Telium Retail Base Application (RBA) Developer's Guide Rev 17.6

DIV350779

02/15/2019

Ingenico Inc. - 3025 Windward Plaza, Suite 600 - Alpharetta, GA 30005

Telium Retail Base Application (RBA) Developer's Guide Rev 17.6

Part Number DIV350779

Customer Service Centers:

Ingenico Canada Ltd.
5180 Orbitor Drive, 2nd Floor
Toronto, Ontario L4W 5L9
Tel: 905.212.9464
Fax: 905.212.9155
www.ingenico-us.com
Ingenico, Inc.
3025 Windward Plaza, Suite 600
Alpharetta, GA 30005

North America Customer Support

Tel: 888.900.8221
Fax: 905.795.9343
Email: customersfirst.us@ingenico.com

Care has been taken to ensure that the content of this document is as accurate as possible. In compliance with P2PE requirements, INGENICO updates this document when changes occur, annually, or when P2PE requirements change. INGENICO, however, declines any responsibility for inaccurate, incomplete or outdated information. The contents of this document may change from time to time without prior notice, and do not create, specify, modify, or replace any new or prior contractual obligations agreed upon in writing between INGENICO and the user. INGENICO is not responsible for any use of this device, which would be inconsistent with the present document. All trademarks used in this document remain the property of their rightful owners.

Find the latest version of this guide on the Ingenico Developer Portal: https://developer.ingenico.us/

Table of Contents

1 Introduction

The Retail Base Application (RBA) is recommended to use as a plug-and-play application with a point-of-sale (POS) system, that conforms to the standard IBMEFT protocol. All additions to the standard IBMEFT protocol are described in this guide; however, to take advantage of the iSC series signature-capture features, additional code must be added to the POS application (see Retrieval Using Get Variable).

1.1 Terminals

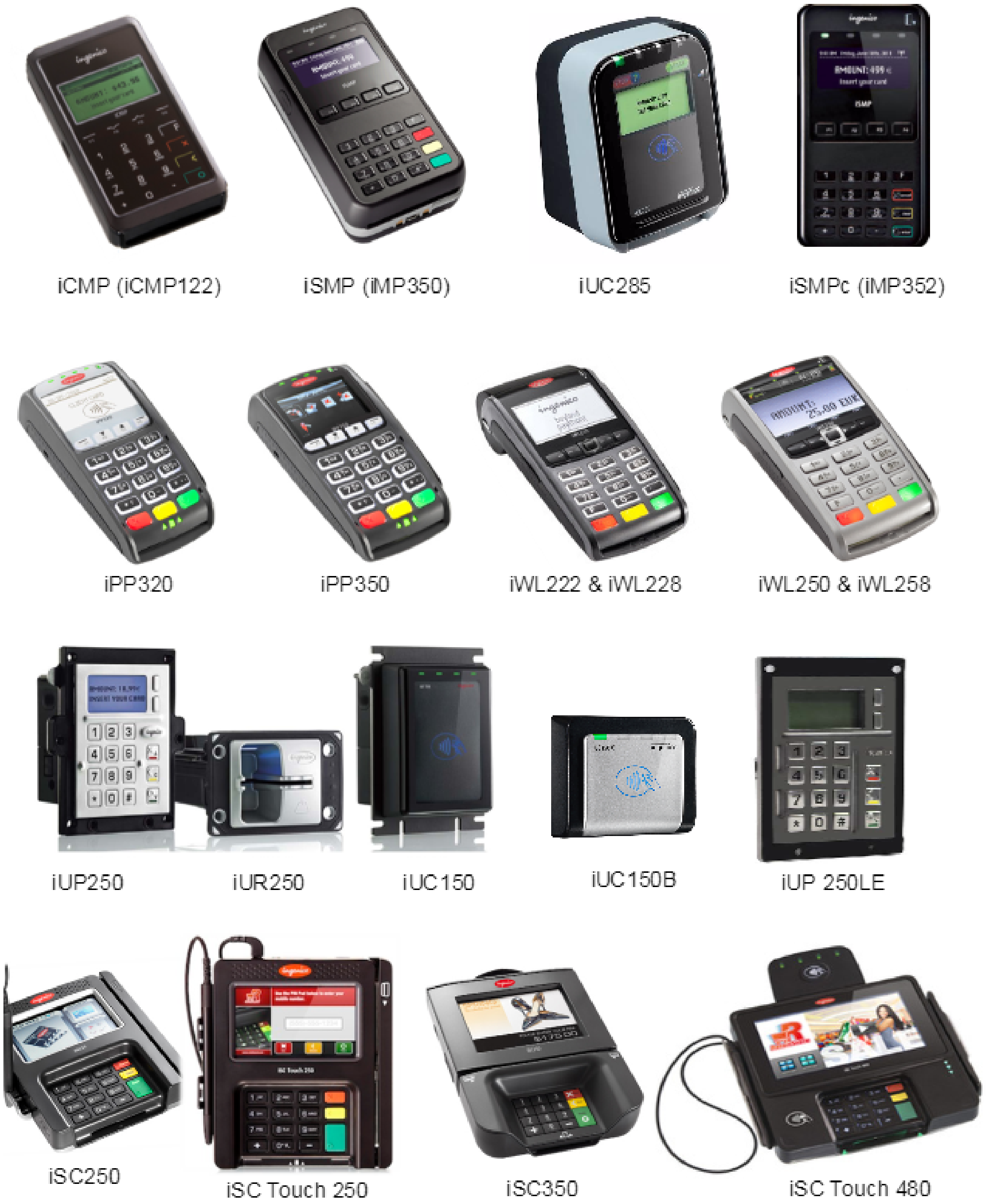
The following Ingenico Telium2 terminals are covered in this guide :

- iCMP (also referred to as iCM122)
- iSMP (also referred to as iMP350)
- iSMPc (also referred to as iMP352)
- iSMP V4 (also referred to as iSMP4)*
- iPP320
- iPP320 V4
- iPP350
- iPP350 V4
- iSC250 and iSC Touch 250
- iSC350
- iSC Touch 480
- iUN2xx (includes iUP250 and iUR250 and/or iUC150/iUC150B)
- iUN V4 (includes iUP250LE and iUR250)
- iUC285 (RBA supports on-demand mode only on an iUC285 terminal)
- iWL222
- iWL228
- iWL250
- iWL258

1.1.1 Terminal Notes

- All V4 terminals automatically reboot after 24 hours run time for PCI PTS compliance.
- *Though barcode buttons are present on all iSMP4 terminals, only those with a reader lens on the back side contain a barcode reader.
- The product name is the terminal ID, which displays when using the 07.x Unit Data Request and 08.x Health Stat messages.
- All references to the iSC250 terminal throughout this document are relevant to both the iSC250 and iSC Touch 250, unless otherwise stated. All references to the iSC480 pertain to the iSC Touch 480.

The following image gallery illustrates the Telium 2 terminal suite:



iCMP (iCMP122) iSMP (iMP350) iUC285 iSMPc (iMP352)

iPP320 iPP350 iWL222 & iWL228 iWL250 & iWL258

iUP250 iUR250 iUC150 iUC150B iUP 250LE

iSC250 iSC Touch 250 iSC350 iSC Touch 480

Image Gallery of Telium 2 Terminals

The application supports the following functions:

- Credit
- Debit
- Electronic Benefits Transfer (EBT)
- Electronic Signature Capture (iSC250/iSC350/iSC480 only)
- Item-Scrolling
- Customer Graphics Display
- Advertising
- Personal Messaging
- Surveys
- Loyalty Programs
- Internal/External BIN Range Checking
- Contactless Card Reader (optional hardware module)
- Cross-Selling
- Instant Credit
- Electronic Couponing
- Time and Attendance
- Hospitality
- Electronic ACH
- Frequent Shopper
- Transaction Data Encryption

1.2 About This Guide

This guide provides a full overview of the Retail Base Application, including an explanation of all the tools used to configure it.

It addresses various customer requirements and describes a global approach to using the communication messages described in Host Interface Messages, as well as the needs of customers with different point-of-sale (POS) environments. Encryption methods and related information can be found in the Additional Features section. The Implementing EMV section describes the transaction sequence, host interface messages, transaction flow, configuration, and other EMV features. The Managing Keys section includes information on such features as offline remote key injection, voice referral, and contactless key card. The following environments can integrate to RBA:

- NCR Register / DOS environment
- IBM 4680/4690 Register environment
- IBM 4694 Register / Windows NT environment
- Windows XP, 7, or 8 operating system
- Mac OS 10.6 (Snow Leopard) or later operating system

For additional information pertaining to the operation of your Telium terminal, refer to the corresponding user’s guide, which explains how to download the software package, including the binary data, parameters, and Telium operating system.

1.3 Definitions

Term	Definition
DFS	Data File System
ECR	Electronic Cash Register.
EFT	Electronic Funds Transfer
Financial Transaction	Refers to processes executed between two hard reset commands: 10.x or equivalent of the hard reset message.
Form File	Refers to an HTML-format file (*.K3Z) used to position and format text, buttons and images used for standard screens on Ingenico’s Telium terminals.
Host Interface	A communications interface that connects the terminal to the POS equipment, which connects to the host computer (also called an in-store system, POS or Point of Sale system, or register).
MSR	Magnetic stripe reader.
OS	Operating system.
Spin the BIN	IBM-specific terminology for the BIN lookup process (also known as PIN Encouragement).
Terminal	See definition for Telium Terminals.
POS	Point-of-sale system or device. Sometimes referred to as an Electronic Cash Register (ECR).
Prompt File	File referenced by form building utility to load button text and prompts.
RBA	Retail Base Application.
TDA	Telium Download Application.

Term	Definition
Telium Terminals	For the purposes of this document, refers to the Ingenico: • iCM122 • iMP350 • iSMP V4 • iMP352 • iPP320 • iPP320 V4 • iPP350 • iPP350 V4 • iSC250 • iSC Touch 250 • iSC350 • iSC Touch 480 • iUN2xx (includes iUP250, iUR250 and iUC150/iUC150B) • iUC285 • iWL222, iWL228 • iWL250, iWL258

1.4 Copyright Notice for Lato and Crimson Fonts

1.4.1 SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

1.4.1.1 PREAMBLE

The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

1.4.1.2 DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation. "Reserved Font Name" refers to any names specified as such after the copyright statement(s). "Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s). "Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment. "Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

1.4.1.3 PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1. Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.
2. Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.
3. No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.
4. The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.
5. The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

1.4.1.4 TERMINATION

This license becomes null and void if any of the above conditions are not met.

1.4.1.5 DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

2 Getting Started

This section describes the integration kit contents, the minimum requirements, and more information you need to get started using RBA.

2.1 Minimum Requirements for RBA Customization

To modify RBA configuration settings, you need the following tools and equipment:

- Text file editor that does not insert hidden characters (for viewing and editing `config.dfs` and prompts files)
- Windows PC equipped with one of the following types of connections:
    - RS-232
    - USB_HID (supported for all terminals except the iCM122, iMP350, iMP352, and iWL250)
    - Network connection to local area network with Internet gateway
    - Tailgate
- Image editing software capable of handling .bmp, .gif, .jpg, or .png formats
- Microsoft .NET Framework (required to run the RBA Testing Tool)

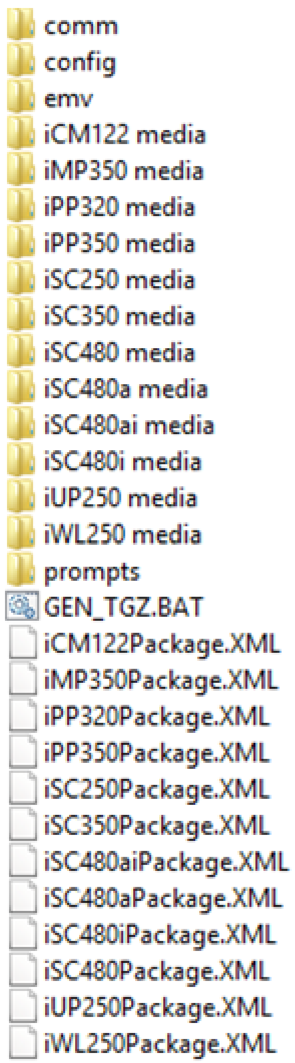
2.2 RBA Integration Kit Contents

The integration kit contains:

- Ingenico documentation
- The RBA financial application
- Telium utilities
    - RBA Testing Tool
    - Telium LLT
    - Telium Tools
        - Form Builder
        - Script Builder
        - Data Packaging Tool
        - SAT

2.2.1 Data and Parameters Package

The following file folder image shows the contents included in the Data and Parameters folder:



comm
config
emv
iCM122 media
iMP350 media
iPP320 media
iPP350 media
iSC250 media
iSC350 media
iSC480 media
iSC480a media
iSC480ai media
iSC480i media
iUP250 media
iWL250 media
prompts
GEN_TGZ.BAT
iCM122Package.XML
iMP350Package.XML
iPP320Package.XML
iPP350Package.XML
iSC250Package.XML
iSC350Package.XML
iSC480aiPackage.XML
iSC480aPackage.XML
iSC480iPackage.XML
iSC480Package.XML
iUP250Package.XML
iWL250Package.XML

Data and Parameters Package Structure

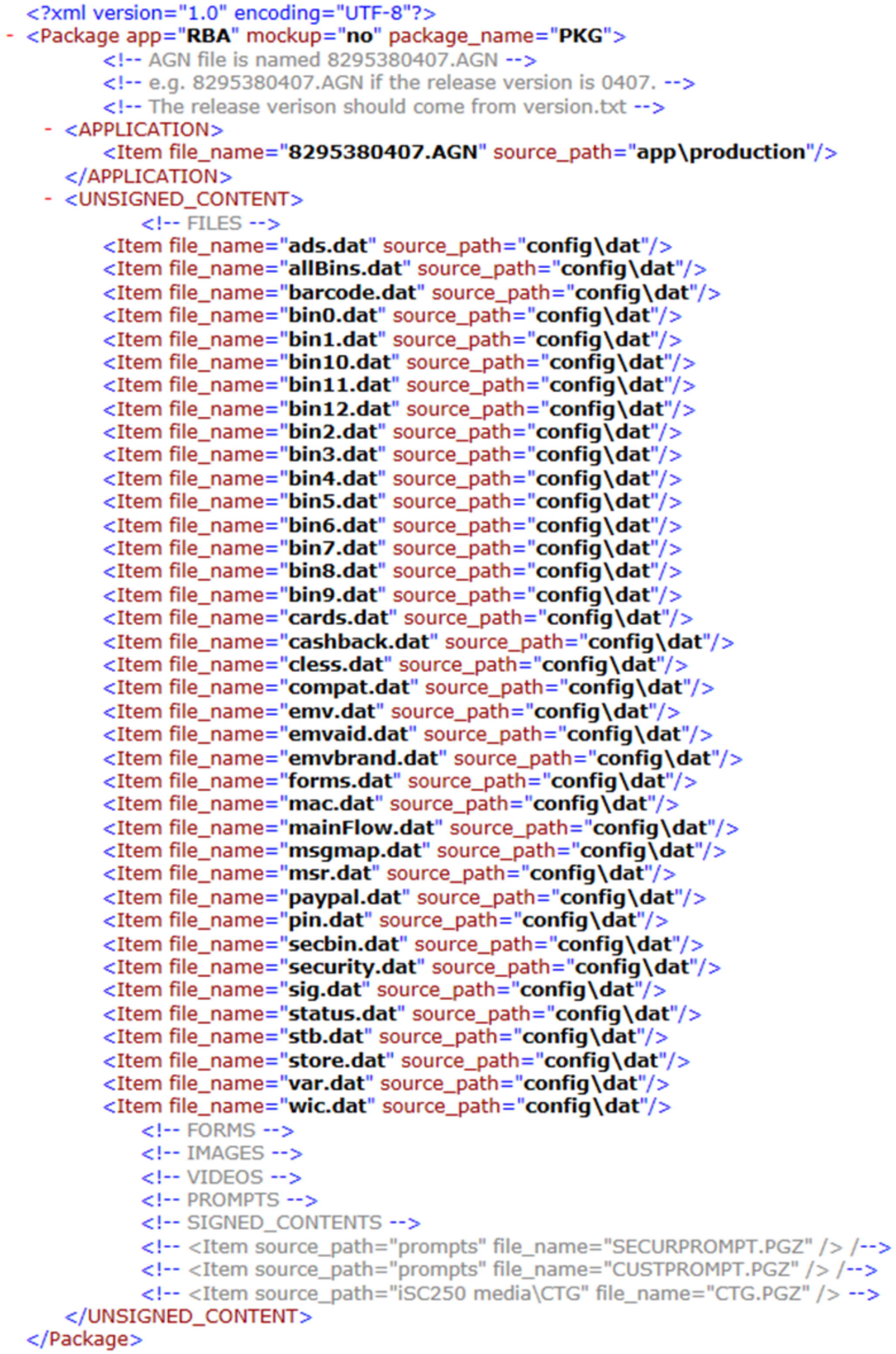
The following table describes the items contents of the Data and Parameters Package included with the integration kit.

Integration Kit Contents

Contents	Description
comm folder	Contains data files and TDA.XML files to set terminals to use specific communication types.

Contents	Description
config folder	Contains the following: • *.DAT files • config.dfs parameter file • ctr, ctr_config, and ctr_trans files
emv folder	Contains configuration files for EMV contact and EMV contactless. • EMVCONTACT.XML • EMVCLESS.XML
Terminal-specific media folders	Contain images and form (*K3Z) files.
Terminal-specific **`multimedia`** folders	Contain audio and video resources.
prompts folder	Contains prompts. • **`prompt.xml`** – non-secure application prompts. • **`custprompt.xml`** – non-secure user-defined prompts for application customization. • **`tc1.xml`** – Terms and Conditions verbiage for use in tc.k3z. • **`securprompt.xml`** – secure application prompts.
Terminal-specific files	**`Package definition *.XML file`**, also known as the manifest, specifies all the information needed to package *._GZ files that will be used to load RBA content onto the terminal: • Package name • Files to package • The file path within the original application package folder structure • The file intended location on the terminal (optional)
`PackageGZ batch file`	Generates *._GZ files for loading RBA to the corresponding terminal using the information specified in that terminal's manifest file.
`PackageLLT batch file`	Initiates LLT download of the GZ file.
`PackageEFT batch file`	Generates the EFT file with the Packaging Toll using the terminal's manifest file.

An example of an RBA manifest file (`iSC250Package.XML`):



```
  <?xml version="1.0" encoding="UTF-8"?>
- <Package app="RBA" mockup="no" package_name="PKG">
        <!-- AGN file is named 8295380407.AGN -->
        <!-- e.g. 8295380407.AGN if the release version is 0407. -->
        <!-- The release verison should come from version.txt -->
  - <APPLICATION>
        <Item file_name="8295380407.AGN" source_path="app\production"/>
    </APPLICATION>
  - <UNSIGNED_CONTENT>
           <!-- FILES -->
        <Item file_name="ads.dat" source_path="config\dat"/>
        <Item file_name="allBins.dat" source_path="config\dat"/>
        <Item file_name="barcode.dat" source_path="config\dat"/>
        <Item file_name="bin0.dat" source_path="config\dat"/>
        <Item file_name="bin1.dat" source_path="config\dat"/>
        <Item file_name="bin10.dat" source_path="config\dat"/>
        <Item file_name="bin11.dat" source_path="config\dat"/>
        <Item file_name="bin12.dat" source_path="config\dat"/>
        <Item file_name="bin2.dat" source_path="config\dat"/>
        <Item file_name="bin3.dat" source_path="config\dat"/>
        <Item file_name="bin4.dat" source_path="config\dat"/>
        <Item file_name="bin5.dat" source_path="config\dat"/>
        <Item file_name="bin6.dat" source_path="config\dat"/>
        <Item file_name="bin7.dat" source_path="config\dat"/>
        <Item file_name="bin8.dat" source_path="config\dat"/>
        <Item file_name="bin9.dat" source_path="config\dat"/>
        <Item file_name="cards.dat" source_path="config\dat"/>
        <Item file_name="cashback.dat" source_path="config\dat"/>
        <Item file_name="cless.dat" source_path="config\dat"/>
        <Item file_name="compat.dat" source_path="config\dat"/>
        <Item file_name="emv.dat" source_path="config\dat"/>
        <Item file_name="emvaid.dat" source_path="config\dat"/>
        <Item file_name="emvbrand.dat" source_path="config\dat"/>
        <Item file_name="forms.dat" source_path="config\dat"/>
        <Item file_name="mac.dat" source_path="config\dat"/>
        <Item file_name="mainFlow.dat" source_path="config\dat"/>
        <Item file_name="msgmap.dat" source_path="config\dat"/>
        <Item file_name="msr.dat" source_path="config\dat"/>
        <Item file_name="paypal.dat" source_path="config\dat"/>
        <Item file_name="pin.dat" source_path="config\dat"/>
        <Item file_name="secbin.dat" source_path="config\dat"/>
        <Item file_name="security.dat" source_path="config\dat"/>
        <Item file_name="sig.dat" source_path="config\dat"/>
        <Item file_name="status.dat" source_path="config\dat"/>
        <Item file_name="stb.dat" source_path="config\dat"/>
        <Item file_name="store.dat" source_path="config\dat"/>
        <Item file_name="var.dat" source_path="config\dat"/>
        <Item file_name="wic.dat" source_path="config\dat"/>
           <!-- FORMS -->
           <!-- IMAGES -->
           <!-- VIDEOS -->
           <!-- PROMPTS -->
           <!-- SIGNED_CONTENTS -->
           <!-- <Item source_path="prompts" file_name="SECURPROMPT.PGZ" /> /-->
           <!-- <Item source_path="prompts" file_name="CUSTPROMPT.PGZ" /> /-->
           <!-- <Item source_path="iSC250 media\CTG" file_name="CTG.PGZ" /> -->
    </UNSIGNED_CONTENT>
  </Package>
```

iSC250Package.XML File

2.2.2 Ingenico Documentation

Ingenico's RBA Integration Kit contains the following types of documentation:

- DIV350779 Telium RBA User's Guide (this document)
- User's guides for supported Telium terminals:
  - iPP2xx
  - iPP3xx
  - iPP3xx V4
  - iSC250
  - iSC350
  - iSC480
  - iUP250
  - iUC285
  - iWL250
- Documentation for Ingenico Telium utilities provided with the kit:
  - LLT
  - Form Builder
  - Script Builder
  - Data Packaging Tool

2.2.3 RBA-Related Utilities

Ingenico's RBA Integration Kit contains the following RBA-related Telium utilities:

- RBA Testing Tool
- Telium LLT
- Form Builder
- Script Builder
- Data Packaging Tool
- SAT

2.3 RBA Content Locations

When RBA is installed on a terminal, each type of application file is moved to a specific path in the terminal's memory. The following table shows the path that each application file is installed to:

Files and their Parent Directories

File Name	Location on Terminal Memory
Images	`/HOST`
`SECURPROMPT.XML`	`/F_SECURITY_APP`

File Name	Location on Terminal Memory
PROMPT.XML	/HOST
CUSTPROMPT.XML	/F_SECURITY_APP
CTG graphics	/F_SECURITY_APP/CTG
TC1.XML	/HOST
*.DAT files	/HOST
*.K3Z files	/HOST
BOOT.HTM	/F_SECURITY_APP
Templates	/F_SECURITY_APP
TRACE.XML	/F_SECURITY_APP

2.4 Signing

All applications, data files, images, videos, and form files must be signed by Ingenico before they can be used in a production terminal, with the exception of unpackaged files sent to HOST (refer to the 62.x File Write for more information).

