

This function starts an EMV L2 transaction for smart card.

```
byte[] startTransaction(
byte timeLimit,
byte cardType,
byte option,
byte[] amount,
byte transactionType,
byte[] cashBack,
byte[] currencyCode,
byte mode);
```

Parameter	Description
timeLimit	Specifies the maximum time, in seconds, allowed to complete the total transaction. This includes time for the user to insert the card, choose a language, choose an application, and online processing. If this time is exceeded, the transaction will be aborted and an appropriate Transaction Status will be available. Value 0 is not allowed.
cardType	Card Type to Read: 0x01 = Magnetic Stripe (as alternative to EMV L2, card swipe causes abort of EMV L2) 0x02 = Contact chip card 0x03 = Magnetic Stripe and Contact chip Card. 0x04 = Contactless chip card 0x05 = Magnetic Stripe and Contactless chip card. 0x06 = Contact chip card and Contactless chip card. 0x07 = Magnetic Stripe, Contact chip card, Contactless chip card. Refer to **4.11Appendix G** for supported devices.
option	0x00 = Normal 0x01 = Bypass PIN 0x02 = Force Online 0x04 = Acquirer not available (Note: prevents long timeout on waiting for host approval) (causes “decline” to be generated internally if ARQC is generated) To use Quick Chip mode, set the most significant bit to ‘1’. 0x80 = Quick Chip, Normal 0x81 = Quick Chip, Bypass PIN 0x82 = Quick Chip, Force Online Refer to **4.11Appendix G** for supported devices.
amount	Amount Authorized (EMV Tag 9F02, format n12, 6 bytes) in hex string For example: “000000000999”, means 9.99 dollars.
transactionType	Valid values: 0x00 = Purchase (listed as “Payment” on ICS) 0x01 = Cash Advance (not supported for this reader) 0x02 or 0x09 = Cash back (0x09 only supported when using contactless) 0x04 = Goods (Purchase) 0x08 = Services (Purchase) 0x10 = International Goods (Purchase) 0x20 = Refund 0x40 = International Cash Advance or Cash Back 0x80 = Domestic Cash Advance or Cash Back
cashBack	Cash back Amount (if non-zero, EMV Tag 9F03, format n12, 6 bytes) in hex string. For example: “000000001000”, means 10.00 dollars.

currencyCode	Transaction Currency Code (EMV Tag 5F2A, format n4, 2 bytes) Sample Valid values: 0x0840 – US Dollar 0x0978 – Euro 0x0826 – UK Pound
mode	This single byte field indicates the level of Transaction Status notifications the host desires to receive during the course of this transaction. 0x00 = Termination Status only (normal termination, card error, timeout, host cancel) 0x01 = Major Status changes (terminations, card insertions, waiting on user) 0x02 = All Status changes (documents the entire transaction flow)

Return Value: This function will always return an empty string. To get the result cold of this command, use getResultCode() function.

3.56 setUserSelectionResult (EMV Only)

This function sets the user selection result. It should be called after receiving the OnUserSelectRequest event which is triggered after the user makes a selection.

```
byte[] setUserSelectionResult(byte status, byte selection);
```

Parameter	Description
status	Indicates the status of User Selection: 0x00 – User Selection Request completed, see Selection Result 0x01 – User Selection Request aborted, cancelled by user 0x02 – User Selection Request aborted, timeout
selection	Indicates the menu item selected by the user. This is a single byte zero based binary value.

Return Value: This function will always returns an empty string. To get the result code of this command, use getResultCode() function.

3.57 setAcquirerResponse (EMV Only)

This function informs EMV device to process transaction decision from acquirer.

```
byte[] setAcquirerResponse(byte[] response);
```

Parameter	Description
response	See **0**. Hex string for the response data. First two bytes indicate message length, following TLV response message.

Return Value: This function will always returns an empty string. To get the result code of this command, use getResultCode() function.

3.58 cancelTransaction (EMV Only)

This function informs EMV device to cancel current transaction.

```
byte[] cancelTransaction();
```

Return Value: This function will always returns an empty string. To get the result code of this command, use getResultCode() function.

3.59 sendExtendedCommand (EMV Only)

This function sends a direct extended command to the device using a hex value.

```
long sendExtendedCommmand(String data);
```

Return Value: 0 for no error.

4 MTSCRAEvent

If you are using the Java library, after calling the functions in section **2.1 How to Download and Set Up the Java Sample Software**, the MTSCRA Windows SDK libraries for Java will invoke the callback functions in this section to provide the requested data and/or a detailed response. Custom software that uses the MTSCRA libraries for Java should create an object that implements the following functions to process the returning data, then register it as a listener using the **init** function. For sample code that demonstrates how to use these functions, see `JavaSample.java` in the SDK files.

If you are using the Java applet, after calling the functions in section **2.5 How to Set Up the Applet With the 32-bit JRE/JVM**, the MTSCRA Java applet will invoke the callback functions in this chapter to provide the requested data and/or a detailed response. Custom code that uses the MTSCRA Java applet should implement the following JavaScript functions to process the returning data. For sample code that demonstrates how to use these functions, see the `MTSCRASample.html` sample code in the SDK files.

4.1 onLibLoaded

```
public void onLibLoad(int status);
```

Parameter	Description
status	An integer 1 indicating the DLL is loaded and ready to call.

4.2 onDeviceConnectionStateChanged

```
public void onDeviceConnectionStateChanged(int lpDevState);
```

Parameter	Description
lpDevState	An integer value: 0 = Device is disconnected 1 = Device is connected.

4.3 onError

```
public void onError(int errorCode);
```

Parameter	Description
errorCode	An integer error code for an error handler.

4.4 onDataReceived

This event is called when the device has card data to transmit to the host.

```
public void onDataReceived(String data);
```

Parameter	Description
lpData	A string containing card data.

4.5 onDeviceResponse

Return event for **sendData**.

```
public void onDeviceResponse(String data);
```

Parameter	Description
lpData	A response string for sendData function

4.6 onTransactionStatus (EMV Only)

This event is called when the device has transaction status update to send to the host.

```
void onTransactionStatus(String data);
```

This notification will send hex string to represent transaction status.

Offset	Field Name	Value
0	Event	Indicates the event that provoked this notification • 0x00 – No events since start of transaction • 0x01 – Card inserted (Contact only) • 0x02 – Card error • 0x03 – Transaction Progress Change • 0x04 – Notification that device is waiting for user selection • 0x05 – Timeout on user selection • 0x06 – Transaction Terminated • 0x07 – Host Cancelled Transaction • 0x08 – Card Removed (Contact only)
1	Current Transaction Time remaining	Indicates the remaining time available, in seconds, for the transaction to complete. If the transaction does not complete within this time it will be aborted.

Offset	Field Name	Value
2	Current Transaction Progress Indicator	This one byte field indicates the current processing stage for the transaction: • 0x00 = No transaction in progress • 0x01 = Waiting for cardholder to present payment • 0x02 = Powering up the card • 0x03 = Selecting the application • 0x04 = Waiting for user language selection (Contact Only) • 0x05 = Waiting for user application selection (Contact Only) • 0x06 = Initiating application (Contact Only) • 0x07 = Reading application data (Contact Only) • 0x08 = Offline data authentication (Contact Only) • 0x09 = Process restrictions (Contact Only) • 0x0A = Cardholder verification (Contact Only) • 0x0B = Terminal risk management (Contact Only) • 0x0C = Terminal action analysis (Contact Only) • 0x0D = Generating first application cryptogram (Contact Only) • 0x0E = Card action analysis (Contact Only) • 0x0F = Online processing • 0x10 = Waiting online processing response • 0x11 = Transaction Complete • 0x12 = Transaction Error • 0x13 = Transaction Approved • 0x14 = Transaction Declined • 0x15 = Transaction Cancelled by MSR Swipe (MSR Only) • 0x16 = EMV error - Conditions Not Satisfied (Contact Only) • 0x17 = EMV error - Card Blocked (Contact Only) • 0x18 = Application selection failed (Contact Only) • 0x19 = EMV error - Card Not Accepted (Contact Only) • 0x1A = Empty Candidate List • 0x1B = Application Blocked
3-4	Final Status	TBD

Return Value:
None

4.7 onDisplayMessageRequest (EMV Only)

This event is called when the device has transaction message update for the host to display to user.

```
void onDisplayMessageRequest(String data);
```

This notification will send a hex string to represent transaction status.
For example:

Hex string: “505245534454e542043415244”
Represent displaying message: “PRESENT CARD”

Return Value:
None

4.8 onUserSelectionRequest (EMV Only)

This event is called when the device has user selection message in transaction for the host to present to the user.

```
void onUserSelectionRequest(String data)
```

This notification will send a hex string to represent user selection request.

Offset	Field Name	Value
0	Selection Type	This field specifies what kind of selection request this is: • 0x00 – Application Selection • 0x01 – Language Selection • Others TBD
1	Timeout	Specifies the maximum time, in seconds, allowed to complete the selection process. If this time is exceeded, the host should send the User Selection Result command with transaction will be aborted and an appropriate Transaction Status will be available. Value 0 is not allowed.
2	Menu Items	This field is variable length and is a collection of “C” style zero terminated strings (maximum 17 strings). The maximum length of each string is 20 characters, not including a Line Feed (0x0A) character that may be in the string. The last string may not have the Line Feed character. The first string is a title and should not be considered for selection. It is expected that the receiver of the notification will display the menu items and return (in the User Selection Result request) the number of the item the user selects. The minimum value of the Selection Result should be 1 (the first item, #0, was a title line only). The maximum value of the Selection Result is based on the number of items displayed.

Return Value:
None

4.9 onARQCReceived (EMV Only)

This event is called when the device has an ARQC message send to the host.

```
void onARQCReceived(String data);
```

This notification will send a hex string for ARQC of this transaction.

Offset	Field Name	Value
0	Message Length	Two byte binary, most significant byte first. This gives the total length of the ARQC message that follows.
2	ARQC Message	See **4.11Appendix D**. It is expected that the host will use this data to process a request.