2.5 Drivers and Tools

2.5.1 Ingenico iConnectEFT Constants

The iConnectEFT constant identifies enumerators in the Application Programing Interface (API). Refer to the following table of example iConnectEFT constants used in the 24.x Form Entry Request (On-Demand) message.

Example iConnectEFT Constants Used in 24.x: Form Entry Request (On-Demand) Message

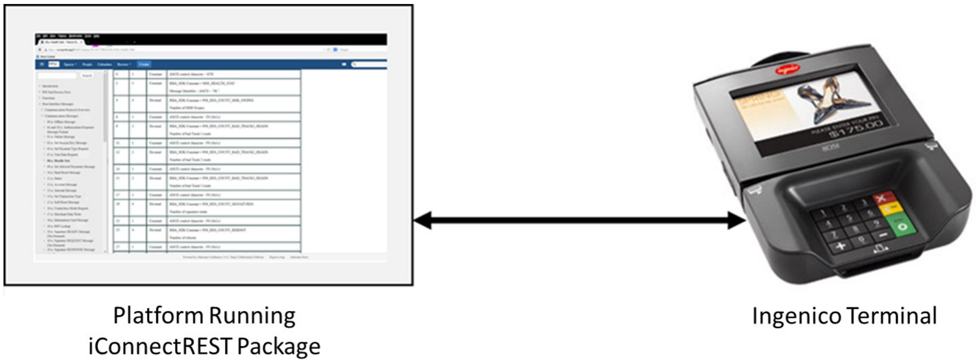
iConnectEFT Constant	Description
M24_FORM_ENTRY_REQUEST_ON_DEMAND	24.x Form Display On-Demand request for customer text entry.
P24_REQ_FORM_NUMBER	Form number or form name in REQUEST message.
P24_REQ_TEXT_ELEMENTID	Text element ID in REQUEST message.
P24_REQ_PROMPT_IDX	Prompt index number in REQUEST message.
P24_REQ_BUTTONID	Button ID in REQUEST message.

iConnectEFT Constant	Description
P24_REQ_BUTTON_STATE	Button state in REQUEST message.
P24_RES_EXIT_TYPE	Exit type in RESPONSE message.
P24_RES_KEYID	Key ID in RESPONSE message.
P24_RES_BUTTONID	Button ID in RESPONSE message.
P24_RES_BUTTON_STATE	Button state in RESPONSE message.

2.5.2 Ingenico iConnectREST

2.5.2.1 Overview of iConnectREST

Ingenico iConnectREST is a collection of software packages to use in the development of web applications that communicate with Ingenico Telium terminals using the RESTful application programming interface (API).



Using iConnectREST to Communicate with Telium Terminal

2.5.2.2 Supported Terminals and Connection Methods

The following table describes the iConnectREST-supported terminals and connection methods:

iConnectREST Supported Payment Terminals and Connection Methods

Telium Terminals	Server Location	Connection Methods
iMP350 iMP352	iOS	Bluetooth
	Telium	n/a
	Windows	Bluetooth (via iPassThru), USB-CDC, USB-HID
iPP320 iPP350	iOS	Ethernet, USB-CDC, USB-HID
	Telium	Ethernet

Telium Terminals	Server Location	Connection Methods
	Windows	Ethernet, USB-CDC, USB-HID
iSC250 iSC Touch 250 iSC350 iSC Touch 480	iOS	USB-CDC, USB-HID
	Telium	Ethernet
	Windows	USB-CDC, USB-HID
iUP250	iOS	USB-CDC, USB-HID
	Telium	Ethernet
	Windows	Ethernet, USB-CDC, USB-HID
iWL220	iOS	Bluetooth, Wi-Fi
	Telium	Wi-Fi
	Windows	Bluetooth, Wi-Fi

2.5.3 RBA Testing Tool

The RBA Testing Tool is a Windows application that allows for RBA messages to be sent to, viewed, and received from the Telium 2 terminal running RBA. The RBA Testing Tool has the ability to simulate a Point of Sale (POS) and authorizing host to complete a purchase transaction using many payment methods. The RBA Testing Tool allows for individual RBA commands to be sent and has the ability to show the transaction log and parse message to help an integrator developer their interface from their application to the Telium device.

Refer to the RBA Testing Tool documentation for more information.

2.6 LCD Screen Preservation

Terminals use backlit LCD screens to display transaction and advertising information. Take the following precautions to minimize image persistence, which occurs when an image is displayed for extended periods, leaving a residual impression of the image on the screen:

- Do not allow a still image to be displayed for more than four hours.
- Use a screensaver with a black or medium-gray background when the terminal has been inactive for 10 minutes.
- Power down the terminal when not in use.

3 Telium Terminal Information

The VID and PID settings required for USB communications are compiled in a table that addresses both CDC and HID modes. To establish USB communications with a terminal, a POS device must be configured to reference the terminal using the correct VID and PID settings. Refer to the table on the VID and PID Settings for HID and CDC Communications page for the required settings. Also refer to the Communication Settings page for more information on communication settings.

3.1 Communication Settings

Refer to the **Telium Support Guide** for more information on using the Telium Manager and Telium Download Application (TDA) for setting up the communication type for your terminal. The following table lists the supported communication types for each terminal.

Supported communication types include:

- RS-232 (19200, 8, N, 1)
- RS-232 (115200, 8, N, 1)
- USB HID
- USB<>SerialConv
- MagicBox
- Ethernet (DHCP)
- Ethernet (Static) - See note 1
- Tailgate
- Bluetooth
- Wi-Fi

Refer to the Communications Supported per Terminal Model section, which lists the available communication options for each terminal model.

For more information on Bluetooth and Wi-Fi settings, refer to the **Bluetooth Settings** section of the **Telium Support Guide,** which includes the following information:

- Associating the iWL250 Terminal with Bluetooth Cradle/Base
- Bluetooth Pairing for the iWL250 and iSMP Companion

Also refer to the **Configuring Device Communication Settings** section of the **Telium Support Guide** for information on the Jungo Driver.

Customers can request other types of RS-232 settings from their Ingenico account representative.

Note 1
If the terminal is configured with an IP Address of '0.0.0.0' or '192.168.002.002', then the user will be prompted for an IP Address, Subnet Mask, and Gateway Address.

3.1.1 Communications Supported per Terminal Model

The following table shows the interfaces supported by each terminal as they appear in their respective menus. To access the communication menu easily, press the **F** key four times.

Supported Communication Types by Terminal

Terminal(s)	Supported Communication Types	Default Setting	Notes
• iPP320/ iPP320 V4 • iPP350/ iPP350 V4 • iSC250 • iSC480	• Serial • Ethernet • USB-HID • USB<>Serial Conv • Tailgate • MAGICBOX Serial	USB<>Serial Conv	iPP-series terminals configured for Ethernet display *Network not available. Restarting* . . . at boot unless either: • Connected to Ethernet • Connected to the commbox (the terminal always considers Ethernet to be connected.)
iSC350	• Serial • Ethernet • USB-HID • USB<>Serial Conv • Tailgate	USB<>Serial Conv	
iUC285 (see note 1)	• Serial • USB<>Serial Conv	USB<>Serial Conv	
iUP250	• Serial • Ethernet • USB<>Serial Conv	USB<>Serial Conv	

Terminal(s)	Supported Communication Types	Default Setting	Notes
iWL220	Two models available: - iWL222 - USB<>Serial Conv - Bluetooth - iWL228 - USB<>Serial Conv - Wi-Fi - Ethernet via Smart Base	- iWL222 - Bluetooth - iWL228 - USB<>Serial Conv	

Terminal(s)	Supported Communication Types	Default Setting	Notes
iWL250	• Two models available: ◦ iWL250 ▪ USB<>Serial Conv ▪ Bluetooth ▪ Ethernet via Smart Base ◦ iWL258	• iWL250 ◦ Bluetooth • iWL258 ◦ USB<>Serial Conv	

Terminal(s)	Supported Communication Types	Default Setting	Notes
	▪ USB<>Serial Conv ▪ Wi-Fi ▪ Ethernet via Smart Base		
iMP350 (iSMP)	• USB<>Serial • Bluetooth • Ethernet	USB<>Serial	

Terminal(s)	Supported Communication Types	Default Setting	Notes
iSMP V4	• USB<>Serial • Bluetooth • Wi-Fi • Ethernet • USB<>Wi-Case	USB<>Serial	
iMP352 (iSMPc)	• USB<>Serial Conv • Bluetooth • Ethernet	Bluetooth	
iCM122 (iCMP)	• USB<>Serial Conv • Bluetooth	Bluetooth	

From either the **Communication** or **Select Comm. Type** menu, pressing the **Clear** button three times changes the menus to include interfaces not supported by the terminal.

Note 1

The iUC2xx terminal functions exclusively in on-demand mode. Refer to Communication Messages for details.

3.1.1.1 Actions that Cannot Be Reversed

When you enter the Communications Configuration screen, the following actions cannot be undone by pressing **Cancel** to return to the offline screen:

- Bluetooth pairing
- Bluetooth unpairing
- Configuring a Wi-Fi access point
- Deleting a Wi-Fi access point

3.1.1.2 Configuring the Telium Download Application

The `TDA.XML` file includes configuration settings for the Telium Download Application (TDA). Settings are related to:

- Choice of download type (EFT or The Estate Manager)
- Specific settings for EFT downloads

- Communications parameters affecting TDA and other applications including RBA

An EFT download can be initiated from the POS when RBA is running by sending an 01. command with values for the Program Load (EFTL) and Parameter Load (EFTP) variables. See 01.x Online Message for more information.

The following table describes the configuration options. Note that the user should not change certain values because they are set by the system. These values are marked *set by the system*.

Note: Before making changes to the `TDA.XML` file, make a backup copy of the file so you can refer to the original configuration of communication settings if needed.

Item	Description
Configuration Section	
DOWNLOADTYPE	Download type: • 0 = EFT • 1 = Estate Manager (IngEstate) • 2 = PINPad Agent (not supported)
EFTDOWNLOAD	**Set by the system**. Values are: • 0 = No download completed recently • 1 = A download was completed before the last reboot
EFTERROR	**Set by the system**. Result of the last EFT download attempt: • 0 = Successful download • -1 = Failed to unpack EFT files • -2 = No message received, and download timed out • -3 = Ten invalid messages received during download • -4 = Error when both EFTL and EFTP are given as 0000 • -5 = Failure to execute pre script (not supported) • -6 = Failure to execute post script (not supported) • -7 = Unknown error
EFTLVERSION	**Set by the system.** The application version from the last successful EFT download. For example, if the online message was `01.11112222`, then the version is `1111`.
EFTPVERSION	**Set by the system**. The parameter version from the last successful EFT download. For example, if the online message was `01.11112222`, then the version is `2222`.

Item	Description
RKIVERSION	**Set by the system when an Remote Key Injection file is downloaded**. This allows a merchant to keep track of RKI files loaded to the device. See Offline Remote Key Injection (RKI) Support for more information.
MANUFACTUREID	**INGNAR** by default. Value used as the Manufacture ID in the 07.x Unit Data Request response.
PRODUCTID	Normally set to blank to allow the application to retrieve the Terminal Name.
COMMTYPE	Communications type: • Default = Uses a default based on the terminal type • Serial = RS-232 serial • Ethernet = Wired Ethernet or Wi-Fi. To enable WiFi, set COMMTYPE to Ethernet and WIFIPOWER to ON. • USB-HID • USB-CDC • USB-Base = USB to iWL base (base-to-terminal communication is Bluetooth) • Tailgate = RS-485 serial • Bluetooth • Serial(MB) = MagicBox serial • PCL-USB = PCL via USB (allows IP communications to POS application over physical USB connnection). Not supported • PCL-Bluetooth = PCL via Bluetooth (allows IP communications to POS application over physical Bluetooth connection) • PCL-Serial = PCL via Serial (allows IP communications to POS application over physical Serial connection). Not supported
	Ethernet Section (applies to both Ethernet and WiFi, except where Wi-Fi-specific settings exist in the WiFi section)
TMUPDATED	Indicates whether Telium Manager Ethernet settings should be updated: • 1 = Use current Telium Manager settings • 0 = Update Telium Manager with settings from this TDA.XML file **Always set this value to 0 when downloading a new file.**

Item	Description
IPDHCP	• 0 = Use the static IP address from the IPADDRESS parameter (DHCP is off) • 1 = Use DHCP to assign IP address
IPPORT	IP port to listen on when Ethernet communications is set to server mode
IPADDRESS	Terminal IP address to use when DHCP is off
HOSTIPPORT	Host IP port to connect to when Ethernet communications is set to client mode
HOSTIPADDRESS	Host IP address to connect to when Ethernet communications is set to client mode. It can be set to a host name. The terminal resolves the name to an IP address via the DNS server. Applicable to wired Ethernet and Wi-Fi connections
IPSUBMASK	IP subnet mask to use when DHCP is off
IPGATEWAY	IP address of the gateway to use when DHCP is off
IPDNS1	IP address of the primary DNS server to use when DHCP is off
IPDNS2	IP address of the secondary DNS server to use when DHCP is off
SERVER	Indicates whether the terminal calls the host (client mode) or the host calls the terminal (server mode). • 0 = Client mode • 1 = Server mode
IPDISPLAYINFO	• YES = Display terminal IP address on the splash screen • NO = Do not display terminal IP address on the splash screen
SSLMODE	SSL mode: • 0 = SSL is off • 1 = SSL is on
RETRYINTERVAL	Delay between disconnect detection and reconnection attempt, in ms. Applies to client mode only.
	Serial Section (Serial and Magic Box communication settings)
BAUDRATE	Baud rate: 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200, 300

Item	Description
STOPBIT	Number of stop bits: 1, 2
BITSPERBYTE	Bits per byte: 7, 8
PARITY	Parity: • None • Odd • Even
FLOWCONTROL	Flow control: • Hard = Hardware flow control • None = No flow control
	Tailgate Section
TGADDRESS	Tailgate address: options are 64h, 65h, 68h, 69h
	Bluetooth Section
BLUETOOTHMODE	Bluetooth mode: • Empty = None • iOS = Connect with an iOS device • Standard = Connect with a non-IOS device
BLUETOOTHPAIRING	**Set by the system.** The current Bluetooth pairing state: • 0 = Not paired • 1 = Pairing in progress • 2 = Paired
BLUETOOTHPIN	**Set by the system**. Encrypted pairing PIN from the most recent pairing.
	Wi-Fi Section
WIFIPOWER	Turn on the Wi-Fi radio: • On • Off To enable Wi-Fi, set COMMTYPE to Ethernet and WIFIPOWER to ON.
WIFIBOOTPROTO	Selects the WiFi IP address mode: • NONE = static IP • DHCP = use DHCP

Item	Description
WIFIIPADDRESS	Terminal IP address to use when DHCP is off
WIFISUBNETMASK	IP subnet mask to use when DHCP is off
WIFIGATEWAY	IP address of the gateway to use when DHCP is off
WIFIDNS1	IP address of the primary DNS server to use when DHCP is off
WIFIDNS2	IP address of the secondary DNS server to use when DHCP is off

The following code is an example `TDA.XML` parameter file.

```
<?xml version="1.0"?>
<APP>
   <Configuration>
       <Item name="DOWNLOADTYPE" value="0" />
       <Item name="EFTDOWNLOAD" value="0" />
        <Item name="EFTERROR" value="0" />
        <Item name="EFTLVERSION" value="0000" />
        <Item name="EFTPVERSION" value="0000" />
        <Item name="RKIVERSION" value="0000" />
        <Item name="MANUFACTUREID" value="INGNAR" />
        <Item name="PRODUCTID" value="" />
        <Item name="COMMTYPE" value="Default" />
   </Configuration>
    <Comm>
       <Ethernet>
           <Item name="TMUPDATED" value="1" />
            <Item name="IPDHCP" value="1" />
            <Item name="IPPORT" value="12000" />
            <Item name="IPADDRESS" value="000.000.000.000" />
            <Item name="HOSTIPPORT" value="6000" />
            <Item name="HOSTIPADDRESS" value="000.000.000.000" />
            <Item name="IPSUBMASK" value="255.255.255.000" />
            <Item name="IPGATEWAY" value="000.000.000.000" />
            <Item name="IPDNS1" value="000.000.000.000" />
            <Item name="IPDNS2" value="000.000.000.000" />
            <Item name="SERVER" value="1" />
            <Item name="IPDISPLAYINFO" value="YES" />
            <Item name="SSLMODE" value="0" />
            <Item name="RETRYINTERVAL" value="5000" />
       </Ethernet>
       <Serial>
            <Item name="BAUDRATE" value="115200" />
            <Item name="STOPBIT" value="1" />
            <Item name="BITSPERBYTE" value="8" />
            <Item name="PARITY" value="NONE" />
            <Item name="FLOWCONTROL" value="NONE" />
       </Serial>
        <Tailgate>
            <Item name="TGADDRESS" value="64h" />
       </Tailgate>
       <Bluetooth>
            <Item name="BLUETOOTHMODE" value="" />
            <Item name="BLUETOOTHPAIRING" value="" />
            <Item name="BLUETOOTHPIN" value="" />
       </Bluetooth>
       <Wifi>
            <Item name="WIFIPOWER" value="OFF" />
            <Item name="WIFIBOOTPROTO" value="DHCP" />
            <Item name="WIFIIPADDRESS" value="000.000.000.000" />
            <Item name="WIFISUBNETMASK" value="255.255.255.000" />
            <Item name="WIFIGATEWAY" value="000.000.000.000" />
            <Item name="WIFIDNS1" value="000.000.000.000" />
            <Item name="WIFIDNS2" value="000.000.000.000" />
       </Wifi>
   </Comm>
</APP>
```

3.1.1.3 VID and PID Settings for HID and CDC Communications

Because Ingenico terminals include a USB interface, they are assigned a USB device classification. Classifications for Ingenico Telium terminals include:

- USB CDC - Communications Device Class
- USB HID - Human Interface Device

USB interface products are also identified using a vendor ID (VID) and product ID (PID). The following table specifies the CDC and HID vendor IDs and product IDs for Ingenico Telium terminals.

CDC and HID Vendor IDs and Product IDs for Ingenico Telium Terminals

Terminal	String	Screen	Colors	MP4	Touch	Flash	Audio	CDC VID	CDC PID	HID VID	HID PID
iPP320	Ingenico iPP320	128x64	Black/ white	No	No	128m	Buzzer	0x0B00	0x0059	0x0B00	0x0071
iPP320 V4	Ingenico iPP320	128x64	Black/ white	No	No	128m	Buzzer	0x0B00	0x0059	0x0B00	0x0071
iPP350	Ingenico iPP350	320x240	4k	No	No	128m	Buzzer	0x0B00	0x0060	0x0B00	0x0072

Terminal	String	Screen	Colors	MP4	Touch	Flash	Audio	CDC VID	CDC PID	HID VID	HID PID
iPP350 V4	Ingenico iPP350	320x240	4k	No	No	128m	Buzzer	0x0B00	0x0060	0x0B00	0x0072
iSC250	Ingenico iSC250	480x272	240k	Yes	Yes	128m	Yes	0x0B00	0x0062	0x0B00	0x0074
iSC350	Ingenico iSC350	640x480	240k	Yes	Yes	128m	Yes	0x0B00	0x0061	0x0B00	0x0073
iSC480	Ingenico iSC480	800x480	262k	Yes	Yes	128m	Yes	0x0B00	0x0061	0x0B00	0x0073
iUC285	Ingenico iUC285	128x64	Black/white	No	No	128m	Buzzer	0x0B00	0x0057	NS	NS
iUP250	Ingenico iUP250	128x64	Black/white	No	No	128m	Buzzer	0x0B00	0x0057	0x0B00	0x0076
iWL222	Ingenico iWL222	128x64	Black/white	No	No	128m	Buzzer	0x0B00		NS	NS
iWL228	Ingenico iWL228	128x64	Black/white	No	No	128m	Buzzer	0x0B00		NS	NS
iWL250	Ingenico iWL250	320x240	240k	No	No	128m	Buzzer	0x0B00	0x0064	NS	NS
iWL258	Ingenico iWL258	320x240	4096	No	No	128m	Buzzer	0x0B00		NS	NS

3.1.2 Bluetooth Support

This section describes how to pair a terminal with an android or iOS device via Bluetooth using communications settings or a QR code. It also describes how iWL terminals display signal strength.

3.1.2.1 Bluetooth Pairing and Unpairing

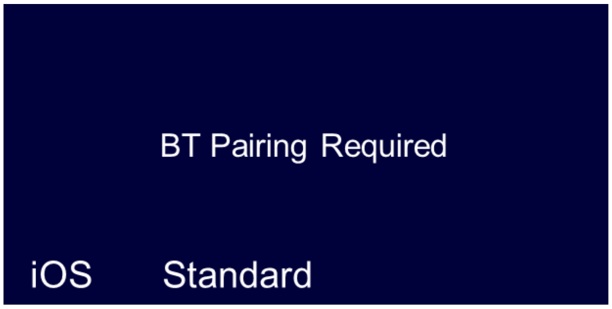
This section describes how to pair a terminal via Bluetooth with an iOS or Android device.

This section describes how to pair and unpair a terminal via Bluetooth with an iOS or Android (standard) device.

3.1.2.1.1 iOS Bluetooth Pairing

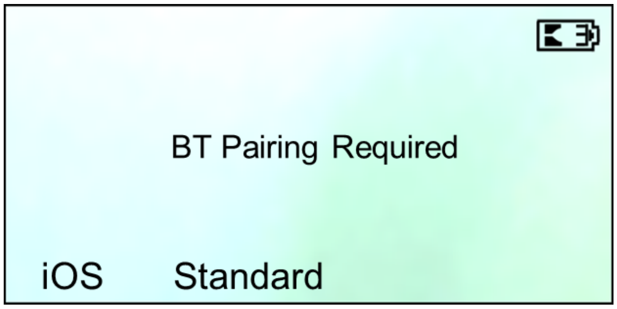
If the terminal was previously paired with a standard (non-iOS) Bluetooth device, the terminal automatically reboots to allow it to pair with an iOS device.

1. Ensure the terminal is powered on, and the iOS device has Bluetooth connectivity enabled.
   The terminal displays the Bluetooth Pairing Required screen. If not, the terminal must be unpaired (see Bluetooth Unpairing).