Return Value:
None

4.10 onTransactionResult (EMV Only)

This event is called when the device has completed the transaction and sends the transaction result to the host.

```
void onTransactionResult(String data);
```

This notification will send a hex string for result of this transaction.

Offset	Field Name	Value
0	Signature Required	This field indicates whether a card holder signature is required to complete the transaction: • 0x00 – No signature required • 0x01 – Signature required If a signature is required, it is expected that the host will acquire the signature from the card holder as part of the transaction data.
1	Batch Data Length	Two byte binary, most significant byte first. This gives the total length of the ARQC message that follows.
3	Batch Data	See **4.11Appendix F**. It is expected that the host will save this data as a record of the transaction.

Return Value:
None

4.11 OnDeviceExtendedResponse (Emv Only)

This event is called when the device has completed processing the command and sends the command result to the host.

```
void onDeviceExtendedResponse(String data)
```

This notification will send a hex string for result of the extended command.

Offset	Field Name	Value
0	Result Code	Result code of the command sent to device in Hex format.
2	Data Length	Two byte binary, most significant byte first. This gives the total length of the Data that follows.
4	Data	Hex data of the command response.

The following are Result Codes when a command is an EMV command from an EMV device.

EMV Command Result Code Description
• 0x0000 = Success, the transaction process has been started • 0x0381 = Failure, DUKPT scheme is not loaded • 0x0382 = Failure, DUKPT scheme is loaded but all of its keys have been used • 0x0383 = Failure, DUKPT scheme is not loaded (Security Level not 3 or 4) • 0x0384 = Invalid Total Transaction Time field • 0x0385 = Invalid Card Type field • 0x0386 = Invalid Options field • 0x0387 = Invalid Amount Authorized field • 0x0388 = Invalid Transaction Type field • 0x0389 = Invalid Cash Back field • 0x038A = Invalid Transaction Currency Code field • 0x038B = Invalid Selection Status • 0x038C = Invalid Selection Result • 0x038D = Failure, no transaction currently in progress • 0x038E = Invalid Reporting Option • 0x038F = Failure, transaction in progress, card already inserted • 0x0390 = Device Has No Keys • 0x0391 = Invalid Device Serial Number • 0x0396 = Invalid System Date and Time

Return Value:
None

Appendix A Status Codes

A.1 Library Status Codes

0x00 = SUCCESS
0x01 = FAILED
0x02 = OPENED
0x03 = MTSCRA_ST_INVALID_PARAM

A.2 Device Status Codes

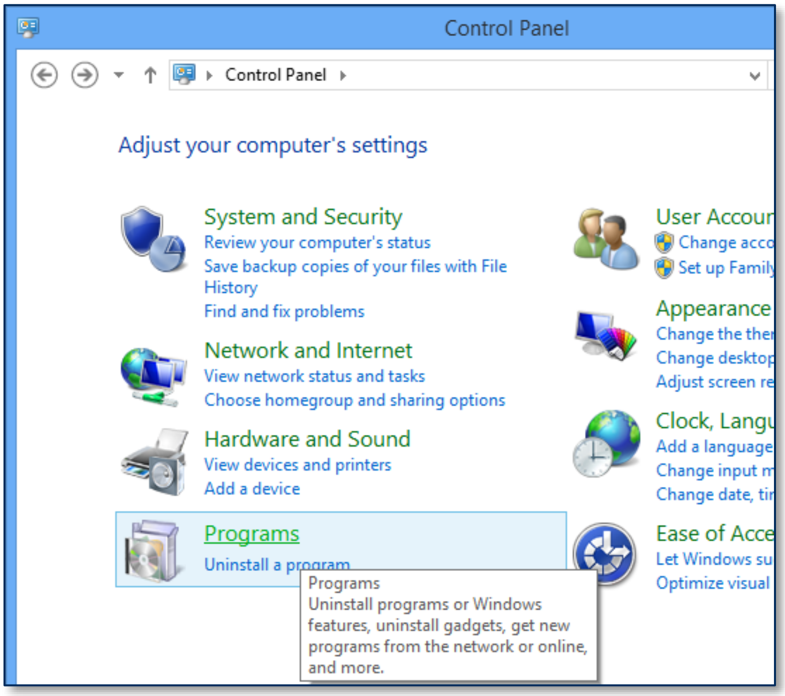
0x00 = State Disconnected
0x01 = State Connected
0x02 = State Error

Appendix B Applet Troubleshooting

B.1 How to Clean Out Previous Applet Versions

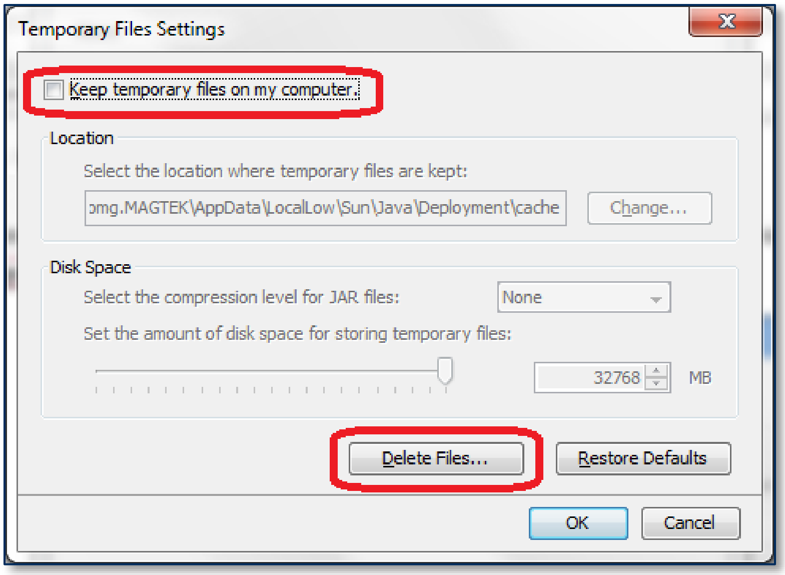
If the Java applet is not launching from the web site correctly (such as “Unable to launch application” error messages or silent failures during applet load), follow these steps to completely remove any previously installed versions:

1) If you are using Windows 8, switch to Desktop mode.
2) Open the Windows **Control Panel**.
3) If the Control Panel is in **View by: Category** mode, select **Programs**.



Programs
Uninstall programs or Windows features, uninstall gadgets, get new programs from the network or online, and more.

4) Click the **Java (32-bit)** link (Windows 8) or the **Java** link (Windows 7) to open the **Java Control Panel** window.
5) Select the **General** tab.
6) Under the **Temporary Internet Files** heading, press the **Settings…** button to launch the **Temporary Files Settings** window.
7) Turn off the checkbox for **Keep temporary files on my computer**.
8) Press the **Delete Files…** button to launch the **Delete Files and Applications** window.



☐ Keep temporary files on my computer.

Location

Select the location where temporary files are kept: mg.MAGTEK\AppData\LocalLow\Sun\Java\Deployment\cache

Disk Space

Select the compression level for JAR files: None

Set the amount of disk space for storing temporary files: 32768 MB

9) In the **Delete Files and Applications** window, turn on **all the checkboxes**, then press the **OK** button to clear all downloaded Java-based software and log files.

10) Press the **OK** button to close the **Temporary Files Settings** window.

11) Press the **OK** button to close the **Java Control Panel** window.

12) Launch Windows Explorer. If you are using a 32-bit version of Windows, navigate to **C:\Windows\System32**. If you are using a 64-bit version of Windows, navigate to **C:\Windows\SysWOW64.**

13) In that folder, search for the following files and delete any that exist:
    a) MTSCRA.dll
    b) MTSCRAJ.dll
    c) MTSCRABLE.dll

14) Re-install the applet by following the steps in section **2.5 How to Set Up the Applet With the 32-bit JRE/JVM.**