BT Pairing Required

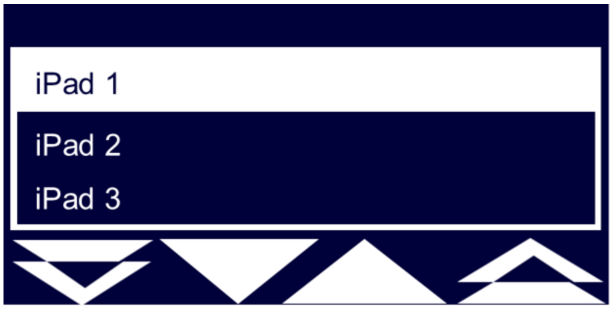
iSMPc



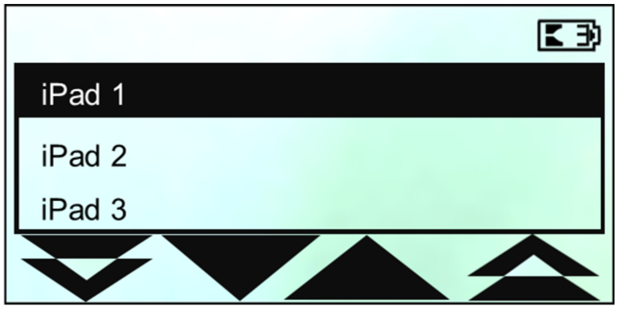
BT Pairing Required

iCMP

2. To begin the pairing process, select the iOS key (F1) on the terminal.
   a. Some iSMPc terminals are configured to support one type of Bluetooth pairing only. In this case, the iOS and Standard options illustrated are replaced with a single option that reads **Begin**.
3. The terminal displays all Bluetooth-enabled iOS devices in range:

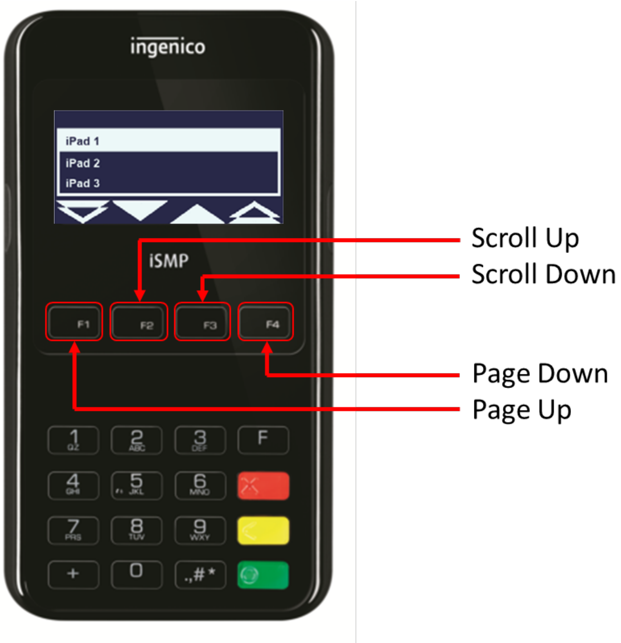


iSMPc



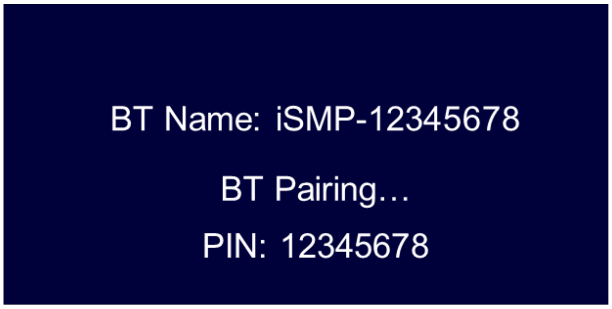
iCMP

a. Use the [F2] and [F3] keys to scroll up and down, respectively, through the list of available Bluetooth devices.
b. Use the [F1] and [F4] keys to page up and page down, respectively.



4. Highlight the Bluetooth device to pair with, and press the [Green] key:

5. The terminal displays an eight-digit, randomly generated pairing PIN:

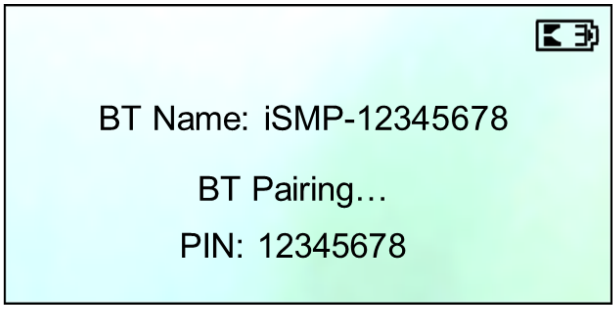


BT Name: iSMP-12345678

BT Pairing…

PIN: 12345678

iSMPc



BT Name: iSMP-12345678

BT Pairing…

PIN: 12345678

iCMP

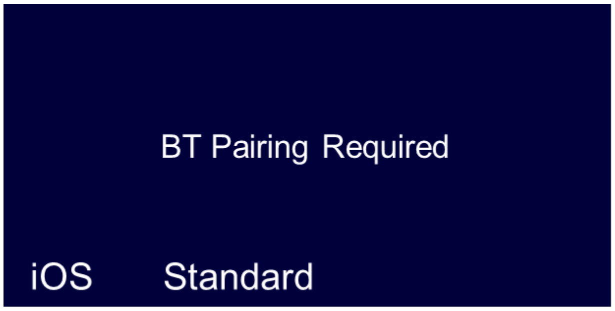
6. The host device prompts for a pairing PIN.
7. The user enters the generated PIN in the prompt and selects **Pair** on the iOS device.
8. The iOS device displays the following information and shows the status of the pairing process:
   From: <PINPADNAME>
   To: <iOSDEVICE>
   PIN: <BLUETOOTHPIN>

   While validating and exchanging secure credentials, the iOS device might cycle through *Connected* and Not *Connected* statuses.

3.1.2.1.2 Standard Bluetooth Pairing

1. Ensure that:
   - The terminal is powered on
   - The standard Bluetooth device has Bluetooth connectivity enabled

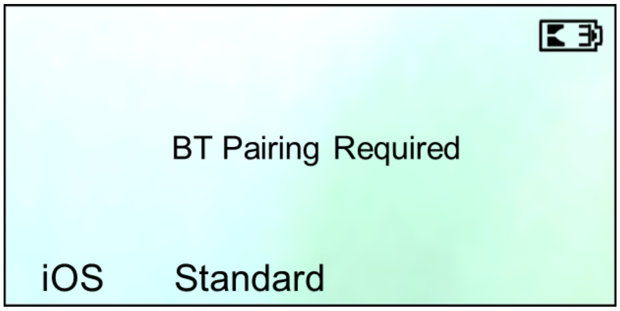
The terminal displays the Bluetooth Pairing Required screen. if the terminal is not at that screen then the terminal must be unpaired (see section Bluetooth Unpairing).



BT Pairing Required

iOS Standard

iSMPc



BT Pairing Required

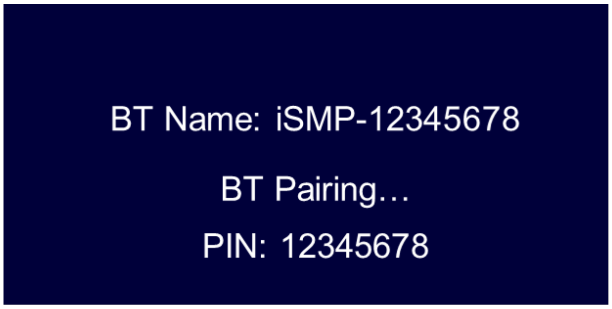
iOS Standard

iCMP

1. To begin the pairing process, select the Standard key (F2) on the terminal.

   a. Some iSMPc terminals are configured to support only one type of Bluetooth pairing. If you have one of these terminals, then the iOS and Standard options pictured to the right are replaced with a single option that reads *Begin*.

2. The terminal goes into discovery mode and displays an eight-digit, randomly generated pairing PIN with the terminal unique Bluetooth name:

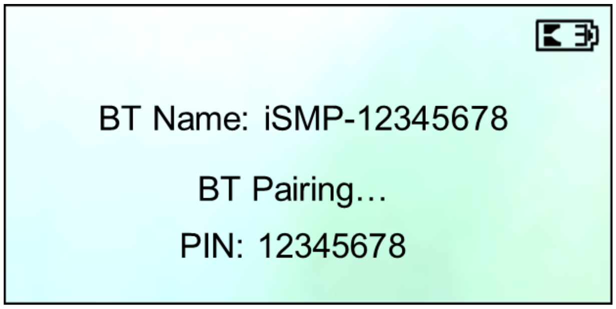


BT Name: iSMP-12345678

BT Pairing…

PIN: 12345678

iSMPc



BT Name: iSMP-12345678

BT Pairing…

PIN: 12345678

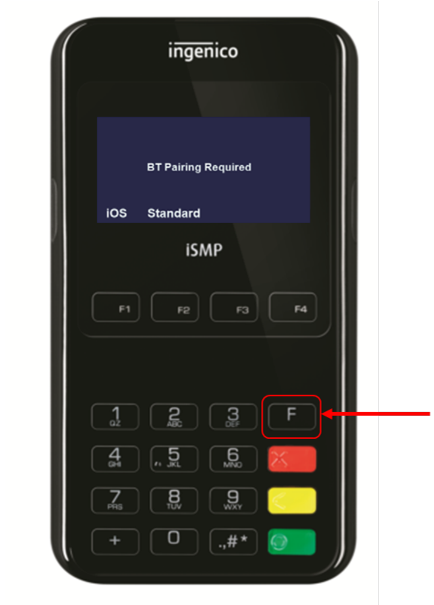
iCMP

3. On the standard Bluetooth device, search for the terminals logical Bluetooth name displayed on the screen of the terminal, and select it to pair.
4. When the standard Bluetooth device prompts for a PIN, enter the PIN that is displayed on the terminal screen.
5. During the pairing process, the terminal displays the following information:

```
 Awaiting remote pairing

To:  <PINPADNAME>

PIN: <BLUETOOTHPIN>
```

3.1.2.1.3 Bluetooth Unpairing

To unpair the terminal from the host or tablet, press the Function key four times in under two seconds:



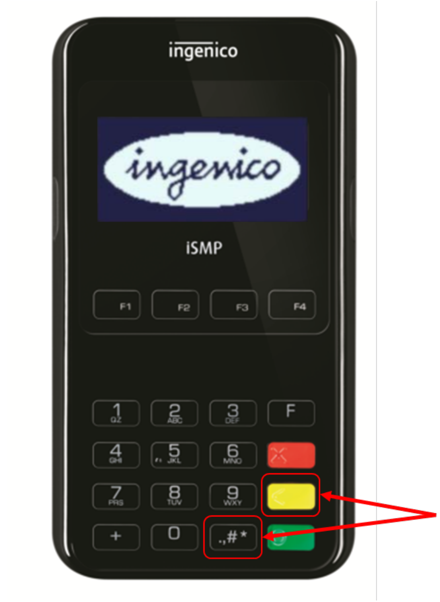
Function Key

The terminal beeps and displays the BT Pairing Required screen.

3.1.2.1.4 Troubleshooting

If the barcode scanner does not power on for the unpairing process:

1. Ensure that the terminal was forgotten on the host device.
2. Turn Bluetooth connectivity off on the host device.
3. Reboot the terminal.



Press yellow key
and "#"/"-" key together
to reboot terminal

4. When terminal reboots, the barcode scanner is enabled to finish the unpairing process.

3.1.2.1.5 Troubleshooting

If the terminal continuously prompts the host device/user to enter a Bluetooth PIN and the unpairing process has been completed:

1. Ensure the terminal has been forgotten on the host device.
2. Turn the host device Bluetooth connectivity off.
3. Reboot the terminal.
4. When the terminal reboots, turn back on the host device Bluetooth connectivity, which stops Bluetooth PIN prompting.

3.1.2.1.6 Pairing Using a QR Code

Using a QR code, a mobile terminal can be paired with a tablet quickly in a retail environment. The terminal can display a QR code of its Bluetooth MAC address and pairing PIN, which the tablet scans, and initiates pairing. The QR code contains:

- The 12-digit hexadecimal MAC address with no separator character
- The eight-digit decimal pairing PIN separated by a single space character

Supported Terminals

The QR code display for Bluetooth pairing is supported on the following terminals:

- iCMP
- iSMPc
- iSMP v4

Configuring the Terminal

To pair a terminal with a standard Bluetooth device, the terminal must:

- Be configured for Bluetooth connectivity
- Have Standard set as the Bluetooth mode in the configuration files
- Include the QR code element to display dynamically a form

Setting Terminal Connectivity to Bluetooth

The terminal must be configured for Bluetooth connectivity. Press *FFFF* to access the communications settings, and select **Bluetooth Settings->Mode = 0->Standard = 2**.

Setting the Bluetooth Type to Standard

Set Bluetooth to Standard in the `config.dfs` file under the heading, *Main Flow*, and file, `mainFlow.dat.`

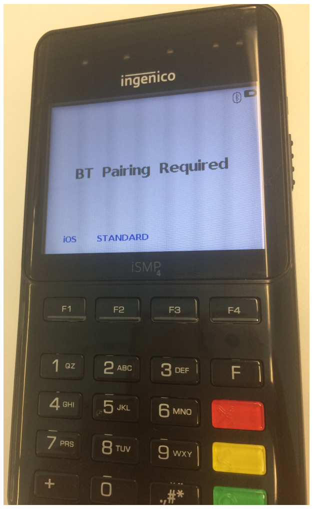
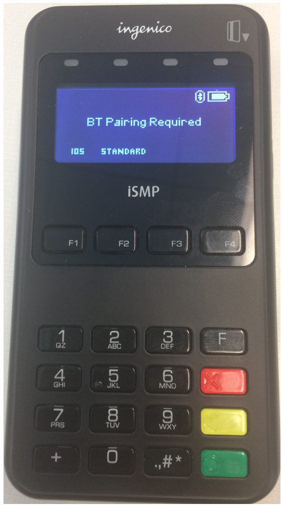
Parameter Name	DFS Data Index	Default Value	Description
Bluetooth Pairing	0007_0033	0	• 0 = Not configured (default) • 1 = iOS • 2 = Standard

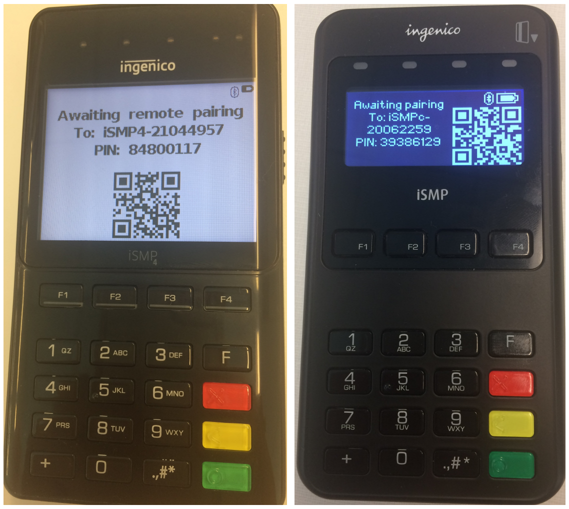
Adding a QR Code Element to a Form

The K3Z form can use “QRImage” to indicate the image is a QR code. The form requires a default image as a place holder for the QR code. Use the VAR_GRA_QR_DATA variable to set QR data dynamically before displaying the form. 70.TQRDATA,<QR code data> generates/updates QR code image with <QR code data>

Pairing a Tablet with a Terminal

Complete the following steps to pair a tablet with a terminal via Bluetooth:

Step	Action
1	Ensure that: • The terminal is powered on • The tablet has Bluetooth connectivity enabled The terminal displays the Bluetooth Pairing Required screen if it has not been paired previously. If it has been paired, unpair it from the original device. **Note:** The screen display is slightly different depending on the terminal screen size.  ingenico BT Pairing Required iOS STANDARD iSMP4 F1 F2 F3 F4  ingenico BT Pairing Required IOS STANDARD iSMP F1 F2 F3 F4 To begin the pairing process, if the Bluetooth mode was: • Set to Standard, select the Begin key (F1) on the terminal (Image not shown) • Not set to Standard (as in these images), select the Standard key (F2) on the terminal The terminal displays the Bluetooth MAC address and PIN as a QR code.

Step	Action
2	Scan the QR code with the tablet.  ingenico Awaiting remote pairing To: iSMP4-21044957 PIN: 84800117 iSMP Awaiting pairing To: iSMPc-20062259 PIN: 39386129 iSMP The tablet decodes the MAC address and PIN and pairs with the terminal.

3.1.2.2 Bluetooth Signal Strength Meter

The application header displays a meter illustrating the Bluetooth signal strength relative to the smart base with which the iWL252 or iWL258 terminal is paired.

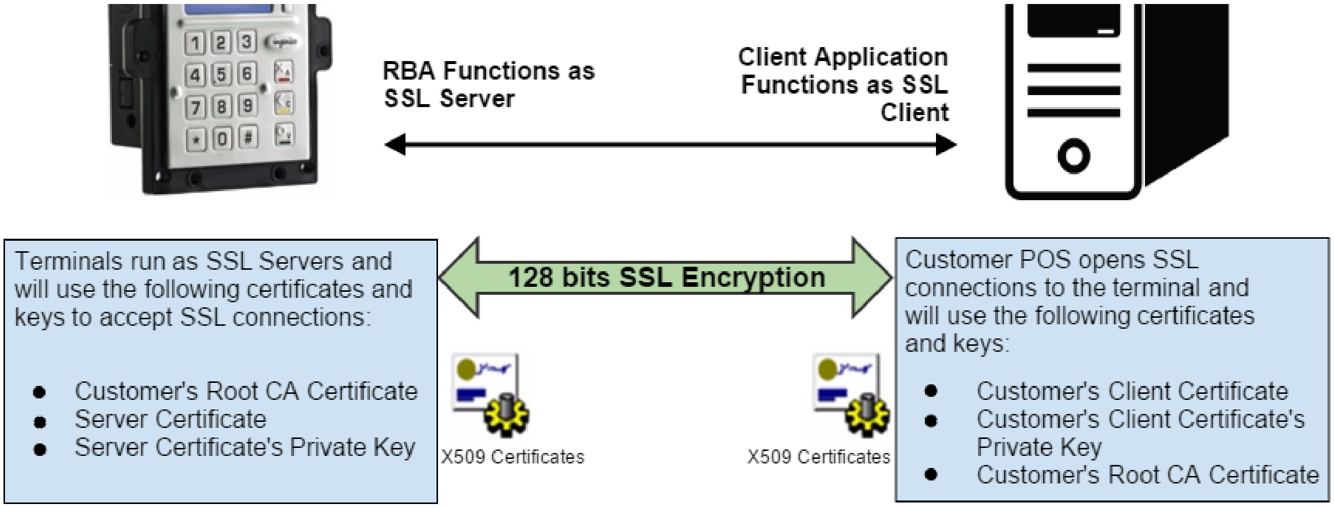
3.1.3 SSL/TLS for Ethernet

Secure Sockets Layer (SSL) handshake and encryption using the client-authenticated Transport Layer Security (TLS) handshake, also known as mutual authentication, can be used with the application.

RBA acts as the SSL server, and the client application acts as the SSL client. Because the authentication is mutual, both parties (server and client) are required to send its certificate to the other. Therefore, during installation, server and client should install the root certificate from the other party. During the handshake, both client and server use the root certificate to validate the certificate presented by the other.

Info

Although the iUP250 terminal is illustrated in the Figure, below, SSL applies to any Telium terminal.


RBA Functions as SSL Server
Client Application Functions as SSL Client
128 bits SSL Encryption
X509 Certificates
X509 Certificates

Terminals run as SSL Servers and will use the following certificates and keys to accept SSL connections:

- Customer's Root CA Certificate
- Server Certificate
- Server Certificate's Private Key

Customer POS opens SSL connections to the terminal and will use the following certificates and keys:

- Customer's Client Certificate
- Customer's Client Certificate's Private Key
- Customer's Root CA Certificate

SSL Connection

For SSL, the Heartbeat keep-alive timer is set to 30 seconds with retries every 15 seconds on failure.

3.1.3.1 *SSL Implementation Requirements*

A set of requirements is outlined in the following table, including certificates, private keys, and configuration.

SSL Implementation Requirements

Requirement	Environment	Purpose	Description
Customer’s Root CA Certificate	Ingenico Terminal	To validate the certificate presented by the client during handshake.	One copy of this certificate is used by all terminals.This certificate should be presented by the customer during the installation and installed on all terminals requiring SSL. Because it is used for validating the client POS is the actual POS. This CA certificate should be sent to Ingenico by the customer and packaged together with the Server Certificate (see below) and Server Certificate Private Key (see below) into a PKCS12 (PFX) container.

Requirement	Environment	Purpose	Description
Server Certificate		To present to the client during the handshake.	This certificate is generated by Ingenico as a Certificate Signing Request (CSR) and is sent to the customer for signing. The resulting CRT (Certificate in PEM format) is packaged by Ingenico into a PKCS12 container.
Server Certificate Private Key		To encrypt the PreMasterSecret during the handshake.	This private key is part of the Server Certificate and should also be stored in the PKCS12 container.
Customer's Root CA Certificate	Client POS	To validate the certificate presented by the Ingenico terminal.	This is the root CA certificate used by the customer when signing the Server Certificate described above. The POS should have this certificate to validate the Server Certificate during the handshake.
Client's Certificate		To present to the server during the handshake.	Each client POS should have a unique copy of this certificate.
Client's Private Key		To encrypt the PreMasterSecret during the handshake.	Each client POS should have a unique private key that matches the Client's certificate.
Set SSL Protocol Version Identifier		To select the SSL protocol version.	TLS version 1.1 or 1.2 must be selected. Refer to security.dat parameter '0091_0034' for setting the TLS version. This setting is checked when a customer has enabled SSL on the terminal and the correct `server.pgz` file has been uploaded.

Info

For a review of the SSL sequence events that occur during a handshake, see also Wikipedia's page on Transport Layer Security.

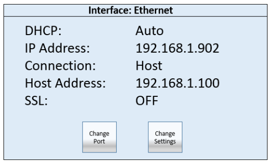
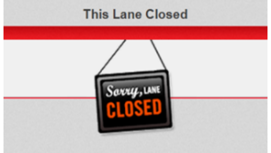
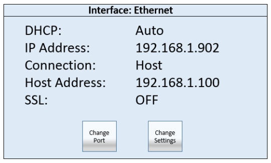
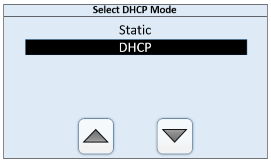
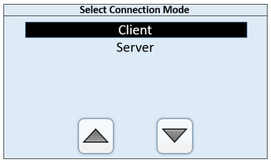
3.1.3.2 Enabling SSL

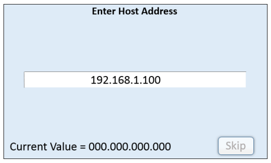
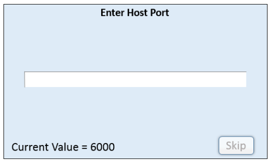
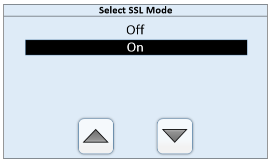
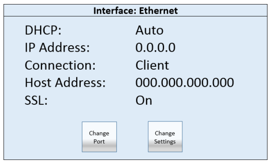
3.1.3.2.1 Requirements

Ingenico provides the following files required for SSL:

- SERVER.PFX in a signed .PGZ file to load to the terminal
- Corresponding CLIENT_CERT.PEM and CLIENT_KEY.PEM files loaded to the POS application directory

3.1.3.2.2 Enabling SSL

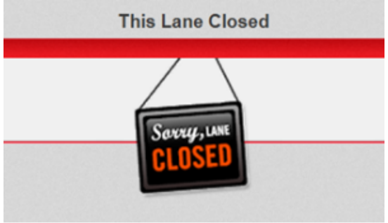
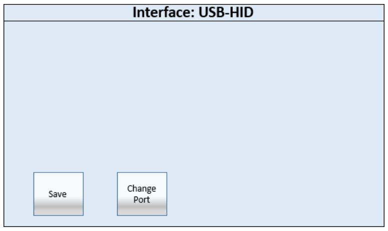
Step	Action	Screen
1	Load the .PGZ file containing SERVER.PFX to the terminal. **Result:** The terminal automatically reboots after disconnecting from LLT. If using 62.x File Write, you must send a 97.x message to reboot the terminal.	 Interface: Ethernet DHCP: Auto IP Address: 192.168.1.902 Connection: Host Host Address: 192.168.1.100 SSL: OFF
2	With the terminal in offline mode, press the F or + key four times. **Result:** The terminal displays the Interface screen.	
3	Select **Change Settings**. **Result:** The terminal displays the Select DHCP Mode menu.	 Interface: Ethernet DHCP: Auto IP Address: 192.168.1.902 Connection: Host Host Address: 192.168.1.100 SSL: OFF
4	Select **DHCP** or **Static** and press Enter. **Result:** The terminal displays the Select Connection Mode menu.	
5	Select **Client** and press Enter. **Result:** The terminal displays the Enter Hostname screen (RBA 21.0.1 or higher) or Enter Host Address screen (older versions).	