B.2 Examining Java Console Outputs for the Applet

Troubleshooting the applet can sometimes involve examining Java console outputs to see where the software load / initialize / run process went wrong. For comparison purposes, this appendix contains a sample Java console output which shows a successful load and initialization of the MTSCRA Java applet.

```
Java Plug-in 10.72.2.14
Using JRE version 1.7.0_72-b14 Java HotSpot(TM) Client VM
User home directory = C:\Users\longv
----------------------------------------------------
c:   clear console window
f:   finalize objects on finalization queue
g:   garbage collect
h:   display this help message
l:   dump classloader list
m:   print memory usage
o:   trigger logging
q:   hide console
r:   reload policy configuration
s:   dump system and deployment properties
t:   dump thread list
v:   dump thread stack
x:   clear classloader cache
0-5: set trace level to <n>
----------------------------------------------------
cache: Initialize resource manager:
com.sun.deploy.cache.ResourceProviderImpl@ede19e
basic: Added progress listener:
sun.plugin.util.ProgressMonitorAdapter@15d4f53
security: Expected Main URL: http://localhost/SCRA/mtscra.jar
basic: Plugin2ClassLoader.addURL parent called for
http://localhost/SCRA/mtscra.jar
network: Connecting http://localhost/SCRA/mtscra.jar with proxy=DIRECT
network: Connecting http://localhost:80/ with proxy=DIRECT
network: Connecting http://localhost/SCRA/mtscra.jar with proxy=DIRECT
network: Connecting http://localhost:80/ with proxy=DIRECT
network: ResponseCode for http://localhost/SCRA/mtscra.jar : 200
network: Encoding for http://localhost/SCRA/mtscra.jar : null
network: Server response: (length: 31983, lastModified: Thu Nov 13
16:50:46 PST 2014, downloadVersion: null, mimeType: application/java-
archive)
network: Downloading resource: http://localhost/SCRA/mtscra.jar
      Content-Length: 31,983
      Content-Encoding: null
network: Wrote URL http://localhost/SCRA/mtscra.jar to File
C:\Users\longv\AppData\Local\Temp\jar_cache8004804528504519879.tmp
security: blacklist: created: NEED_CREATE, lastModified: 0
security: Blacklist file not found or revocation check is disabled
security: Trusted libraries list file not found
security: Blacklist file not found or revocation check is disabled
network: Disconnect connection to http://localhost/SCRA/mtscra.jar
```

```
network: Downloaded http://localhost/SCRA/mtscra.jar:
C:\Users\longv\AppData\Local\Temp\jar_cache8004804528504519879.tmp
cache: Adding MemoryCache entry: http://localhost/SCRA/mtscra.jar
cache: registerReference:
com.sun.deploy.cache.MemoryCache$CachedResourceReference@316e5215: 1
security: http://localhost/SCRA/mtscra.jar is asserting Trusted-Only
security: Loading Deployment certificates from
C:\Users\longv\AppData\LocalLow\Sun\Java\Deployment\security\trusted.c
erts
security: Loaded Deployment certificates from
C:\Users\longv\AppData\LocalLow\Sun\Java\Deployment\security\trusted.c
erts
security: Loading certificates from Deployment session certificate
store
security: Loaded certificates from Deployment session certificate
store
security: Loading certificates from Deployment session certificate
store
security: Loaded certificates from Deployment session certificate
store
security: Loading certificates from Deployment session certificate
store
security: Loaded certificates from Deployment session certificate
store
security: Loading certificates from Deployment session certificate
store
security: Loaded certificates from Deployment session certificate
store
security: Validate the certificate chain using CertPath API
security: Loading Root CA certificates from C:\Program Files
(x86)\Java\jre7\lib\security\cacerts
security: Loaded Root CA certificates from C:\Program Files
(x86)\Java\jre7\lib\security\cacerts
security: Obtain certificate collection in Root CA certificate store
security: Obtain certificate collection in Root CA certificate store
security: Obtain certificate collection in Root CA certificate store
security: Obtain certificate collection in Root CA certificate store
security: The OCSP support is enabled
security: The CRL support is enabled
network: Connecting http://ocsp.verisign.com/ with proxy=DIRECT
network: Connecting http://ocsp.verisign.com:80/ with proxy=DIRECT
security: OCSP Response: GOOD
network: Connecting http://ocsp.verisign.com/ with proxy=DIRECT
security: OCSP Response: GOOD
network: Connecting http://ocsp.verisign.com/ with proxy=DIRECT
security: OCSP Response: GOOD
security: Certificate validation succeeded using OCSP/CRL
security: Saving certificates in Deployment session certificate store
security: Saved certificates in Deployment session certificate store
network: Created version ID: 1.7.0.72
network: Created version ID: 1.7.0.71
```

```
basic: Dialog type is not candidate for embedding
security: User has granted the privileges to the code for this session
only
security: Saving certificates in Deployment session certificate store
security: Saved certificates in Deployment session certificate store
security: Lap null for ai: Appinfo:
type = 2
title = mtscra.jar
vendor = null
from = http://localhost/SCRA/mtscra.jar
security = 0
lapURL = http://localhost/SCRA//MagTekUSCRA
appArgs =
##docbase:http://localhost/SCRA/MTSCRASample.html##Parameters:{mayscri
pt=mayscript, dll_ver=1.0.0,
pluginspage=http://java.com/en/download/index.jsp, java_arguments=,
width=240, cache_option=No,
code=com/magtek/windows/scra/usb/MTSCRA.class, type=application/x-
java-applet;version=1.6, height=200, classloader_cache=true,
scriptable=true, style=visibility:hidden;, __applet_relaunched=false,
name=MagTekUSCRA, archive=mtscra.jar, dll_auto_update=Yes,
codebase=http://localhost/SCRA/}
security: Grant socket perm for http://localhost/SCRA/mtscra.jar :
java.security.Permissions@10ac287 (
 ("java.net.SocketPermission" "localhost" "connect,accept,resolve")
)

security: Validate the certificate chain using CertPath API
basic: Plugin2ClassLoader.getPermissions CeilingPolicy allPerms
security: Validate the certificate chain using CertPath API
security: SSV validation:
    running: 1.7.0_72
    requested: null
    range: null
    javaVersionParam: null
    Rule Set version: null
network: Created version ID: 1.7.0.72
network: Created version ID: 1.7.0.72
security: continue with running version
network: Created version ID: 1.7.0.72
network: Created version ID: 1.7
network: Created version ID: 2.2.72
basic: Applet loaded.
basic: Applet resized and added to parent container
basic: PERF: AppletExecutionRunnable - applet.init() BEGIN ; jvmLaunch
dt 1185153 us, pluginInit dt 3918768 us, TotalTime: 5103921 us
com.magtek.windows.scra.usb.MTSCRA::init: Init : START
onLibLoaded has=1
basic: Applet initialized
basic: Starting applet
basic: completed perf rollup
```

```
basic: Applet made visible
basic: Applet started
basic: Told clients applet is started
security: Javascript from a non secure page is accessing privileged
code. Consider using HTTPS protocol when using Javascript -> Java
liveconnect calls.
network: Checking for update at: https://javadl-esd-
secure.oracle.com/update/baseline.version
network: Checking for update at: https://javadl-esd-
secure.oracle.com/update/blacklist
network: Checking for update at: https://javadl-esd-
secure.oracle.com/update/blacklisted.certs
network: Connecting https://javadl-esd-
secure.oracle.com/update/blacklist with proxy=DIRECT
network: Connecting https://javadl-esd-
secure.oracle.com/update/blacklisted.certs with proxy=DIRECT
network: Connecting https://javadl-esd-
secure.oracle.com/update/baseline.version with proxy=DIRECT
network: Connecting http://javadl-esd-secure.oracle.com:443/ with
proxy=DIRECT
network: Connecting http://javadl-esd-secure.oracle.com:443/ with
proxy=DIRECT
network: Connecting http://javadl-esd-secure.oracle.com:443/ with
proxy=DIRECT
security: Loading SSL Root CA certificates from C:\Program Files
(x86)\Java\jre7\lib\security\cacerts
security: Loaded SSL Root CA certificates from C:\Program Files
(x86)\Java\jre7\lib\security\cacerts
security: Obtain certificate collection in SSL Root CA certificate
store
security: Obtain certificate collection in SSL Root CA certificate
store
security: Loading certificates from Deployment session certificate
store
security: Loaded certificates from Deployment session certificate
store
security: Checking if SSL certificate is in Deployment permanent
certificate store
security: Obtain certificate collection in SSL Root CA certificate
store
security: Obtain certificate collection in SSL Root CA certificate
store
security: Obtain certificate collection in SSL Root CA certificate
store
security: Obtain certificate collection in SSL Root CA certificate
store
security: Saving certificates in Deployment session certificate store
security: Saved certificates in Deployment session certificate store
security: Obtain certificate collection in SSL Root CA certificate
store
```

```
security: Obtain certificate collection in SSL Root CA certificate
store
security: Loading certificates from Deployment session certificate
store
security: Loaded certificates from Deployment session certificate
store
security: Obtain certificate collection in SSL Root CA certificate
store
security: Obtain certificate collection in SSL Root CA certificate
store
security: Loading certificates from Deployment session certificate
store
security: Loaded certificates from Deployment session certificate
store
network: Updating file at:
C:\Users\longv\AppData\LocalLow\Sun\Java\Deployment\security\baseline.
versions from url: https://javadl-esd-
secure.oracle.com/update/baseline.version
network: Updating file at:
C:\Users\longv\AppData\LocalLow\Sun\Java\Deployment\security\blacklist
ed.certs from url: https://javadl-esd-
secure.oracle.com/update/blacklisted.certs
network: Updating file at:
C:\Users\longv\AppData\LocalLow\Sun\Java\Deployment\security\blacklist
.dynamic from url: https://javadl-esd-
secure.oracle.com/update/blacklist
network: Created version ID: 1.7.0.72
network: Created version ID: 1.7.0.71
```

Appendix C Java Code Examples

C.1 Open Device Example

```
public void openDevice()
{
MTSCRAEventHandler  mEvent       = new MTSCRAEventHandler();
     MagTekUSCRA mMTSCRA = new MagTekUSCRA();
     mMTSCRA.init(mEvent);
     long rv = mMTSCRA.openDevice("");
     if(rv==0)
          System.out.print("SUCESS");
     else
          System.out.print("FAIL");
}
```

C.2 Close Device Example

```
public void closeDevice()
{
MTSCRAEventHandler  mEvent       = new MTSCRAEventHandler();
     MagTekUSCRA mMTSCRA = new MagTekUSCRA();
     mMTSCRA.init(mEvent);
     long rv = mMTSCRA. closeDevice ();
     if(rv==0)
          System.out.print("SUCESS");
     else
          System.out.print("FAIL");
}
```

Appendix D ARQC Message Format

This section gives the format of the ARQC Message delivered in the ARQC Message notification. The output is controlled by Property 0x68 – EMV Message Format. There are currently 2 selectable formats: Original and DynaPro. It is a TLV object with the following contents.

Original Format:

```
FD<len>/* container for generic data */
        DFDF25(IFD Serial Number)<len><val>
        FA<len>/* container for generic data */
                <tags defined by DFDF02 >
                  . Note: Sensitive Data cannot be defined in DFDF02
                  .
                DFDF4D(Masked T2 ICC Data)
                DFDF52 - Card Type Used
                F8<len>/* container tag for encrypted data */
                        DFDF56(Encrypted Transaction Data KSN)<len><val>
                        DFDF57(Encrypted Transaction Data Encryption Type)<val>

                        FA<len>/* container for generic data */
                                DF30(Encrypted Tag 56 TLV, T1 Data)<len><val>
                                DF31(Encrypted Tag 57 TLV, T2 Data)<len><val>
                                DF32(Encrypted Tag 5A TLV, PAN)<len><val>
                                DF35(Encrypted Tag 9F1F TLV, T1 DD)<len><val>
                                DF36(Encrypted Tag 9F20 TLV, T2, DD)<len><val>
                                DF37(Encrypted Tag 9F61 TLV, T2 CVC3)<len><val>
                                DF38(Encrypted Tag 9F62 TLV, T1,PCVC3)<len><val>
                                DF39(Encrypted Tag DF812A TLV, T1 DD)<len><val>
                                DF3A(Encrypted Tag DF812B TLV, T2 DD)<len><val>
                                DF3B(Encrypted Tag DFDF4A TLV, T2 ISO Format)<len><val>
                                DF40(Encrypted Value only of DFDF4A, T2 ISO Format)<len><val>
```