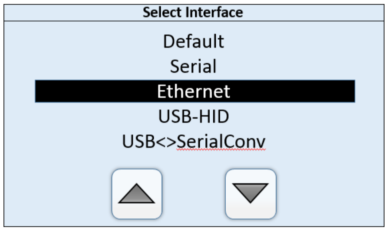
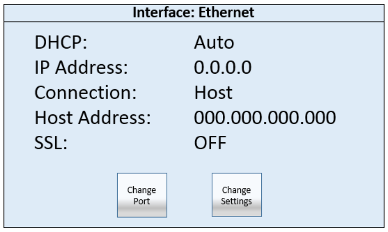
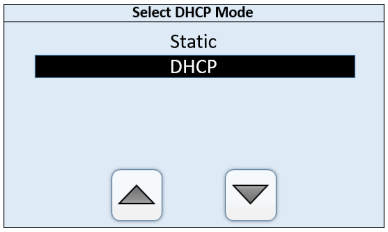
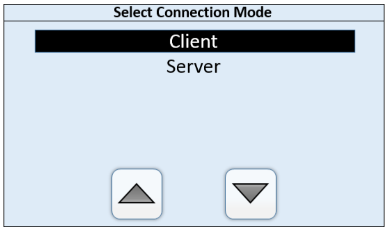
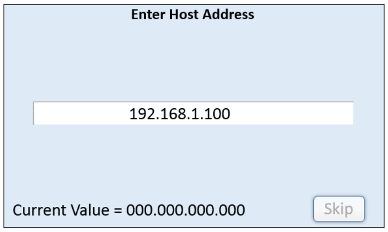
Step	Action	Screen
6	Enter the host IP address (such as 192.168.1.100) and press Enter. If you don't want to enter new data, press **Skip**. **Result:** The terminal displays the Enter Host Port screen.	 Enter Host Address 192.168.1.100 Current Value = 000.000.000.000
7	Enter the Host Port and press Enter. If you don't want to enter new data, press **Skip**. **Result:** The terminal displays the Select SSL Mode screen.	 Enter Host Port Current Value = 6000
8	Select **On** and press Enter. **Result:** The terminal displays the Interface screen.	 Select SSL Mode Off On
9	Press **Save**. **Result:** The terminal saves the communication settings and reboots. The splash screen displays "SSL: ON" to show that SSL is enabled.	 Interface: Ethernet DHCP: Auto IP Address: 0.0.0.0 Connection: Client Host Address: 000.000.000.000 SSL: On

3.1.3.3 Connecting over Ethernet as a Client

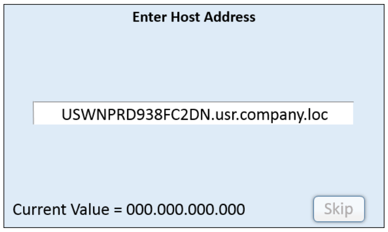
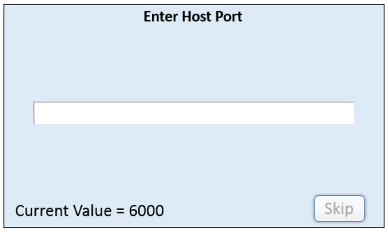
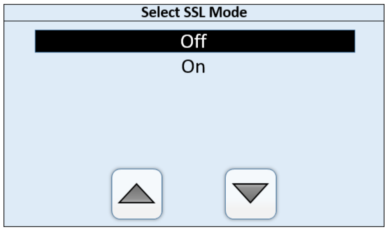
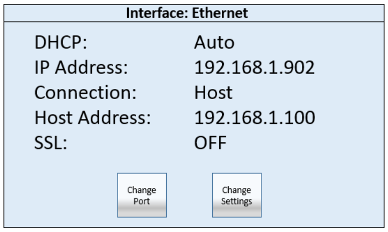
Complete the following steps to connect the terminal as a client over Ethernet to the POS as a host device.

Procedure Using TMS

Step	Action	Screen
1	With the terminal in offline mode, press the F or + key four times. **Result:** The terminal displays the Interface screen.	
2	If you are: • Connected by a method other than Ethernet, select **Change Port** and continue to Step 3. • Connected over Ethernet, skip to Step 4. **Result:** The terminal displays the Select Interface menu.	 Interface: USB-HID

Step	Action	Screen
3	Select **Ethernet**, and press Enter. Result: The terminal displays the Interface screen.	 Select Interface Default Serial Ethernet USB-HID USB<>SerialConv
4	Select **Change Settings**. **Result:** The terminal displays the Select DHCP Mode menu.	 Interface: Ethernet DHCP: Auto IP Address: 0.0.0.0 Connection: Host Host Address: 000.000.000.000 SSL: OFF
5	Select **DHCP** and press Enter. **Result:** The terminal displays the Select Connection Mode menu.	
6	Select **Client**, and press **Enter**. **Result:** The terminal displays the Enter Hostname screen (RBA 21.0.1 or higher) or Enter Host Address screen (previous versions).	
7	Enter either: • The host IP address (such as 192.168.1.100) • DNS name (such as USWNPRD938FC2DN.usr.company.loc) Press **Enter.** If you don't want to enter a new data, press **Skip**. **Result:** The terminal displays the Enter Host Port screen.	**All RBA Versions**  Enter Host Address 192.168.1.100 Current Value = 000.000.000.000

Only RBA 21.0.1 or higher supports DNS name entry.

Step	Action	Screen
		RBA 21.0.1 and Higher  Enter Host Address USWNPRD938FC2DN.usr.company.loc Current Value = 000.000.000.000
8	Enter the Host Port, and press **Enter**. If you don't want to enter a new data, press **Skip**. **Result:** The terminal displays the Select SSL Mode screen.	 Enter Host Port Current Value = 6000
9	Select **Off** or **On,** and press **Enter.** **Result:** The terminal displays the Interface screen. Enabling SSL Mode requires a security certificate be loaded to both the terminal and host.	 Select SSL Mode
10	Press **Save**. **Result:** The terminal saves the communication settings and reboots.	 Interface: Ethernet DHCP: Auto IP Address: 192.168.1.902 Connection: Host Host Address: 192.168.1.100 SSL: OFF

3.1.3.3.1 Procedure Using the TDA.XML

You can also set the URL in the TDA.XML file by entering the host URL address in the the Ethernet -> HOSTIPADDRESS field instead of the IP address. For example:

<Item name="HOSTIPADDRESS" value="USWNPR22420-AE.usr.ingenico.loc" />

3.1.4 Wi-Fi Support

3.1.4.1 Overview

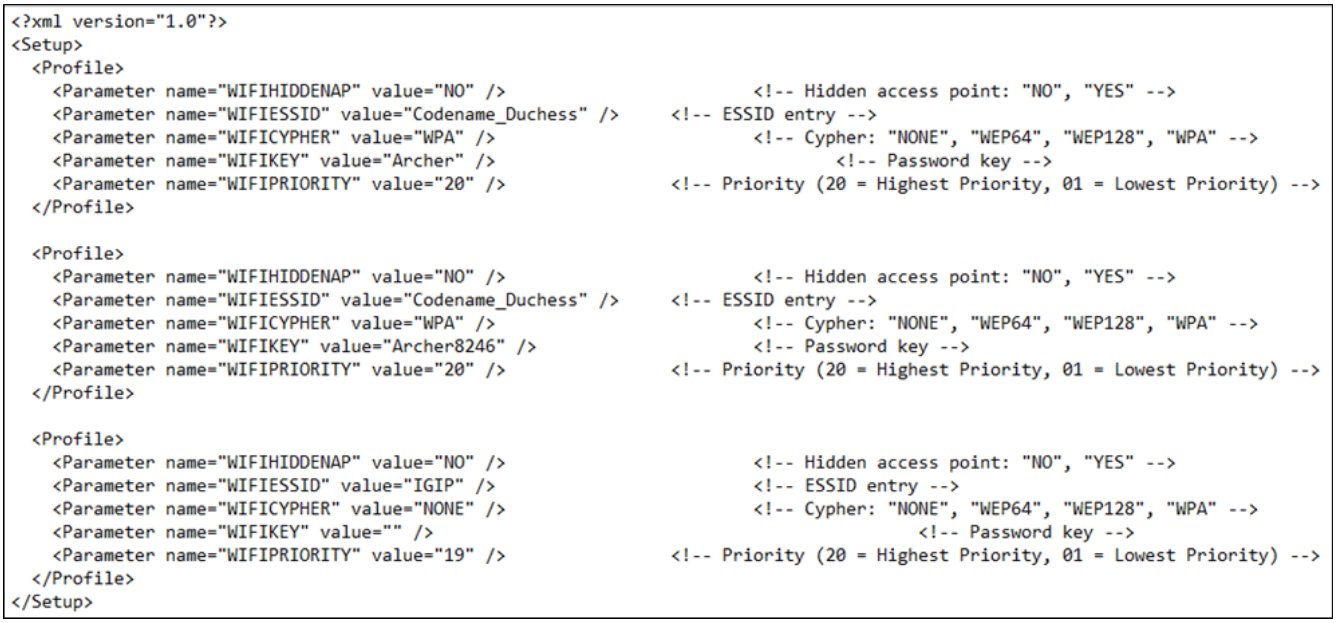
Wi-Fi connections are supported on iWL228, iWL258, and iSMP4 terminals.

The Wi-Fi feature is configurable using the Telium Manager, to automatically scan for, manually select, or search for the SSID connection.

Roaming capabilities are enabled by a chip set in the terminal hardware. An icon in the header displays signal strength.

3.1.4.2 Configuration Files

A terminal is configured for Wi-Fi is via the Telium Manager. A terminal can store as many as 20 Wi-Fi profiles, including IP addresses, passwords, and MAC addresses. Wi-Fi profiles are created and maintained by the network administrator/authority. They are ranked in order of priority, so the terminal attempts to connect with the highest priority Wi-Fi profile first. To load new Wi-Fi profiles automatically, the terminal must be powered up. Then, the `WiFiPROF.XML` file must be downloaded to the /HOST folder. Because this file contains sensitive information, it is deleted when the configuration is complete. Refer to the following `WiFiPROF.XML` file excerpt.



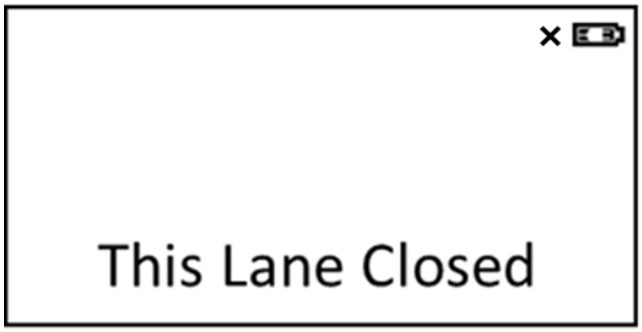
```
<?xml version="1.0"?>
<Setup>
  <Profile>
    <Parameter name="WIFIHIDDENAP" value="NO" />                        <!-- Hidden access point: "NO", "YES" -->
    <Parameter name="WIFIESSID" value="Codename_Duchess" />     <!-- ESSID entry -->
    <Parameter name="WIFICYPHER" value="WPA" />                         <!-- Cypher: "NONE", "WEP64", "WEP128", "WPA" -->
    <Parameter name="WIFIKEY" value="Archer" />                                 <!-- Password key -->
    <Parameter name="WIFIPRIORITY" value="20" />                <!-- Priority (20 = Highest Priority, 01 = Lowest Priority) -->
  </Profile>

  <Profile>
    <Parameter name="WIFIHIDDENAP" value="NO" />                        <!-- Hidden access point: "NO", "YES" -->
    <Parameter name="WIFIESSID" value="Codename_Duchess" />     <!-- ESSID entry -->
    <Parameter name="WIFICYPHER" value="WPA" />                         <!-- Cypher: "NONE", "WEP64", "WEP128", "WPA" -->
    <Parameter name="WIFIKEY" value="Archer8246" />                     <!-- Password key -->
    <Parameter name="WIFIPRIORITY" value="20" />                <!-- Priority (20 = Highest Priority, 01 = Lowest Priority) -->
  </Profile>

  <Profile>
    <Parameter name="WIFIHIDDENAP" value="NO" />                        <!-- Hidden access point: "NO", "YES" -->
    <Parameter name="WIFIESSID" value="IGIP" />                         <!-- ESSID entry -->
    <Parameter name="WIFICYPHER" value="NONE" />                        <!-- Cypher: "NONE", "WEP64", "WEP128", "WPA" -->
    <Parameter name="WIFIKEY" value="" />                                               <!-- Password key -->
    <Parameter name="WIFIPRIORITY" value="19" />                <!-- Priority (20 = Highest Priority, 01 = Lowest Priority) -->
  </Profile>
</Setup>
```

Wi-FiPROF.XML File Excerpt with Wi-Fi Settings

3.1.4.3 Wi-Fi Signal Strength Icon

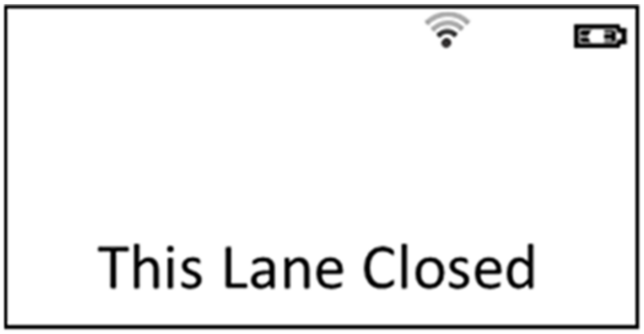
An icon indicating Wi-Fi signal strength is displayed in the upper-right corner of the screen. Variations of this icon represent no signal, 50% signal, 75% signal, and 100% (full) signal. The Wi-Fi chip set in the terminal permits seamless roaming; the terminal can roam between access points (APs) configured for the same SSID without user intervention. Refer to the following illustrations for how this icon is used to indicate Wi-Fi signal strength.

3.1.4.4 *iWL228 Wi-Fi Icons*



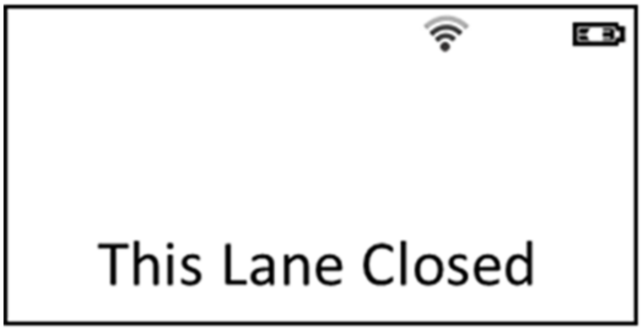
This Lane Closed

WiFi No Signal



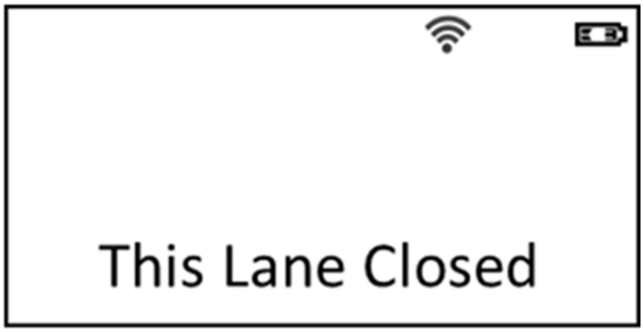
This Lane Closed

WiFi 50% Signal



This Lane Closed

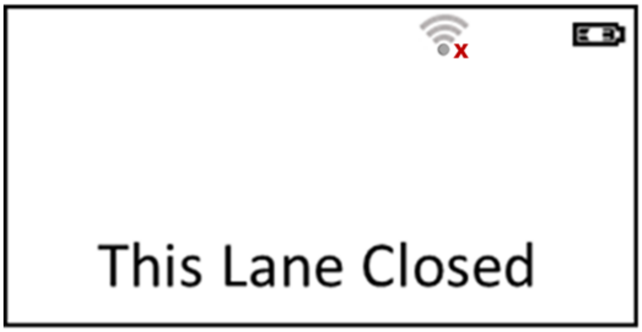
WiFi 75% Signal



This Lane Closed

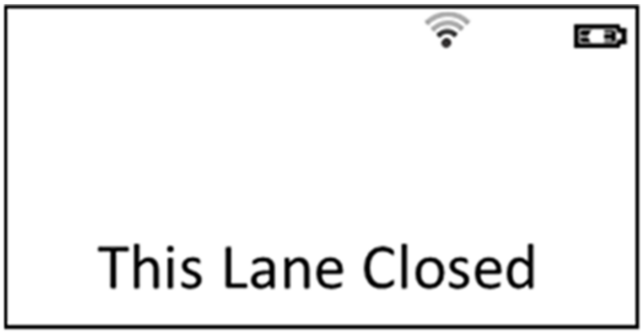
WiFi 100% Signal

3.1.4.5 *iWL258 Wi-Fi Icons*



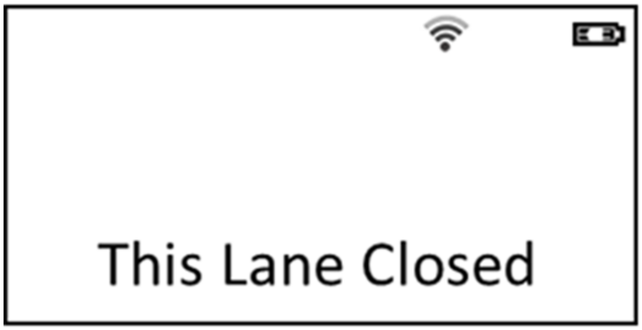
This Lane Closed

WiFi No Signal



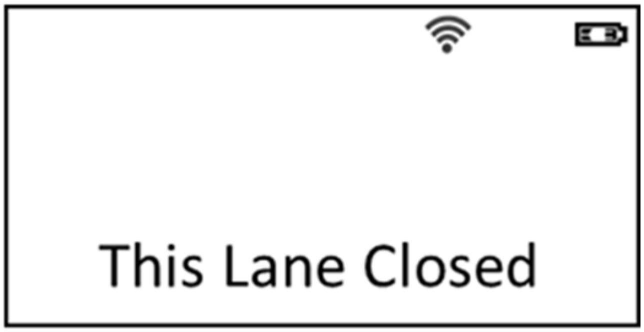
This Lane Closed

WiFi 50% Signal



This Lane Closed

WiFi 75% Signal



This Lane Closed

WiFi 100% Signal

3.1.4.6 *Implementing a Secure Connection*

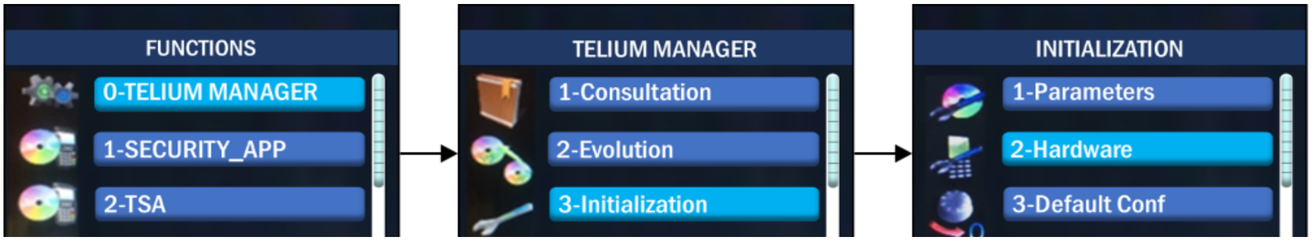
To ensure a secure Wi-Fi connection, P2PE encryption of cardholder data must be enabled.

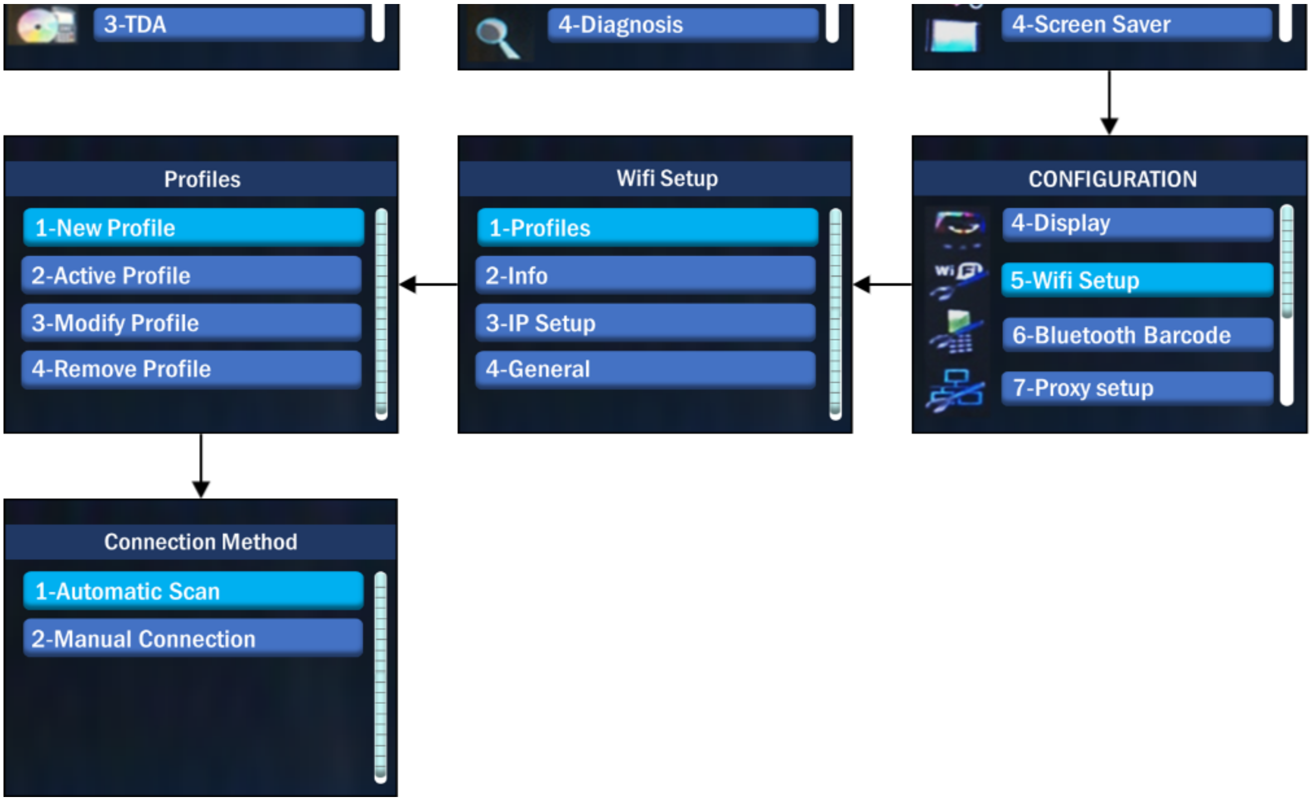
Additionally, SSL Protocol Version 1.1 or 1.2 is required. Refer to security.dat parameter 0091_0034 for configuring the SSL Protocol Version. Note that with SSL/TLS enabled, a terminal must be provisioned with the necessary certificates before being deployed to a location, making it more difficult for unauthorized use elsewhere.