DynaPro Format:

```
F9<len>/* container for MAC structure and generic data */
        DFDF54(MAC KSN)<len><val>
        DFDF55(MAC Encryption Type)<len><val>
        DFDF25(IFD Serial Number)<len><val>
        FA<len>/* container for generic data */
                70<len>/*container for ARQC */
                        DFDF53<len><value>/*fallback indicator */
                        5F20<len><value>/*cardholder name */
                        5F30<len><value>/*service code */
                        DFDF4D<len><value>/* Mask T2 ICC Data */
                        DFDF52<len><value>/* card type */
                        F8<len>/*container tag for encryption */
                                DFDF59(Encrypted Data Primitive)<len><Encrypted Data val (Decrypt
                                data to read tags)>
                                DFDF56(Encrypted Transaction Data KSN)<len><val>
                                DFDF57(Encrypted Transaction Data Encryption Type)<val>
                                DFDF58(# of bytes of padding in DFDF59)<len><val>
(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes, always set to zeroes)
```

The Value inside tag DFDF59 is encrypted and contains the following after decryption:

```
              FC<len>/* container for encrypted generic data */
                       <tags defined by DFDF02 >
                     .
                     .
```

Appendix E ARQC Response (from online processing)

This section gives the format of the data for the Online Processing Result / Acquirer Response message. This request is sent to the reader in response to an ARQC Message notification from the reader. The output is controlled by Property 0x68 – EMV Message Format. There are currently 2 selectable formats: Original and DynaPro. It is a TLV object with the following contents.

Original format:

```
F9<len>/* container for ARQC Response data */
      DFDF25 (IFD Serial Number)<len><val>
      FA<len>/* Container for generic data */
            70<len>/* Container for ARQC */
            8A<len> approval
            Further objects as needed...
```

DynaPro format:

```
F9<len>/* container for MAC structure and generic data */
      DFDF54 (MAC KSN)<len><val>
      DFDF55 (Mac Encryption Type><len><val>
      DFDF25 (IFD Serial Number)<len><val>
      FA<len>/* Container for generic data */
            70<len>/* Container for ARQC */
            8A<len> approval
(ARQC padding, if any, to be a multiple of 8 bytes)
CBC-MAC (4 bytes, use MAC variant of MSR DUKPT key that was used in ARQC request, from
message length up to and including ARQC padding, if any)
```

Appendix F Transaction Result Message – Batch Data Format

This section gives the format of the data the device uses to do completion processing. The output is controlled by Property 0x68 – EMV Message Format. There are currently 2 selectable formats: Original and DynaPro. It is a TLV object with the following contents.

Original Format:

```
FE<len>/* container for generic data */
       DFDF25(IFD Serial Number)<len><val>
       FA<len>/* container for generic data */
             F0<len>/* Transaction Results */
                    F1<len>/* container for Status Data */
                    … /* Status Data tags */
                           DFDF1A - Transaction Status (See DFDF1A descriptions)
                           DFDF1B - Additional Transaction Information (always 0)
                           DFDF52 - Card Type Used

                    F2<len>/* container for Batch Data */
                    … /* Batch Data tags defined in DFDF17 */
                    …/* Note: Sensitive Data cannot be defined in DFDF17*/

                    F3<len>/* container for Reversal Data, if any */
                    … /* Reversal Data tags defined in DFDF05 */
                    …/* Note: Sensitive Data cannot be defined in DFDF05*/

                    F7<len>/* container for Merchant Data */
                    … /* < Merchant Data tags */

                    F8<len>/* container tag for encrypted data */
                           DFDF56(Encrypted Transaction Data KSN)<len><val>
                           DFDF57(Encrypted Transaction Data Encryption Type)<val>

                    FA<len>/* container for generic data */
                           DF30(Encrypted Tag 56 TLV, T1 Data)<len><val>
                           DF31(Encrypted Tag 57 TLV, T2 Data)<len><val>
                           DF32(Encrypted Tag 5A TLV, PAN)<len><val>
                           DF35(Encrypted Tag 9F1F TLV, T1 DD)<len><val>
                           DF36(Encrypted Tag 9F20 TLV, T2, DD)<len><val>
                           DF37(Encrypted Tag 9F61 TLV, T2 CVC3)<len><val>
                           DF38(Encrypted Tag 9F62 TLV, T1,PCVC3)<len><val>
                           DF39(Encrypted Tag DF812A TLV, T1 DD)<len><val>
                           DF3A(Encrypted Tag DF812B TLV), T2 DD<len><val>
                           DF3B(Encrypted Tag DFDF4A TLV, T2 ISO Format)<len><val>
                           DF40(Encrypted Value only of DFDF4A, T2 ISO
                           Format)<len><val>
```