The terminal must also be identifiable by the POS via Wi-Fi. The application responds to UDP broadcasts with the terminal's serial number, IP address, and MAC address, which enables the POS to discover and identify any iWL terminals on the subnetwork without an active connection.

3.1.4.7 *Configuring for Wi-Fi Using the TDA*

Wi-Fi can be configured using the Telium Manager. The following illustration shows the menu selection flow for configuring Wi-Fi.





Using the Telium Manager to Configure Wi-Fi

By selecting the **Automatic Scan** option in the **Connection Method** menu, the terminal searches for all available networks. If multiple access points are available with the same SSID, the terminal displays a list of access points as if each one were its own SSID.

You can also manually enter the SSID of the network, keeping in mind that the terminal does not support alphanumeric entry. When a selection is made, the user is prompted to select the security settings and enter the password for the network (if applicable).

To add an access point that is:

- Broadcasting its SSID, press the **Search** button
- Hidden, press the **Add** button, and enter the SSID

After the connection is established, the Wi-Fi icon is displayed, indicating the signal strength.

> The terminal defaults to DHCP communication. Setting the profile and changing the COM to Ethernet with DHCP enabled allows the POS to connect to the terminal and send and receive messages.

> The Power Save mode must be disabled when running Wi-Fi.

3.1.4.8 *Configuring Wi-Fi Roaming*

Wi-Fi roaming settings are not included in TDA.XML by default, but they can be added if the default settings do not work in a specific location.

When initializing a Wi-Fi communications port, the terminal compares these roaming settings to Telium Manager's current values. The Manager is updates its own parameters to match the values from TDA.XML. The following example shows the optional roaming configuration settings that can be added to TDA.XML:

```
<Wifi>
<! Settings below are Optional -->
<Item name="WIFIROAMING" value="ON" />
<Item name="WIFISCANINTERVAL" value="10000" />
<Item name="WIFI24G" value="ON" />
<Item name="WIFI5G" value="ON" />
<Item name="WIFILOWRSSI" value="-77" />
<Item name="WIFILOWPASSFILTER" value="10" />
<Item name="WIFIMINRSSI" value="-72" />
</Wifi>
```

The following table describes the optional Wi-Fi roaming settings:

TDA.XML Optional Parameters

TDA.XML Parameter	Recommended Value	Description
WIFIROAMING	either ON or OFF	Enables or disables Wi-Fi roaming.
WIFISCANINTERVAL	10000	Interval between each background access point scan in milliseconds. Aceepts values 0 to 60000.
WIFI24G	OFF if roaming is not used	Enables or or disables the background scan for networks broadcasting on the 2.4GHz band.
WIFI5G	OFF if roaming is not used	Enables or or disables the background scan for networks broadcasting on the 5GHz band.
WIFILOWRSSI	-77	RSSI (signal stengh in dBm) threshold to allow roaming to a target (new) access point. If the current connection strength drops below this threshold, the terminal attempts to connect to the available access point with the highest signal strength. Accepts values -1 to -100.
WIFILOWPASSFILTER	10	Minimum seconds between roaming. This minimizes roaming in unstable environments. Accepts values 0 to 60.

TDA.XML Parameter	Recommended Value	Description
WIFIMINRSSI	-72	Minimum RSSI (signal stengh in dBm) value of the target (new) access point to allow roaming to the new access point if RSSI of the current access point drops below the `WIFILOWRSSI` value.

3.1.4.8.1 Other Wi-Fi and Ethernet Settings in TDA.XML

For each Wi-Fi and Ethernet setting in TDA.XML not included in the previous table, TDA checks the parameter's `TMUPDATED` flag. If the flag is set to 0:If it is set to 0:

- TDA copies Ethernet and Wi-Fi settings to Telium Manager.
- After updating Manager, TDA sets `TMUPDATED` to 1 and reboot.
- While TDA updates the Manager, the terminal displays "Updating Telium Manager Settings".

> Changes made in Manager are not copied to TDA.XML. Therefore, all communications changes should be made using RBA or TDA, not Telium Manager.

3.1.5 Quick Access to Communication Settings on iWL Terminals

3.1.5.1 Accessing the Communication Settings Screen on iWL Terminals

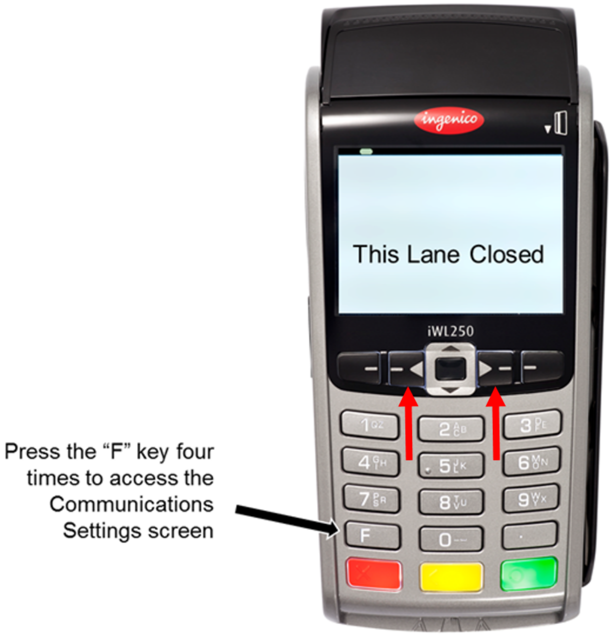
Ingenico iWL-series terminals allow you to change Bluetooth or Wi-Fi settings from the Communication Settings screen. This screen is accessed by pressing the **F** key four times. You can select Bluetooth or Wi-Fi, depending on the terminal model as follows:

- Bluetooth-capable terminals
  - iWL222
  - iWL250
- Wi-Fi-capable terminals
  - iWL228
  - iWL258

Refer to the Communications Supported per Terminal Model section for more information on the communication types supported per terminal model.

> Terminal cases are generically labeled as follows:
>
> - The case for the iWL222 and iWL228 is marked as **iWL220**.
> - The case for the iWL250 and iWL258 is marked as *iWL250*.

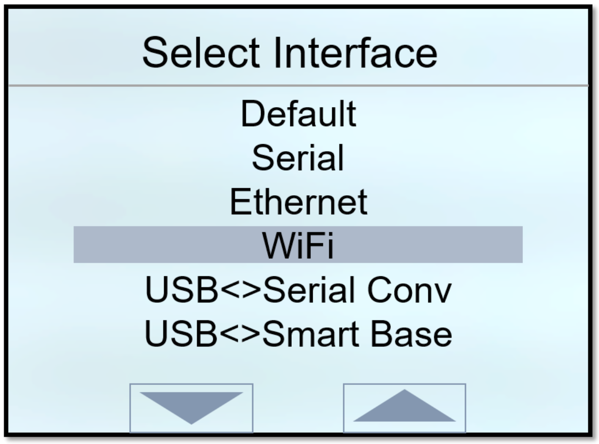
To access the Communication Settings screen, the terminal must be booted to the This Lane Closed screen. Then press the **F** key four times.



Press the “F” key four times to access the Communications Settings screen

Accessing the iWL Communications Screen on an iWL Terminal

After pressing the **F** key four times from the This Lane Closed screen, the Communication Settings menu is displayed as follows:



iWL258 Communications Screen

> This screen is for an iWL258 terminal. Note that the Wi-Fi option is listed, but the Bluetooth option is not. Use the inner Function keys (see red arrows on iWL terminal shown above) to navigate the scroll buttons displayed on the screen.

Select the appropriate link to view connection instructions and menu options:

3.1.5.2 iWL250 Shortcut to Bluetooth Setting

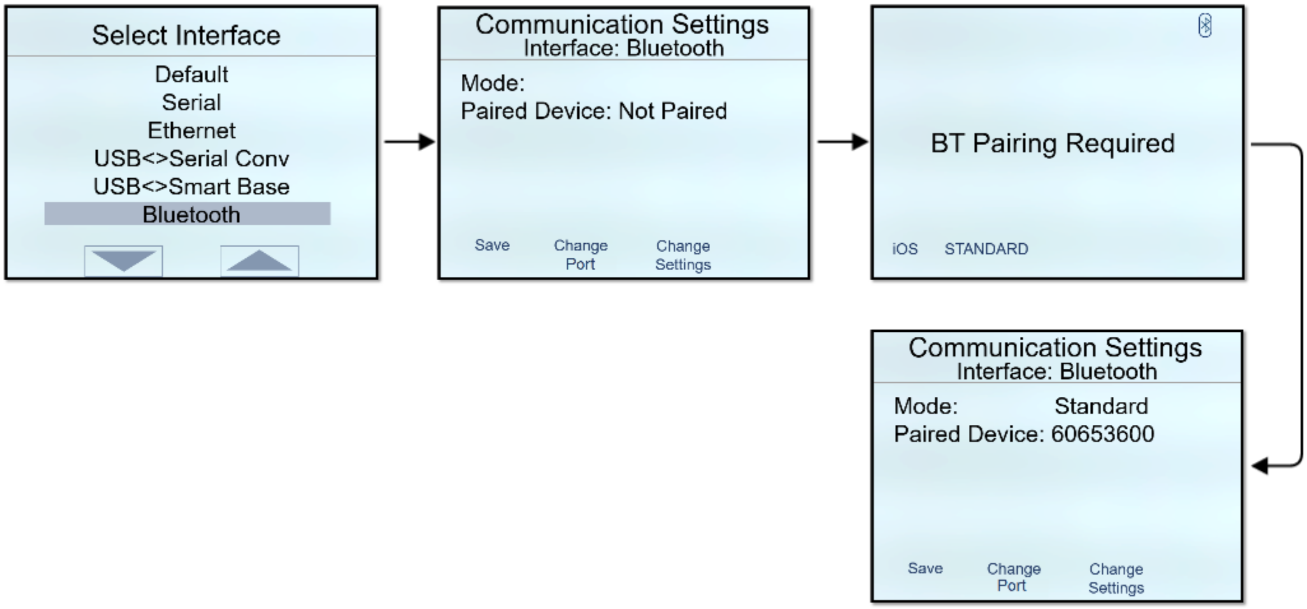
3.1.5.2.1 **Navigating the Bluetooth Settings Menu**

> **Bluetooth Pairing**
> NOTE: The iWL250 cannot be paired until it is associated with a Bluetooth cradle.
> **If the iWL250 terminal is unpaired from the cradle, the terminal must be re-associated with the cradle before pairing again.**

Pairing iWL250 with Bluetooth Cradle

Follow the steps outlined below to pair the iWL250 with the Bluetooth cradle:

1. Scroll to select **Bluetooth** from the menu options.
2. Select **Change Settings** from the Communication Settings screen.
3. Select **IOS** or **Standard** at the BT Pairing Required screen to pair the cradle with the terminal.
4. Select **Change** Port to return to the main menu.



Mode:
Paired Device: Not Paired

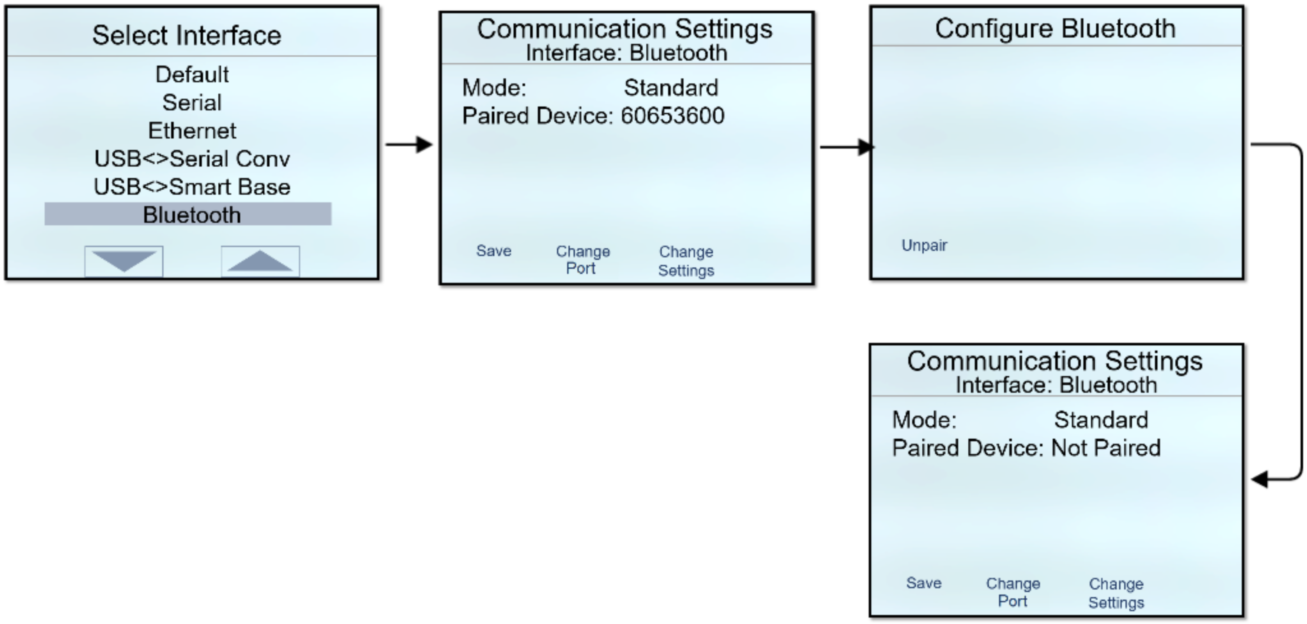
Mode: Standard
Paired Device: 60653600

iWL250 Pairing Terminal with Cradle Menus

Unpairing iWL250 with Bluetooth Cradle

Follow the steps outlined below to unpair the iWL250 from the Bluetooth cradle:

1. Scroll to select **Bluetooth** from the menu options.
2. Select **Change Settings** from the Communication Settings screen.
3. Select **Unpair** from the menu
4. Select **Change Interface** to return to the main menu.



Mode: Standard
Paired Device: 60653600

Mode: Standard
Paired Device: Not Paired

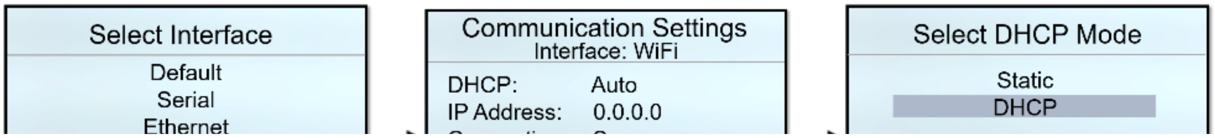
iWL250 Unpairing Terminal with Cradle Menus

3.1.5.3 iWL258 Shortcut to Wi-Fi Settings

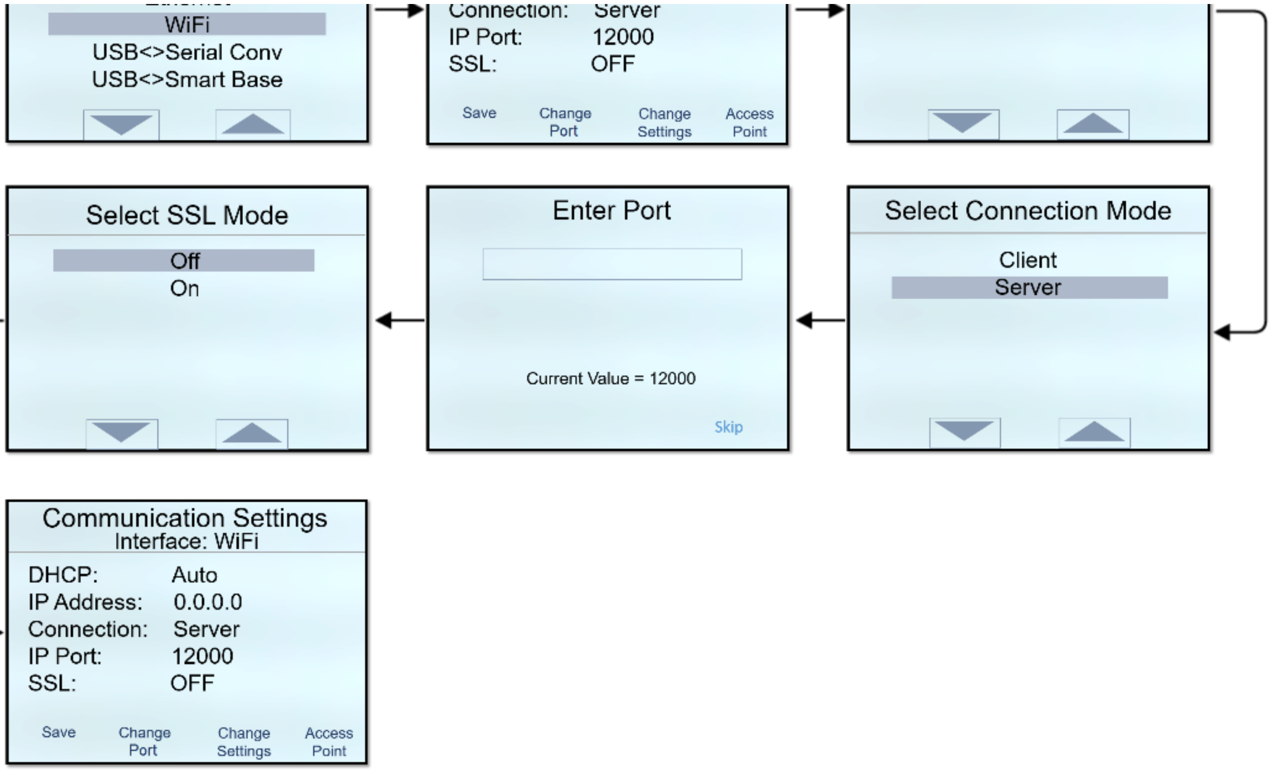
3.1.5.3.1 **Changing Wi-Fi Connection Settings**

Complete the following steps to change Wi-Fi connection settings for the iWL258:

1. Scroll to select **WiFi** from the menu options.
2. Select **Change Settings** from the Communication Settings screen.
3. Set DHCP mode to *Static* or *DHCP*.
4. Set the Connection Mode to *Client* or *Server*.
5. Enter a Port Value if different from the current value.
6. Set the SSL Mode to *Off* or *On*.
7. Select **Change Port** to return to the main menu.



DHCP: Auto
IP Address: 0.0.0.0



Connection: Server
IP Port: 12000
SSL: OFF

Current Value = 12000

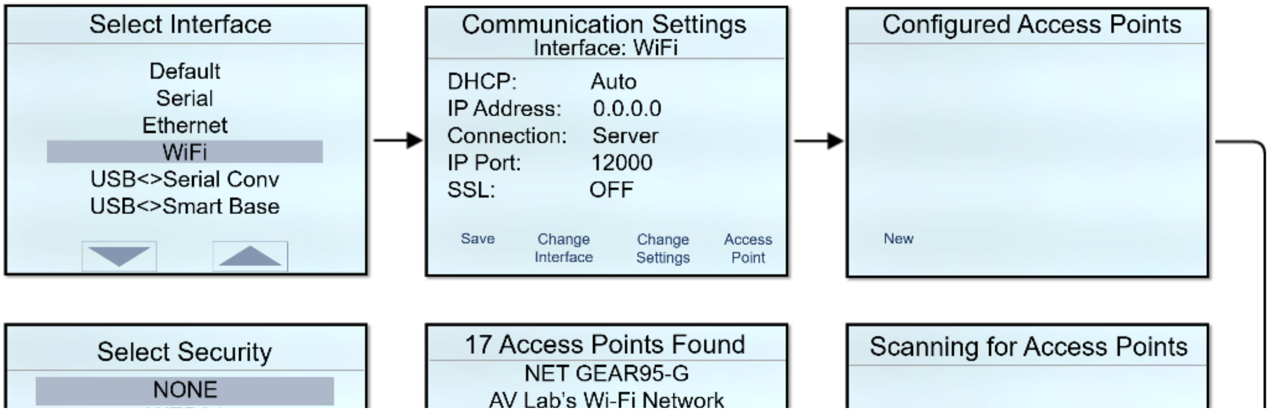
DHCP: Auto
IP Address: 0.0.0.0
Connection: Server
IP Port: 12000
SSL: OFF

iWL258 Wi-Fi Change Settings Menu

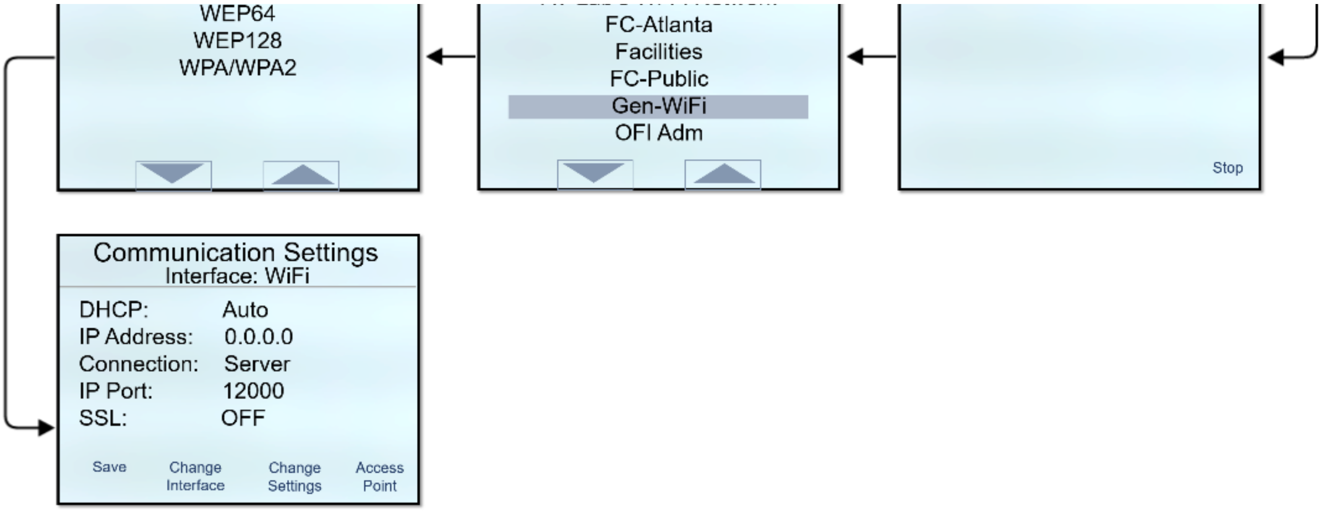
3.1.5.3.2 Navigating the Wi-Fi Access Points Menu

Complete the following steps to select the iWL258 Wi-Fi access point:

1. Scroll to select **WiFi** from the menu options.
2. Select **Access Point** from the Communication Settings screen.
3. Select **New** to scan for new access points.
4. Select your access point from the options listed.
5. Select the security settings for your network, or select *NONE*.
6. Select **Change Interface** to return to the main menu.



DHCP: Auto
IP Address: 0.0.0.0
Connection: Server
IP Port: 12000
SSL: OFF



Communication Settings
Interface: WiFi

DHCP: Auto
IP Address: 0.0.0.0
Connection: Server
IP Port: 12000
SSL: OFF

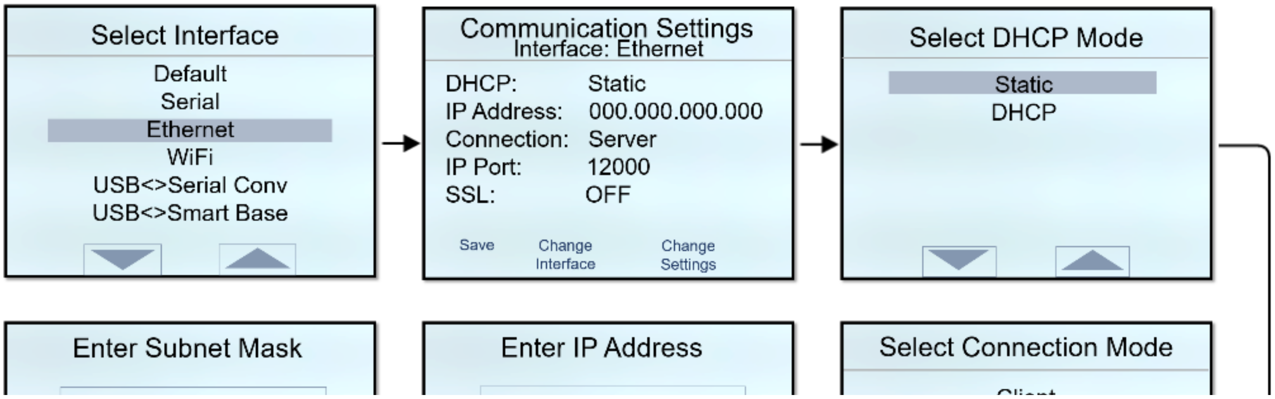
iWL258 Wi-Fi Access Points Menu

3.1.5.4 iWL Shortcut to Ethernet Settings

3.1.5.4.1 Navigating the Ethernet Settings Menu

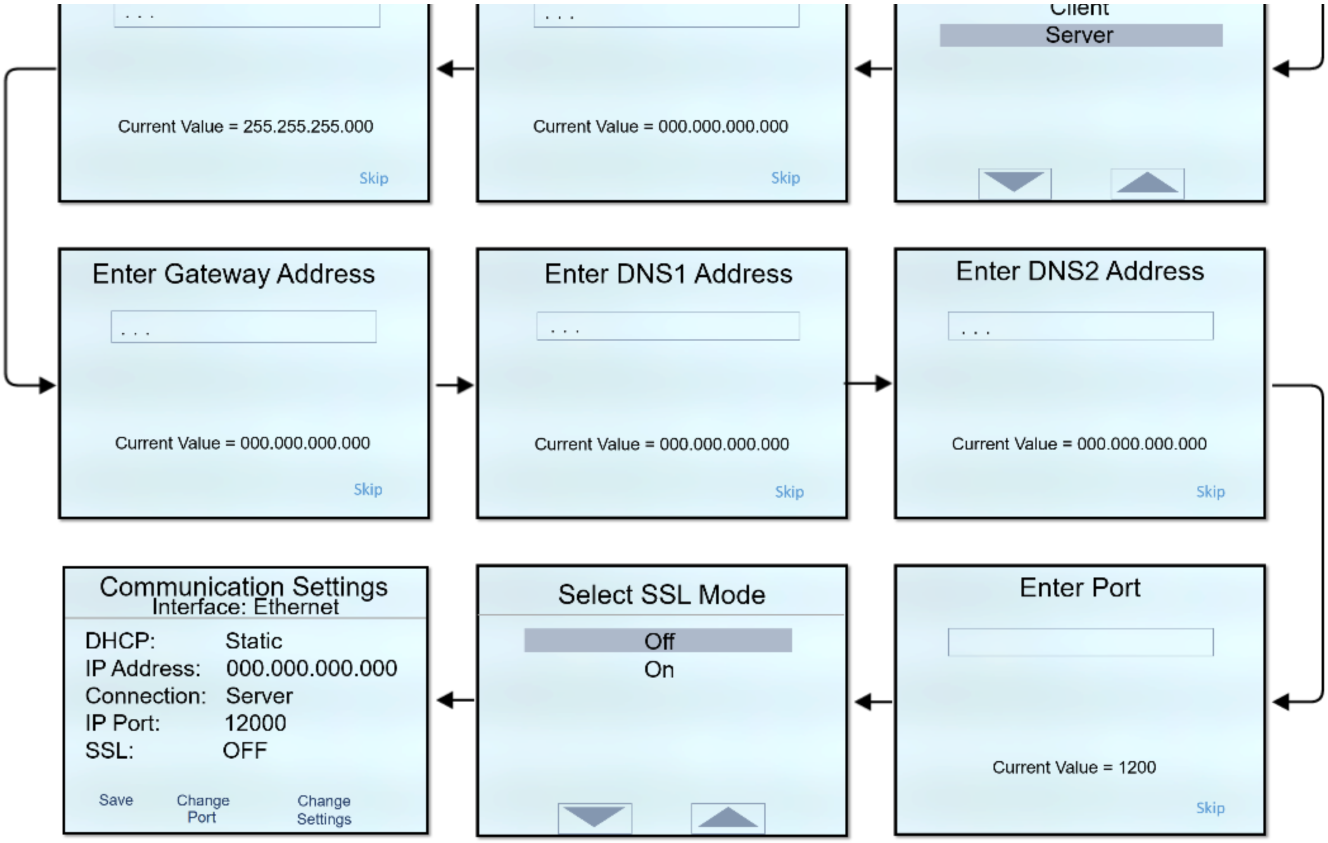
Complete the following steps to change the iWL series Ethernet settings:

1. Scroll to select **Ethernet** from the menu options.
2. Select "Change Settings" from the Communication Settings screen.
3. Set DHCP mode to "Static" or "DHCP".
4. Enter Subnet Mask Value if different from the current value. Skip this step to use the default value.
5. Enter IP Address if different from the current value. Skip this step to use the default value.
6. Set the Connection Mode to "Client" or Server".
7. Enter the Gateway Address. Skip this step to use the default value.
8. Enter the DNS1 Address. Skip this step to use the default value.
9. Enter the DNS2 Address. Skip this step to use the default value.
10. Enter Port Value if different from the current value. Skip this step to use the default value.
11. Set the SSL Mode to "Off" or "On".
12. Select "Change Port" to return to the main menu.



Communication Settings
Interface: Ethernet

DHCP: Static
IP Address: 000.000.000.000
Connection: Server
IP Port: 12000
SSL: OFF



Current Value = 255.255.255.000

Current Value = 000.000.000.000

Client
Server

Enter Gateway Address

Current Value = 000.000.000.000

Enter DNS1 Address

Current Value = 000.000.000.000

Enter DNS2 Address

Current Value = 000.000.000.000

Communication Settings
Interface: Ethernet

DHCP: Static
IP Address: 000.000.000.000
Connection: Server
IP Port: 12000
SSL: OFF

Select SSL Mode

Off
On

Enter Port

Current Value = 1200

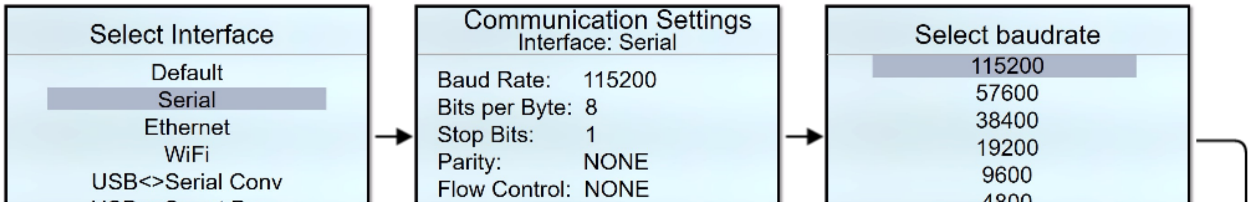
iWL250 Ethernet Settings

3.1.5.5 iWL Shortcut to Serial Settings

3.1.5.5.1 **Navigating the Serial Settings Menu**

Follow the steps outlined below to change the iWL series Serial Settings:

1. Scroll to select "Serial" from the menu options.
2. Select "Change Settings"from the Communication Settings screen.
3. Set baudrate to the desired settting.
4. Set data bits to "7" or "8".
5. Set stop bits to "1" or "2".
6. Set parity to "NONE" "ODD" or "EVEN".
7. Set flow control to "NONE" or "HARD".
8. Select "Change Port" to return to the main menu.



Select Interface

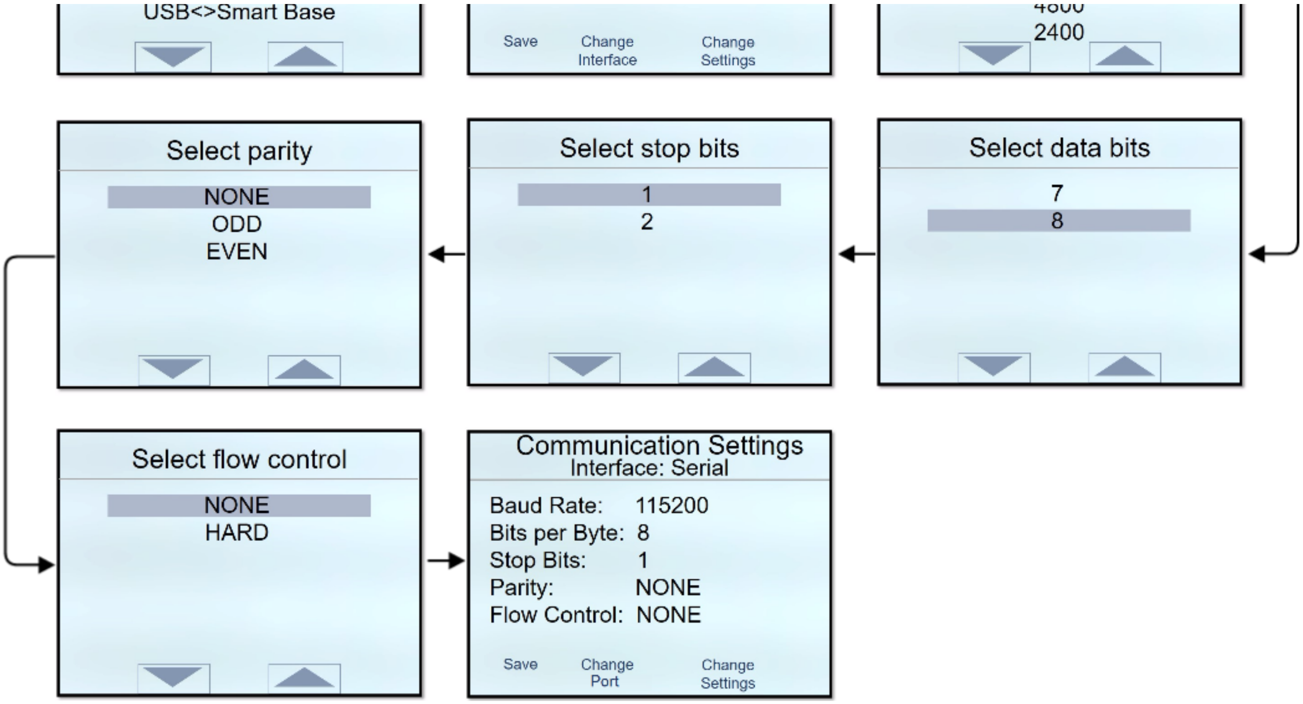
Default
Serial
Ethernet
WiFi
USB<>Serial Conv

Communication Settings
Interface: Serial

Baud Rate: 115200
Bits per Byte: 8
Stop Bits: 1
Parity: NONE
Flow Control: NONE

Select baudrate

115200
57600
38400
19200
9600



Baud Rate: 115200
Bits per Byte: 8
Stop Bits: 1
Parity: NONE
Flow Control: NONE

iWL series Serial Settings Menu

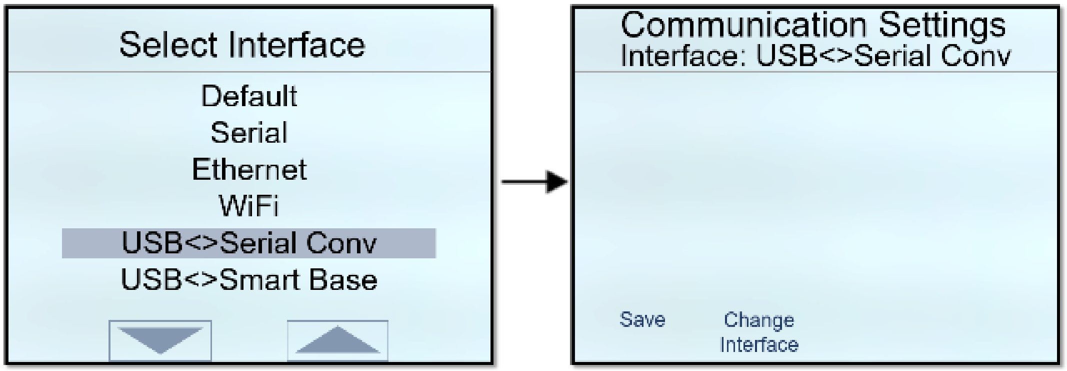
3.1.5.6 iWL Shortcut to USB<>Serial Conv Settings

3.1.5.6.1 Navigating the USB<>Serial Conv Settings Menu

Follow the steps outlined below to change the terminal connection to USB to Serial Converter:

1. Scroll to select "USB<>Serial Conv" from the menu options.
2. Select "Save" to switch connection to USB to Serial Converter.

> The terminal will automatically reboot after saving this option.



iWL USB<>Serial Conv Setting Menus

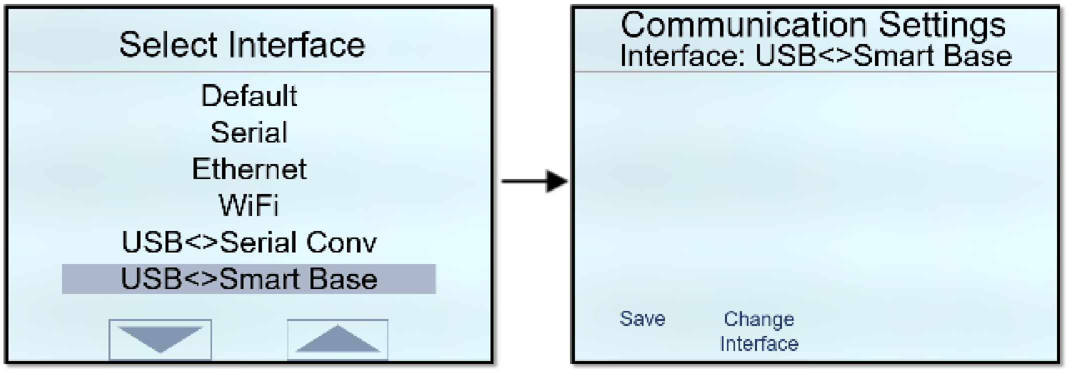
3.1.5.7 iWL Shortcut to USB<>Smart Base Settings

3.1.5.7.1 **Navigating the USB<>Smart Base Settings Menu**

Follow the steps outlined below to change the terminal connection to USB to Smart Base:

1. Scroll to select "USB<>Smart Base" from the menu options.
2. Select "Save" to switch connection to USB to Smart Base.

> The terminal will automatically reboot after saving this option.



Communication Settings
Interface: USB<>Smart Base

iWL USB<>Serial Conv Setting Menus

3.1.6 Setting the Charge Current of an iPad in Serial Mode with iSMP V4

When using an iSMP4 terminal with an iPad and Wi-Case in serial mode, the iPad requires communication with the iSMP4 to set the iPad charge rate.

In the `config.dfs` section of the `mainFlow.dat` file, set 0007_0056 to enable setting the iPad charge rate via the terminal.

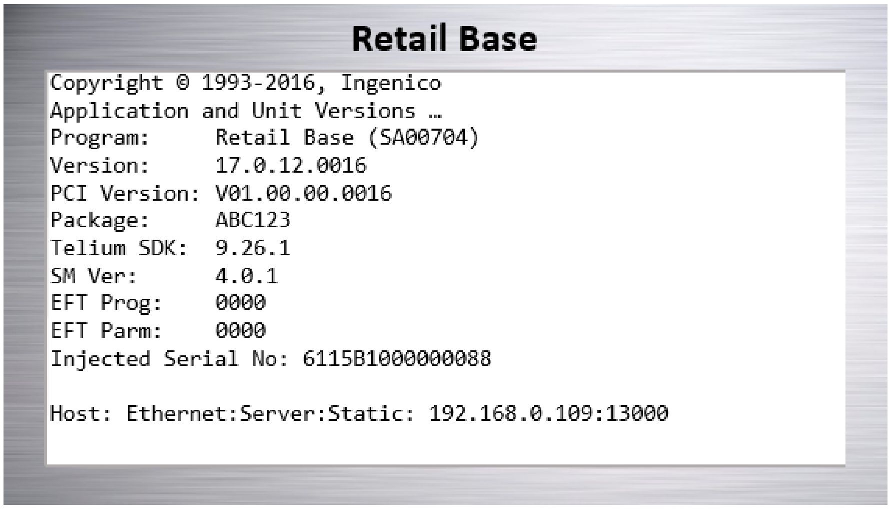
Using a 28.x Set Variable message, use variable 833 to set the iPad charge rate. If enabled, the default charge rate is 1000ma.

3.2 RBA Splash Screen

The following information is optionally displayed on the RBA Splash screen, as well as other terminal and configuration information:

- PCI version
- Package number, called from PACKAGE.TXT in the terminal HOST directory

This sample splash screen shows both the PCI version and package number:



Retail Base

```
Copyright © 1993-2016, Ingenico
Application and Unit Versions …
Program:     Retail Base (SA00704)
Version:     17.0.12.0016
PCI Version: V01.00.00.0016
Package:     ABC123
Telium SDK:  9.26.1
SM Ver:      4.0.1
EFT Prog:    0000
EFT Parm:    0000
Injected Serial No: 6115B1000000088

Host: Ethernet:Server:Static: 192.168.0.109:13000
```

1 RBA Splash Screen

3.3 Text-Entry for Non-iSC Terminals

3.3.1 Overview

To enter the SSID and password for Wi-Fi terminals, text-entry capability is available for the iWL250-series terminals. The implementation is similar to text-entry on a cell phone. Pressing a numeric key multiple times produces successive alphanumeric characters. The iWL250 allows alphanumeric entries (uppercase and lowercase) using the F and special-character keys.

3.3.2 Implementation

The following table describes the characters that are entered with successive key strokes.

Characters by Key

Key	Characters with Successive Key Strokes
0	0
1	1 q z Q Z
2	2 a b c A B C
3	3 d e f D E F
4	4 g h i G H I
5	5 j k l J K L
6	6 m n o M N O
7	7 p r s P R S

Key	Characters with Successive Key Strokes
8	8 t u v T U V
9	9 w x y W X Y
Symbol	! ? , ; : \ / ~ ` # @ . ^ - [ ] { } ( ) < > = *

3.3.3 Procedure

1. Start the terminal and wait for the This Lane Closed form to be displayed.
2. Press the F key four times to display the communication screen.
3. Click Change Port.
4. Select Wi-Fi communication type, and press Enter.
5. Select Access Point.
6. Select New, and wait while the terminal scans for available SSIDs.
7. Select an SSID from the list.
8. Select security, such as WPA/WPA2.
9. As outlined in the previous table, use the keyboard to enter the password.
10. Confirm that the SSID is selected.
11. Press the green Enter button.
12. Select Save and exit.

3.4 Associating iSMP and iSMPc Terminals with Multi-Charging Base

Overview

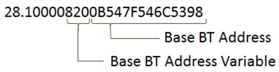
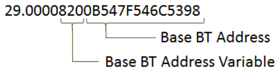
An enhancement to the RBA enables iSMP and iSMPc association with a multi-charging base using the 28.x Set Variable Request message. This enhancement enables the POS to set the base association without having to manually step through the Telium Manager. To associate the base, the POS sends the base Bluetooth address to the terminal via a 28.x message. Variable 820 holds a string of 14 hex-ASCII characters which make up the address of the base, also referred to as cradle. When the 28.x message is received, the previous base association is deleted and replaced with the new base address. The BT address must be 14 characters in length. If more than 14 characters or less than 14 characters are received, the terminal will display an "OUT OF RANGE" message. With the base address received in the correct format, the terminal will display "CONNECTION" once the online message is sent from the POS.

To retrieve the address of the base associated with the terminal, the POS sends a 29.x Get Variable Request message with variable 820 as the variable to be read. The terminal then reads the base address stored in this variable (if present) and returns it to the POS using the 29.x Get Variable Response message. As a note, the first two characters of this variable are always '00' because the Telium Manager does not save the first two bytes in its settings.

3.4.1 Usage Example

In the following example, a base address of '0B547F546C5398' is sent to the terminal using the 28.x message. The terminal receives and stores the base address in variable 820, overwriting the previous address if present. The POS then sends a 29.x request message to confirm the new base association. The terminal returns a 29.x response message with a value of '0B547F546C5398' for variable 820, confirming the new base association. With that, the POS sends a 01.x Online Message and the terminals displays "CONNECTION".

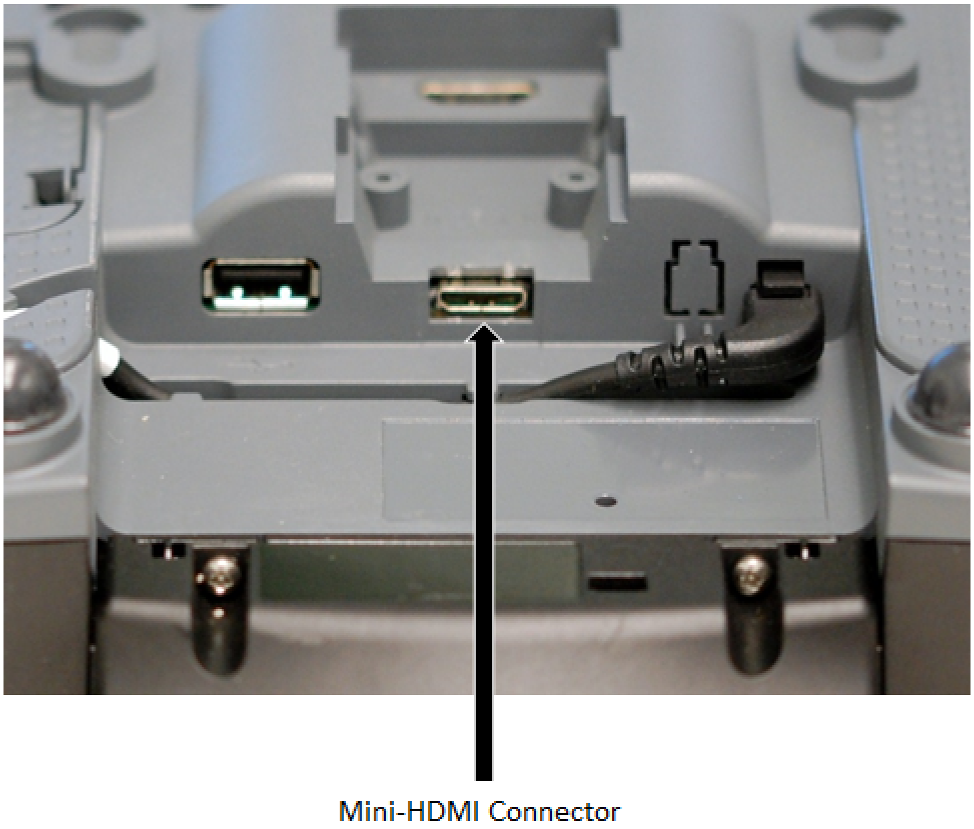
Example Base Association Using the 28.x Set Variable Request Message with Variable 820

Step	Data	POS	Terminal
Connect to terminal.		→	
Send 28.x Set Variable Request message with 14-character base Bluetooth (BT) address. BT address = '0B547F546C5398'	 28.1000008200B547F546C5398	→	
The terminal overwrites variable 820 with the base address provided in the 28.x message. This becomes the new BT address for base association. In this example, variable 820 = '0B547F546C5398'			
The POS sends a 29.x Get Variable Request message with variable 820 to retrieve the base BT address.	29.00000820	→	
The terminal returns a 29.x: Get Variable Response message with the base BT address.	 29.00008200B547F546C5398		←
The POS sends a 01.x Online Message to the terminal.	01.0TMS0TMS	→	
Terminal displays "CONNECTION".			