F.1 DFDF1A Transaction Status Return Codes

0x00 = Approved
0x01 = Declined
0x02 = Error
0x10 = Cancelled by Host
0x1E = Manual Selection Cancelled by Host
0x1F = Manual Selection Timeout
0x21 = Waiting for Card Cancelled by Host
0x22 = Waiting for Card Timeout
0x23 = Cancelled by Card Swipe
0xFF = Unknown

Appendix F - Transaction Result Message – Batch Data Format

DynaPro Format:

```
F9<len>/* container for MAC structure and generic data */
        DFDF54(MAC KSN)<len><val>
        DFDF55(MAC Encryption Type)<len><val>
        DFDF25(IFD Serial Number)<len><val>
        FA<len>/* container for generic data */
                F0<len>/* Transaction Results */
                        F1<len>/* container for Status Data */
                                … /* Status Data tags */
                        F8<len>/* container tag for encryption */
                                DFDF59(Encrypted Data Primative)<len><Encrypted
                        Data val (Decrypt data to read tags)>
                                DFDF56(Encrypted Transaction Data KSN)<len><val>
                                DFDF57(Encrypted Transaction Data Encryption Type)<val>
                                DFDF58(# of bytes of padding in DFDF59)<len><val>
                        F7<len>/* container for Merchant Data */
                                … /* < Merchant Data tags */
(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes, always set to zeroes)
```

Appendix G Supported Device Features

Feature / Product	cDynamo	DynaMAX	eDynamo	iDynamo 5	iDynamo 5 (Gen II)	iDynamo 6	kDynamo	sDynamo	tDynamo	uDynamo
MSR Swipe	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
MSR Insert	N	N	N	N	N	N	N	N	N	N
MSR 3 Tracks	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
MSR Disable	Y	N	N	Y	N	N	N	N	N	N
MSR Swap Tracks 1/3	N	N	N	N	N	N	N	N	N	N
MSR Embedded V5 Head	N	N	N	N	Y	Y	Y	Y	Y	N
MSR Configurabe MSR Variants	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
MSR Configurable MP Variants	N	Y	Y	N	N	Y	Y	N	Y	N
MSR SureSwipe	N	Y	Y	N	N	N	N	N	N	N
MSR JIS Capable	Y	N	N	Y	N	N	N	Y	N	N
SHA-1	N	Y	Y	N	N	N	N	N	N	N
SHA-256	N	N	N	N	N	N	N	N	N	N
Configurable SHA	N	Y	Y	N	N	N	N	N	N	N
Configurable Encryption Algorithm	N	N	N	N	N	Y	N	N	N	N
Set Mask Service Code	N	N	N	N	N	N	N	N	N	N
Never Mask Service Code	N	N	Y	Y	Y	Y	Y	Y	Y	N
MagneSafe 2.0	N	N	Y	N	N	N	N	N	N	N
EMV Contact	N	N	Y	N	N	Y	Y	N	Y	N
EMV Contactless	N	N	N	N	N	Y	Y	N	Y	N
EMV Offline ODA	N	N	Y	N	N	N	N	N	N	N
EMV MSR Flow	N	N	N	N	N	Y	Y	N	Y	N
EMV Contact Quick Chip	N	N	Y	N	N	Y	Y	N	Y	N
EMV Contactless Quick Chip	N	N	N	N	N	Y	Y	N	Y	N
External PIN Accessory Support	N	N	N	N	N	Y	N	N	N	N
Keypad Entry	N	N	N	N	N	N	N	N	N	N
Fixed Key	N	N	N	N	N	N	N	N	N	N
Secondary DUKPT Key	N	Y	Y	N	N	N	N	N	N	Y
Power Mgt Scheme (PM#)	N	2	3	N	N	7	5	N	5	4
Battery-Backed RTC	N	N	Y	N	N	N	N	N	N	N
OEM Features	N	N	N	N	N	N	N	N	N	N
Transaction Validation	N	N	N	N	N	N	N	N	N	N

Appendix G - Supported Device Features

Feature / Product	cDynamo	DynaMAX	eDynamo	iDynamo 5	iDynamo 5 (Gen II)	iDynamo 6	kDynamo	sDynamo	tDynamo	uDynamo
Display	N	N	N	N	N	N	N	N	N	N
Multi-Language	N	N	Y	N	N	Y	Y	N	Y	N
Tamper	N	N	Y	N	N	N	N	N	N	N
Extended Commands	N	N	Y	N	N	Y	Y	N	Y	N
Extended Notifications	N	N	Y	N	N	Y	Y	N	Y	N
Dual USB Ports	N	N	N	N	N	Y	N	N	Y	N
Pairing Modes	N	N	Y	N	N	N	N	N	Y	N
Custom Advertising	N	N	Y	N	N	N	N	N	Y	N
Configurable Lightning FID	Y	N	N	N	Y	Y	Y	N	N	N
Auxiliary Ports	N	N	N	N	N	N	N	N	N	N
External LED Control	N	N	N	N	N	N	N	N	N	N
Encrypt Bulk Data (b)	120	24	24	120	N	N	N	N	N	24