3.5 External Display for Telium Terminals

The external display feature is supported on iSC480 terminals via an HDMI port.

The following image illustrates the location of the Mini-HDMI connector on the bottom of the iSC480 terminal. To interface with an external display, connect a type C HDMI cable to this port.



Mini-HDMI Connector Location

4 Terminal Process Flow

RBA has a flow of standard processes, ready to use for the majority of financial transactions (see Standard Process Flow for more information). The order of this flow can be customized as follows:

- To alter the flow for all financial transactions, use the configuration parameters (see Configuring the Application for more information).
- To alter the data flow for a single transaction only, use the “on-demand” messages. When an on-demand message is received, RBA stops the execution of the current process and executes the new process. After the new process is finished, the RBA returns to the interrupted process or goes to the transaction start.

The order can be changed by setting parameters in the `config.dfs` file, or when a POS issues an on-demand message. Since each card type (debit, credit, an so on) requires specific processes, the standard flow can be configured for each card type.

Note
The iUC250 terminal does not support standard flow. It uses on-demand messages only.

4.1 EFT Overview

This section defines the communication protocol between a POS and Telium terminal. It also discusses the functional requirements placed on the store controller, user host, or third party switch as a result of this protocol.

There is a payment type referred to as Electronic Funds Transfer (EFT) which provides the customer with an electronic means of paying for goods or services received. This method requires that the customer has a debit card (a plastic card with an encoded magnetic stripe) issued by a financial institution and a Personal Identification Number (PIN) associated with the card and accounts.

In a Point Of Sale (POS) environment, the merchant provides an EFT terminal that the customer uses to make payment for his purchases. This terminal contains:

- A Magnetic Stripe Reader (MSR) for reading the information encoded on the debit card
- A PIN keypad for entering the personal identification number (in some environments, it is required that the PIN keypad is usable for numeric entry of other data such as dollar amount of transaction)
- A display for showing prompts or other information to the customer during the transaction

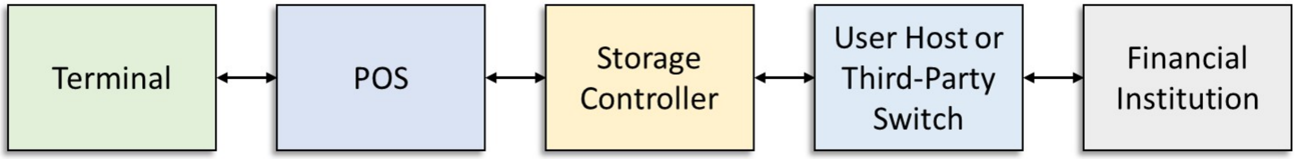
In a typical transaction, the cashier totals up the transaction then asks the customer how they want to pay. If the customer uses EFT as payment, the processing flow is as follows:

1. The cashier activates the EFT terminal.
2. The customer uses the EFT terminal to complete payment.
3. The terminal prompts the user through the process: swipe debit card, select account to be debited, enter PIN number, and authorize amount due.
4. The RBA then formats an authorization message with the information just received and forwards the message to the POS system.
5. The POS system in turn forwards that message to the proper financial institution for approval or disapproval.

6. The POS system receives the approval or disapproval message from the financial institution and forwards it back to the RBA.
7. If approved, the POS system accepts the amount as payment.

4.1.1 Assumptions

The following is an assumed typical configuration for our industry:



Assumed Points of Communication

4.1.2 Environment

The EFT environment is one of interactions between the customer, the merchant, and a financial institution. The simplest configuration is a terminal attached to the POS system, with the POS system attached via communications line to a single financial institution. Many of our merchants are already doing tender approval at their host location (e.g., credit, check authorization). It would therefore be a logical extension if their POS system used that same physical communication connection to route the EFT authorization request and response to the user host and have the user host "switch" to the proper financial institution. This also gives merchants the capability to maintain a certain level of control over the EFT process if these messages pass through their own host.

Since there may be several financial institutions involved with a single merchant, the merchant may choose to use a third party "switch" to manage EFT processing. These third party switches provide the capability to have only one line from the merchant to the switch. The switch exchanges the required authorization request and response message with the proper financial institutions on behalf of the merchant.

The communication protocol, message formats and operational procedures for each of these financial institutions and third party services are currently different. For this reason the following assumptions are made concerning the EFT environment for the POS system:

- Base store controller communication support allows the merchant the capability to participate in any of the configurations discussed above with some amount of user programming.
- The controller implements VISA Second Generation message formats.
- The controller assumes it is talking to a "switch," either third party or user host. This implies the controller communicates with only one message protocol and one message format (VISA II) for EFT.

4.1.3 Dependencies

For the EFT messages to work properly, the dependencies below must be met.

The switch must:

- Limit messages to a maximum length of 247 bytes, including the STX, ETX, and LRC control characters (most third-party switches are capable of this).
- Handle the VISA II parameter table loads to the terminal.

The POS must allow the POS operator to enter the account number and card expiration date on the POS keyboard if the terminal cannot read the card data, and send this data from the POS to the terminal.

The terminal must:

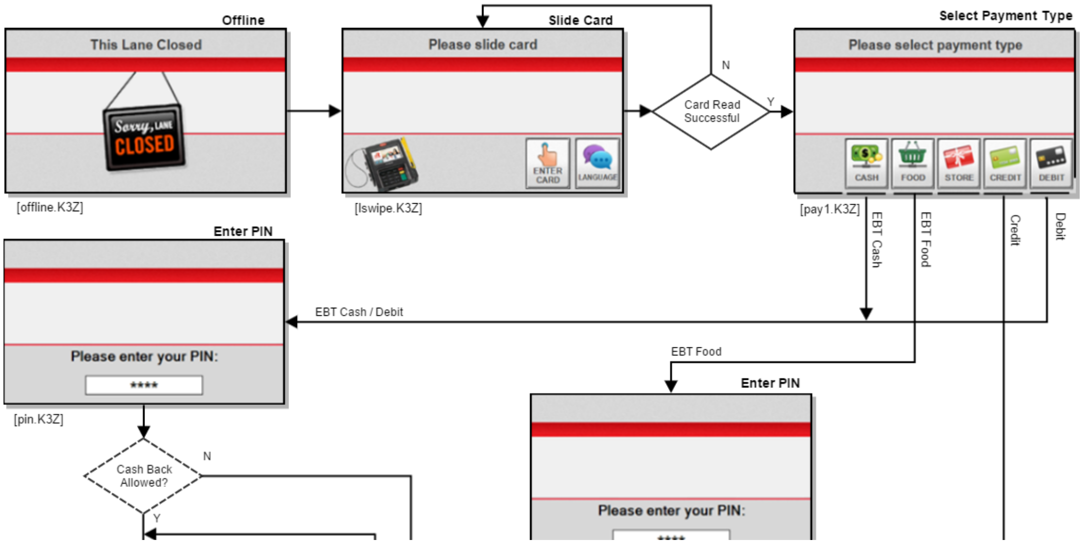
- Determine if a PIN is required, allow PIN keying, encrypt the PIN, and build the proper VISA messages for communication.
- Provide the capability to build a VISA authorization request message without receiving or showing an amount on the terminal.
- Provide the capability to show the amount due received from the POS and allow the customer to validate that amount or to enter and validate a different amount. Build the VISA authorization request message with the validated amount.
- The terminal remains at "Slide Card" until it reads data from a card swipe or receives the account number and card expiration date as entered from the POS, if the card cannot be read. It then collects the remaining required data at the terminal and builds the proper VISA authorization request message.
- Provide the capability for the POS to reject the amount in the authorization request message and have the terminal validate the new amount with the customer. The POS must then accept a new authorization request message containing the new amount.

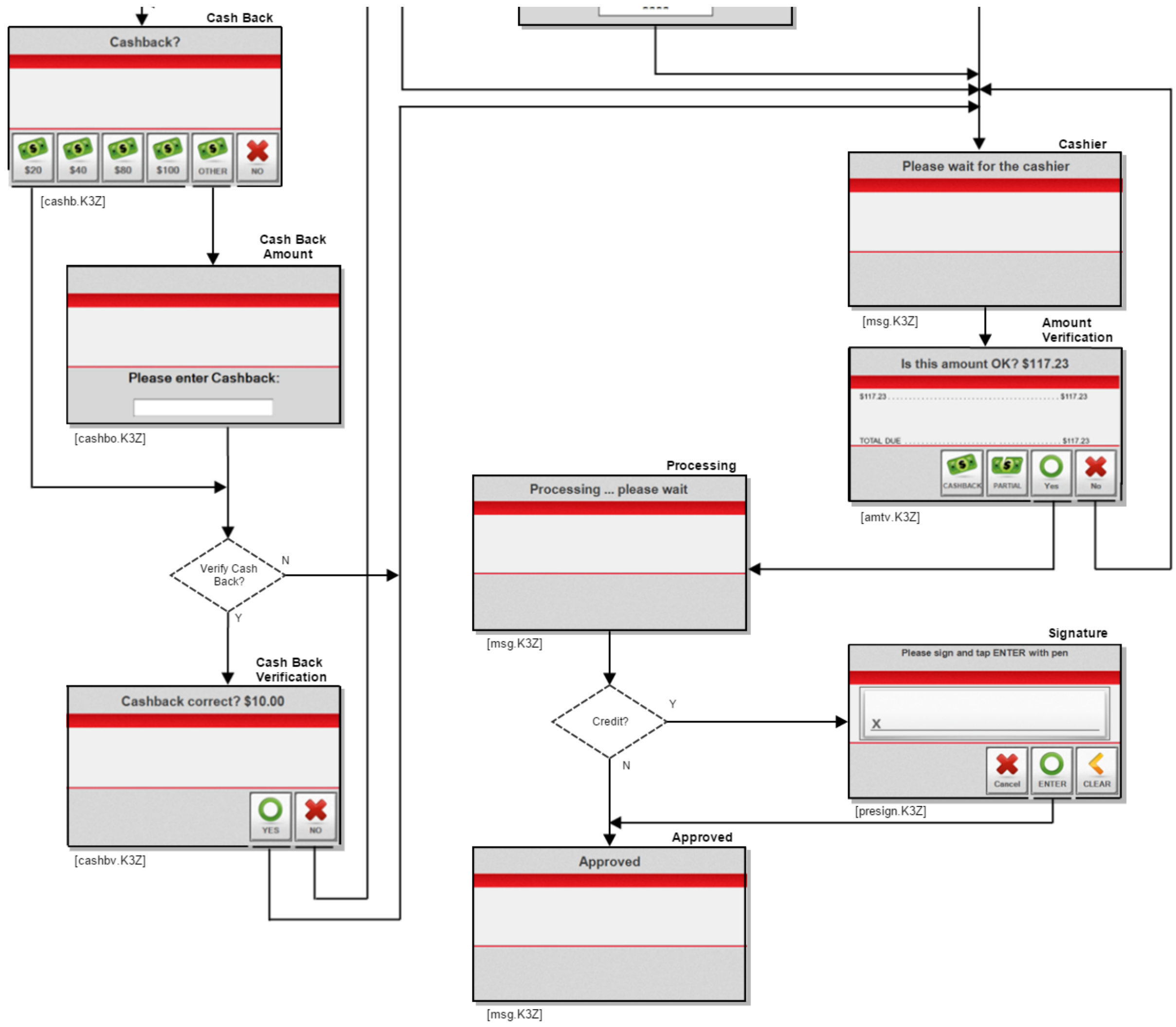
4.2 Standard Process Flow

The RBA standard processes are executed in a specific order. A typical process order, also called a flow or process flow, may be as follows:

- Select language → Swipe card → Select payment → Enter PIN → Enter cash back → Verify purchase amount → Authorization → Approve → Transaction End → Advertising.
- Swipe card → Select payment → Verify purchase amount → Signature → Authorization → Approve → Transaction End → Advertising

The following flow chart shows the high-level host process flow from the customer's perspective for Ingenico's Retail Base Application:





RBA Standard Process Flow

4.3 On-Demand Transaction Process

On-demand messages can be used to deviate from the standard transaction flow and create your own dynamic application. These messages can be initiated from the offline screen, except 31.x when card data is still required.

- 20.x Signature Message (on-demand)
- 21.x Numeric Input Request Message (on-demand)
- 23.x Card Read Request (on-demand)
- 24.x Form Entry Request (on-demand)
- 25.x Terms and Conditions Request (on-demand)
- 26.x Run Script Request (on-demand)
- 27.x Alpha Input Message (on-demand)

- 30.x Advertising Request Message (on-demand)
- 31.x PIN Entry Messages (on-demand)

On-demand messages cannot be nested. When these messages are received during the execution of another on-demand message, they are not executed. RBA sends a response message with a reject status and the execution of the current on-demand message continues. Exceptions to this process are:

- The 30.x message
- When the Automatic On-Demand Function Cancel parameter in Main Flow (mainFlow.dat) 0007_0028 is set to 1
- When the current on-demand message is 20.x and `mainFlow.dat` 0009_0006 (Save State on Signature Request) is set to 0

4.3.1 Offline On-Demand Transaction Recommendations

When performing an offline on-demand transaction, there are a few recommended deviations from standard transaction setup, as follows:

- The 00.x Offline Message and 15.6 Soft Reset Message can both reset an on-demand card read.
- A 15.8 message can dynamically reset any offline line display.
- The 00.x message returns the terminal to the `OFFLINE.K3Z` offline form. For on-demand transactions, the offline form is recommended to be modified as either:
    - A default screen that displays between offline and on-demand transactions (like a company logo, messages, and so on)
    - An acceptable interstitial screen before issuing the next 24.x or 30.x messages, or resuming any offline ads, if configured

See also Communication Messages for additional information.

4.4 Spin the BIN - BIN Lookup Process

The application has a few ways to automatically establish the payment type based on the cardholder account number. The account number may be retrieved from the terminal local magnetic stripe reader or contactless reader, or it can be received from the POS in a message.

The account number is associated with a card type, such as debit, credit, or gift. For example, an account number starting with 6011xxxxx could be a Discover card, while a 4000xxxx account could be a Visa card.

The application uses the following methods to establish the payment type:

- Internal Spin the BIN (STB). The payment is established by the terminal from data included in the local configuration.
- External STB. The payment information is received from the POS.

Each method can operate individually or with other methods. Each method has its own set of parameters listed in the config.dfs file.

When payment is selected, the application performs a final check to see if the Transaction Code for the selected payment is valid.

- If Transaction Code > 0, Continue with the transaction.

- If Transaction Code = 0, Display the Invalid Card Type prompt, reset the payment, reset the payment, and return to the Card Swipe screen.

This method is enabled or disabled by the configuration parameter listed in allBins.dat file, index 0099_0001, Enable BIN range checking (0 = off, 1 = on).

The default config.dfs file contains fourteen files, `bin0.dat` through `bin13.dat`. Each file contains a description of a specific card type, such as MasterCard, which applies to that card only. Each `binX.dat` file contains:

- The first few digits of the account
- Minimum and maximum number of digits in the account number
- List of transaction codes used with selected payment. The transaction code is part of the authorization message sent to the POS

4.4.1 How to Enable BIN Checking

The application can automatically identify a payment type for the card by setting values from the local configuration (internal Spin the BIN - STB) or sending a request to the POS to select the payment type (external STB).

If BIN range checking is enabled, the terminal compares the cardholder account with data from all binX.dat files. If there is no match, the terminal checks whether Spin the BIN (STB) is enabled.

- If enabled, the application goes to STB
- If it is not enabled, the terminal displays the payment screen with the payment buttons and prompts the cardholder to select the payment type

The BIN range checking configuration options common to all binX.dat files are listed in BIN Processing allBins section in config.dfs, which are:

- 0099_0001, Enable BIN range checking (0 = off, 1 = on)
- 0099_0002, Number of BIN ranges (up to 20)
- 0099_0003, BIN length is x digits. It selects how many digits of the cardholder account number are compared with numbers from bin0.dat to binX.dat files. Only the accounts first digits are used for comparison.

See BIN Lookup (stb.dat) for more information on configuring BIN processing settings.

4.4.2 Internal BIN Range Checking

Internal STB means that the payment selection is based on the local configuration data only. This function searches the bin0.dat to binX.dat files to find the payment. The payment type is included in the string listed at index 010x_0005.

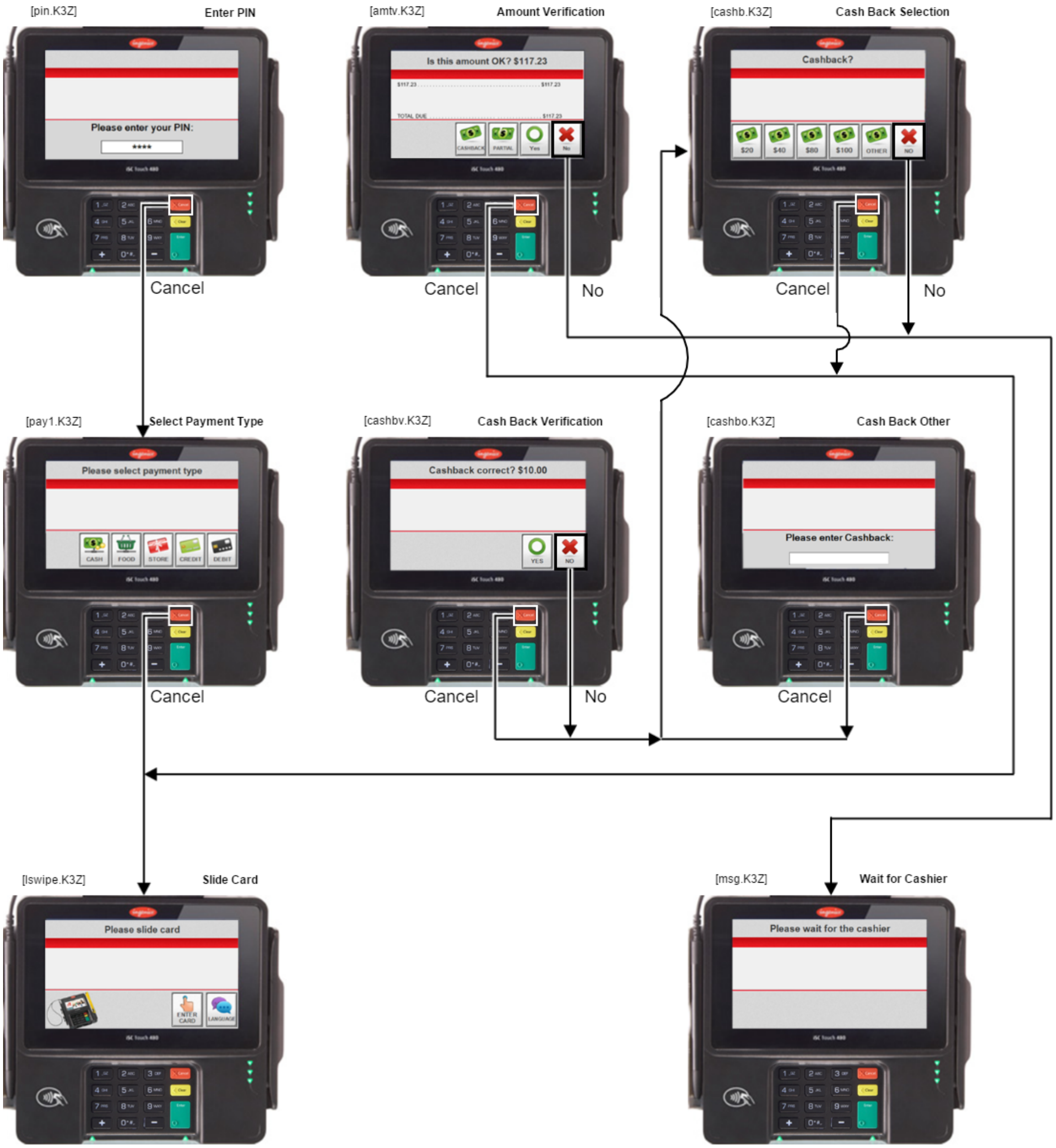
4.4.3 External BIN Range Checking

When the external STB is allowed to execute, the terminal sends a request message to the POS and waits for the POS response with the payment type. The termnal uses the received payment type to continue the transaction or prompt the cardholder to select the payment.

See 19.x BIN Lookup Message for more information.

4.5 Cancelling a Process

The following sections describe how the terminal handles a Cancel key press during various processes. Refer to the following diagram which illustrates the standard cancellation process.



terminals Standard No or Cancellation Process

4.5.1 Amount Verification

When the NO key is pressed during the verify amount state, the terminal always sends the 10.x Hard Reset Message, clears the amount, and waits for the purchase amount state.

When the CANCEL key is pressed during the verify purchase amount state, the terminal sends the 10.x message. The terminal goes back to the transaction start and the transaction is cleared along with the language selection.

4.5.2 Cash Back

The cashback process displays the following screens:

- **`cashb.K3Z`**: for iSC250/iSC350, screen with Fast Cash Back keys ($20, $40…) and OTHER; for iPP350, screen with Cash Back YES, NO buttons
    - NO key press: Skips the cashback selection and goes to the next terminal process
    - CANCEL key press: If 0007_0004 = 0, return to swipe.K3Z, otherwise return to lswipe.K3Z
- **`cashbo.K3Z`**: Screen to enter a cashback value.
    - CANCEL key press: Return to `cashb.K3Z`
- **`cashbv.K3Z`**: Cash Back verification screen with YES, NO, CANCEL buttons
    - CANCEL key press:
        - If 0002_0012 = 0, return to `cashbo.K3Z`
        - If 0002_0012 = 1, return to `cashb.K3Z`
        - If 0002_0012 = 2:
            - If 0007_0004 = 1, return to `lswipe.K3Z`
            - Otherwise return to `swipe.K3Z`
        - If 0002_0012 = 3, return to `pay1.K3Z`
    - NO key press: Return to `cashb.K3Z`

4.5.3 PIN Entry

When the CANCEL key is pressed during PIN entry, the terminal responds in one of the following ways:

- If the payment type was selected automatically or forced by the host in a message ([04.x Set Payment Type Request message), the terminal clears the payment selection, goes back to the payment selection screen, and lets the cardholder make a new payment selection.
- If the payment type was selected by the cardholder pressing a display key, terminal checks if the purchase amount is present in the terminal.
    - If the purchase amount is present, the terminal sends a 10.x message, clears the transaction along with the language selection, and goes to the transaction start.
    - If no purchase amount was received, the terminal clears the transaction along with the language selection, and goes to the transaction start.

4.5.3.1 Configuring the Cancel Key for PIN Entry

The CANCEL key can be configured to function as a Cancel key or as a Clear key. This is determined by setting parameter 0013_0022 in the compat.dat file. If at least one character is entered, the CANCEL key (when configured as a Clear key) clears the entered digits and restarts clear-text or PIN entry. If no digits are entered, then the CANCEL key cancels clear-text or PIN entry.

4.5.4 Signature

The CANCEL button on the Signature forms for the iSC250, iSC350 and iSC480 terminals is functional before the cardholder signs (pre-signature) only. When signing is initiated (post-signature):

- The CANCEL button is removed from the screen
- The CANCEL key on the keypad is processed as a CLEAR

For the on-demand signature request there is no pre-signature or post-signature state, and the Cancel button will always be displayed and processed as a CANCEL action.

4.5.5 Transaction Cancelled

If the CANCEL key is pressed and the transaction is terminated, the Transaction Cancelled message is displayed. The presence of the message is controlled by the configuration switch in the Main Flow section in the config.dfs file, index 0013_0004 (Show prompt Transaction Cancelled 0 = disabled, 1 - 255 = duration in 1/10 second).

If the CANCEL key is pressed during the Language Selection, Card Swipe, or Payment Selection process, the terminal initiates the following processes:

- If the purchase amount is present, the terminal sends a 10.x message, clears the transaction, returns the language selection to the default value, and goes to the transaction start
- If no purchase amount is received, the terminal clears the transaction, returns the language selection to the default value, and goes to the transaction start

4.5.6 Cancel after amount received

Sends 10.x message. Returns to `swipe.K3Z` or `lswipe.K3Z`, depending on settings.

4.5.7 Cancel no amount

Returns to `swipe.K3Z` or `lswipe.K3Z`, depending on settings.

4.6 Transaction End Process

The financial transaction ends in following situations:

- As a normal part of the terminal flow
- At request of the POS when one of the following messages is received:
    - 10.x - hard reset message is received
    - 15.x - soft reset message is received (some variations only)
    - 01.x - online message is received
    - 50.x or 00.x – authorization response message
- When the terminal detects a special condition, such as:
    - The cardholder pressed the CANCEL key. After that, if the amount is received, the terminal sends a 10.x message

    - Some of the configuration parameters are not present in the terminal, without these, the terminal cannot operate normally
    - At the end of the signature on-demand message execution

In this process, the terminal takes action according to the received message type, key press, or error condition. Possible actions are:

- Clear cardholder data, and start a new transaction
- Start advertising
- Exit the terminal transaction and go to the offline state

When the financial transaction is cleared, the terminal makes the following change:

- All data collected from cardholder: all account values, payment selection, amounts, language, and signature is deleted
- It increments the transaction counter, which is used by the 50.x authorization message
- It clears timers, buffers, and pointers - used internally to manage the transaction
- It clears the digital receipt based on two options:
    - 10.x message parameter value
    - RBA configuration switch listed in `mainFlow.dat` file, index 0007_0007 (Clear line-item display on reset):
        - 0 = Do not clear
        - 1 = Clear (display receipt)

At the transaction end, the cardholder can be notified about the result of the transaction through a text prompt. The text presence is controlled by the configuration parameter found in the Main

Flow section in `config.dfs`, Display Approved/Disapproved Message, index 0007_0022: (0 = Do not display, 1 - 65,000 = Duration of display in 1/10 second and in effect only if advertising is on). The prompt displays for five seconds. Next, the terminal might do one of the following actions based on configuration selections:

- Start advertisements
- Wait for a transaction reset message, such as the 10.x message
- Automatically reset the transaction and go to the transaction start

Here are examples of the transaction result texts. They change according to the executed processes:

- Approved (or equivalent translation) - from file `PROMPT.xml`, prompt ID 21
- Declined (or equivalent translation) - from file `PROMPT.xml`, prompt ID 22
- Invalid PIN. Please Re-enter. (or equivalent translation) - from the RBA PIN Prompts section of the `SECURPROMPT.xml` file, prompt ID 15
- Signature Accepted(or equivalent translation) - from file `PROMPT.xml`, prompt ID 92
- Input Accepted (or equivalent translation) - from file `PROMPTS.xml`, prompt ID 93
- Transaction Cancelled (or equivalent translation) - from file `PROMPT.xml`, prompt ID 23

The display of this prompt is controlled by index 0031_0023 from the Main Flow section in the `config.dfs` file. It is used when the CANCEL button is pressed and the terminal resets the transaction.

4.6.1 Configuring

the terminal local configuration provides the ability to control which forms display at the end of the transaction end process:

- When the transaction ends, RBA displays the Host Response for the amount of time specified by the configuration option listed in `mainFlow.dat` file, index 0007_0022, Display Approved/Disapproved Message Timer:
  - 0 = Do not display
  - 1 = Display until a reset is received
  - 2 - 255 = Time in 1/10 second
- After the Host Response message has timed out, the terminal displays advertising based on the configuration option listed in `mainFlow.dat` file, index 0007_0023, After Display Approved/Disapproved Message Timeout:
  - 0 = Reset
  - 1 = Go to advertising
  - 2 = Wait for reset

Note that there are certain restrictions associated with the advertising display parameter:

- Setting 0007_0023 = 1 requires that 0010_0001 be set to either 1 or 3
- Setting 0007_0023 = 2 requires that 0007_0022 be set to 1

4.6.2 Response Messages

Response messages use one of the following prompts:

- Accept (or equivalent translation) - from file `PROMPT.xml`, prompt ID 120
- Decline (or equivalent translation) - from file `PROMPT.xml`, prompt ID 121
- Invalid PIN. Please re-enter (or equivalent translation) - from the RBA PIN Prompts section of the `SECURPROMPT.xml` file, prompt ID 15

See Prompts for more information.

4.7 Clearing Transaction Data

All data from a transaction (transaction and cashback amounts, cardholder data) is cleared when one of the following messages is received:

- 00.x Offline Message
- 01.x Online Message
- 10.x Hard Reset Message
- 15.1 or 15.9 Soft Reset Message

4.7.1 PIN Data

Storage of PIN data depends on the flow implemented.

4.7.1.1 Standard Flow

PIN data remains in memory until cleared by any message that clears transaction data.

4.7.1.2 On-Demand Flow

Only data received from a 23.x Card Read Request (On-Demand) is stored in case of 31.x PIN Entry Messages (On-Demand). PIN data is never stored after the terminal sends a 31.x response.

5 Configuring the Application

This chapter describes RBA prompts and parameters that may be changed to customize the application for various Ingenico Telium terminals.

5.1 DFS Organization

Terminal configurations are stored in the terminal's memory, called the data file system (DFS). Data in the DFS memory is reprogrammable at run time, but it is preserved in case of power loss.

All .dfs files can be edited with any PC editor used for software development, or with a PC editor such as Notepad or WordPad. Only use editors that do not automatically insert hidden characters, such as font selections or page breaks.

The prompts file, as its name suggests, contains a collection of instructions or prompts used by RBA in various situations. Prompts inside the `PROMPT.xml` or `SECURPROMPT.xml` files are used for one of three functions: prompts that display on the terminal screen, prompts that are sent to the POS, and prompts that are used as button labels.

The config.dfs file contains a collection of many parameters, grouped into individual files, which specify a process such as PIN entry, card swipe, or advertising. Since `config.dfs` is not language dependent, only one file is needed.

> **Info**
> RBA users who wish to edit configuration parameters should always make sure they use the correct version of the `config.dfs` file.

Before the configuration parameters can be loaded into the terminal, the .dfs file must be translated into the terminal's internal .dat format using the `CTR_TRANS.EXE` utility. Only translated files can be loaded into the terminal.

At run time, the RBA has read/write access to its configuration files. Access to the dat files is private, limited to the RBA only. They cannot be accessed by other applications which reside in the terminal.

If any of the RBA configuration parameters are not present in the terminal at run time, the RBA cannot function correctly, so it goes to offline mode.

5.1.1 Data Description

All timer values in the config.dfs file are entered in 1/10th of a second.

All monetary values, such as maximum cash back limits, are expressed in dollars only; no cents are allowed. The only exception is the cash back limit value received in the 28.x message, which is entered in cents. This exception makes the message compatible with Ingenico's legacy terminals.

All examples in this chapter are for the English language.

5.1.2 Data Line

A data line is comprised of three possible entries, with at least one data element. The entries are:

- Informational. An informational element is delimited with a single quote ('). This is a 9-character field using an xxxx_yyyy format. This is an optional element but if used must be the first element in a data line.
- Data. The data element is delimited with a double quote ("). This element's value may be continued on the next line with the use of a comma (,) after the terminating quotation mark.
- Comment. A comment may be delimited by a /* or // character set. Everything after a comment is ignored.

The following table gives an example of various data line elements.

Valid Entries

Information	Data	Comment
’0007_0002’	“1”	/* 1, Default language */
’0002_0001’	“99999”	/* 99999, Max cash back value */
’0010_0007’	“50”	/* 50, Time to display each ad */
’0030_0001’	“offline.K3Z”	/* offline form */
’0015_0003’	“10.15.1.149“	/* IP address of server */

5.1.3 File Name Line

The file name contains the following four elements:

- Directive: Either “Write Public” or “Write Private”
- Path and file name: The path and filename element is the name of the file to be generate in quotation mark, for example, `"cashback.dat"`.
- Five-digit format code number: A three-digit data identifier and a two-digit version code.
- Format or revision information: The length of this item is set to 12 bytes.

A comma separates the second, third, and fourth elements.

Use the following syntax rules for naming files. The syntax of the file name is checked by the application. The available characters for file names are:

- The first character in a file name can be {'a'..'z','A'..'Z'}
- The last character in a file name can be {'0'..'9','a'..'z','A'..'Z','_','$','.'}
- The second through next to last character can be {'0'..'9','a'..'z','A'..'Z','-','_','$','.'}

5.1.4 File Rules

A single DFS file typically contains groups of definitions, each headed by the name of the DAT file to which the group will be translated. From a single-source DFS file, several groups of parameters can be translated into individual downloadable DAT files. For example:

- `Config.dfs` is translated into `cashback.dat`, `bin1.dat`, etc.

In config.dfs, the parameters are listed by groups of files. Each group has a header followed by data. The header contains the group file name, such as `msr.dat`, `pin.dat`, or `cashBack.dat`.

Data in the .DFS file consists of two types of data entry:

- File name line, which contains information about the name and location of the file in the terminal
- Data line, which consists of parameters within the file

Each type is described in the sections that follow.

A DFS file must have at least one DAT file name line. A single DFS file can contain many DAT file name definitions. The file name line must be followed by a list of configuration parameters, which are also called data lines. Data lines listed after the DAT file name are added to the DAT file.

A comma (,) following a data line string acts as the line continuation. Data from the first line is concatenated with data from the following line until there is no comma character after the last data string. All data is entered in ASCII string format, enclosed in quotation marks, such as “Please enter PIN:” Comments are allowed in a DFS file. They use either /* */ or // format.

5.2 DAT Files

DAT files are files that reside in an Ingenico terminal. The name and extension of these files are determined at data entry. Each DAT file will consist of two parts: a header and data.

- The header contains basic information about file size, data offset, data format, and revision information, followed by the data. All data is contiguous. The format field determines the format of the data. The file size and data offset will each be 4 bytes in length. The format will be ASCII hex. The format field is five bytes in length. The revision information is 12 bytes in length.
- The data is delimited with the binary number zero (0x0).

Do not edit the DAT file because it is difficult to determine the meaning of the data. Instead, edit the DFS file and run the translator from the DFS to DAT format.

5.2.1 Advertising Parameters (ads.dat)

5.2.1.1 Overview

This section describes the parameters used to configure advertising options. These parameters are found in the `config.dfs` file under the heading, Advertising, with the filename, `ads.dat`.

Advertising can be started automatically by the terminal or on demand by the POS.

5.2.1.2 Automatic Startup of Advertising

Automatic startup is executed by the terminal data flow and occurs in the following situations:

- When the terminal is offline and the following conditions are met, the terminal proceeds directly to the advertising screen once the 01.x Online Message is received.
    - Configuration parameter 0010_0003 is enabled (not set to 0).
    - Configuration parameter 0007_0010 is set to 0.

If configuration parameter 0007_0010 is set to 1 then the terminal starts a new transaction following the 01.x: Online Message.

- By default, configuration parameters 0010_0001 and 0010_0003 are set to 0 to disable offline and online advertising. These parameters must be enabled in order for the terminal to accept any 30.x Advertising Request Message (On-Demand) and proceed with advertising.
- When the terminal is offline, and the 0010_0001 configuration parameter value is not 0, offline advertising is enabled and continues until terminated by the 01.x: Online Message.
- When the terminal goes to the transaction end and advertising is enabled, advertising continues until terminated by a 00.x, 01.x, 10.x, 15.0, 15.6, 20.x, 21.x, or 23.x message.

5.2.1.3 On-Demand Startup of Advertising

When the 30.x: request is received from the POS, the terminal stops executing the current process and proceeds with the advertisements. This process is ignored when the terminal is in the offline state or it is executing another on-demand request.

The order in which the advertising bitmaps are displayed is controlled by the configuration options listed in the Advertising section of `config.dfs`. The following table provides a description for each option from the `ads.dat` file, and explains how these options are used by the terminal.

Advertising Parameters

Parameter Name	DFS Data Index	Default Value	Description
Offline Advertising Mode	0010_0001	0	• 0 = Off. • 1 = Display single ad. • 2 = Display single ad, changing each time terminal goes offline. • 3 = Cycle through all ads once (timed). • 4 = Cycle through all ads and repeat (timed).
Transaction Advertising Mode	0010_0002	0	• 0 = Off. • 1 = Display single ad. • 2 = Display single ad, changing with each transaction. • 3 = Display single ad, changing with each screen. • 4 = Cycle through all ads once (timed). • 5 = Cycle through all ads and repeat (timed).

Parameter Name	DFS Data Index	Default Value	Description
End of Transaction Advertising Mode	0010_0003	0	• 0 = Off. • 1 = Display single ad then reset. • 2 = Display single ad until reset received. • 3 = Display single ad, changing each time, then reset. • 4 = Display single ad, changing each time, then wait for reset. • 5 = Cycle through all ads 1 time, then wait for reset (timed). • 6 = Cycle through all ads until reset (timed).
Allow Display of Online Advertisements	0010_0006	0	• 0 = Online advertising is disabled. • 1 = Online advertising is enabled.
Allow Offline Video Download and Display from Server	0010_0007	0	• 0 = Offline video download and display is disabled. Advertisements will follow configurations in 0010_0001 through 0010_0004. • 1 = Offline video download and display is enabled.
	0010_0008	OFFLINEVID.K3Z	Form to display when 0010_0007 is set to 1.
	0010_0011	AD1.K3Z 50 1111111 00:00 24:00 101	Advertisement 1.
	0010_0012	AD2.K3Z 50 1111111 00:00 24:00 101	Advertisement 2.

Parameter Name	DFS Data Index	Default Value	Description
	0010_0013	AD3.K3Z 50 1111111 00:00 24:00 101	Advertisement 3.
	0010_0014	AD4.K3Z 50 1111111 00:00 24:00 101	Advertisement 4.
	0010_0015	AD5.K3Z 50 1111111 00:00 24:00 101	Advertisement 5.
	0010_0016	AD6.K3Z 50 1111111 00:00 24:00 101	Advertisement 6.
	0010_0017	AD7.K3Z 50 1111111 00:00 24:00 101	Advertisement 7.
	0010_0018	AD8.K3Z 50 1111111 00:00 24:00 101	Advertisement 8.
	0010_0019	AD9.K3Z 50 1111111 00:00 24:00 101	Advertisement 9.
	0010_0020	AD10.K3Z 50 1111111 00:00 24:00 101	Advertisement 10.
	0010_0021	AD11.K3Z 50 1111111 00:00 24:00 101	Advertisement 11.
	0010_0022	AD12.K3Z 50 1111111 00:00 24:00 101	Advertisement 12.
	0010_0023	AD13.K3Z 50 1111111 00:00 24:00 101	Advertisement 13.
	0010_0024	ADV1.K3Z 100 1111111 00:00 24:00 010	Advertisement 14.

Parameter Name	DFS Data Index	Default Value	Description
	0010_0025	ADV2.K3Z 100 1111111 00:00 24:00 010	Advertisement 15.
	0010_0026	ADV3.K3Z 100 1111111 00:00 24:00 010	Advertisement 16.
	0010_0027	ADV4.K3Z 100 1111111 00:00 24:00 010	Advertisement 17.
	0010_0028	ADV5.K3Z 100 1111111 00:00 24:00 010	Advertisement 18.
	0010_0029	ADV6.K3Z 100 1111111 00:00 24:00 010	Advertisement 19.
	0010_0030	AD20.K3Z 50 1111111 00:00 24:00 000	Advertisement 20.
	0010_0031	AD21.K3Z 50 1111111 00:00 24:00 000	Advertisement 21.
	0010_0032	AD22.K3Z 50 1111111 00:00 24:00 000	Advertisement 22.
	0010_0033	AD23.K3Z 50 1111111 00:00 24:00 000	Advertisement 23.
	0010_0034	AD24.K3Z 50 1111111 00:00 24:00 000	Advertisement 24.
	0010_0035	AD25.K3Z 50 1111111 00:00 24:00 000	Advertisement 25.
	0010_0036	AD26.K3Z 50 1111111 00:00 24:00 000	Advertisement 26.

Parameter Name	DFS Data Index	Default Value	Description
	0010_0037	AD27.K3Z 50 1111111 00:00 24:00 000	Advertisement 27.
	0010_0038	AD28.K3Z 50 1111111 00:00 24:00 000	Advertisement 28.
	0010_0039	AD29.K3Z 50 1111111 00:00 24:00 000	Advertisement 29.
	0010_0040	AD30.K3Z 50 1111111 00:00 24:00 000	Advertisement 30.

Advertisement duration is used if advertisements are set to recycle. For advertisement scheduling to work, the terminal date and time must be set via the 28.x Set Variable Request message.

Image file names must be in upper case with a supported image type. All .HTM files may be in either case, but must match the file.

```
/* |||||||||||| Duration in 1/10th of seconds (0 = no timeout) */
/* |||||||||||| ||| */
/* |||||||||||| ||| Day to display this ad (0 = Do not display, 1 = Display) */
/* |||||||||||| ||| SMTWTFS */
/* |||||||||||| ||| ||||||| Start End (Time of day to display this ad. Use 24 hour */
/* |||||||||||| ||| ||||||| HH:MM HH:MM format. Start time must be before end time.) */
/* |||||||||||| ||| ||||||| ||||| ||||| */
/* |||||||||||| ||| ||||||| ||||| ||||| Mode */
/* |||||||||||| ||| ||||||| ||||| ||||| Offline */
/* |||||||||||| ||| ||||||| ||||| ||||| |During transaction */
/* |||||||||||| ||| ||||||| ||||| ||||| ||End of transaction */
/* |||||||||||| ||| ||||||| ||||| ||||| ||| */
```

5.2.2 Barcode Parameters (barcode.dat)

This section describes the parameters used to configure barcode reading capabilities of the terminal. These parameters are found in the `config.dfs` file under the heading, Barcode, and filename `barcode.dat`.

<table>
<tr><th>Parameter Name</th><th>DFS Data Index</th><th>Default Value</th><th>Description</th></tr>
<tr><td>Barcode Scanner Type</td><td>0015_0001</td><td>0</td><td>Sets the type of the attached barcode scanner:
• 0 = None
• 1 = Informatics Wasp WCS3905 CCD Scanner
• 2 = iSMP or iSMP Companion</td></tr>
<tr><td>Keyboard Character Mapping</td><td>0015_0002</td><td>1</td><td>Sets the way barcodes are scanned:
• 0 = Raw scan codes
• 1 = Standard US 102 key keyboard</td></tr>
<tr><td>Scan Mode</td><td>0015_0003</td><td>1</td><td>Scan mode (iSMP only):
• 1 = Single scan
• 2 = Multi scan</td></tr>
<tr><td>Image Mode</td><td>0015_0004</td><td>2</td><td>Image mode (iSMP only):
• 1 = 1D barcodes only
• 2 = 1D and 2D barcodes
• 3 = 1D and 2D barcodes for bright environments
• 4 = 1D and 2D barcodes for shiny or reflective surfaces</td></tr>
<tr><td>Illumination Mode</td><td>0015_0005</td><td>3</td><td>Illumination mode (iSMP only):
<table>
<tr><th>Value</th><th>Scan LED</th><th>Aimer LED</th></tr>
<tr><td>0</td><td>OFF</td><td>OFF</td></tr>
<tr><td>1</td><td><b>ON</b></td><td>OFF</td></tr>
<tr><td>2</td><td>OFF</td><td><b>ON</b></td></tr>
<tr><td>3</td><td><b>ON</b></td><td><b>ON</b></td></tr>
</table>
</td></tr>
</table>

Parameter Name	DFS Data Index	Default Value	Description
Lighting Mode	0015_0006	0	Lighting mode (iSMP only): • 0 = Shorter exposure time • 1 = Longer exposure time (for shiny or reflective surfaces; see **0015_0004** : Image mode :)
Trigger Mode	0015_0007	0	Trigger mode (iSMP only): • 0 = Physical trigger disabled • 1 = Physical trigger enabled to scan barcodes

<table>
<tr><th>Parameter Name</th><th>DFS Data Index</th><th>Default Value</th><th>Description</th></tr>
<tr><td>Symbologies</td><td>0015_0008</td><td>" "</td><td>
A comma-separated list of (non-negative) decimal codes corresponding to barcode symbologies to enable:
Examples:
• " " = Disable all barcode symbologies
• 00 = Enable all barcode symbologies
• 13,23,33,41 = Enable Code39, Code128, PDF417, and QR barcodes
<b>Note:</b> 0/00 can be used as a solitary code to enable all symbologies.
<b>List of supported (iSMP) symbologies</b>:
<table>
<tr><th>Value</th><th>Symbology</th><th>Value</th><th>Symbology</th><th>Value</th><th>Symbology</th></tr>
<tr><td>1 / 01</td><td>EAN-13</td><td>17</td><td>Matrix 2 of 5</td><td>33</td><td>PDF417</td></tr>
<tr><td>2 / 02</td><td>EAN-8</td><td>19</td><td>Codabar</td><td>34</td><td>GS1-128</td></tr>
<tr><td>3 / 03</td><td>UPC-A</td><td>21</td><td>MSI</td><td>35</td><td>ISBT128</td></tr>
<tr><td>4 / 04</td><td>UPC-E</td><td>22</td><td>Plessey</td><td>36</td><td>Micro PDF</td></tr>
<tr><td>7 / 07</td><td>UPC-A with addon2</td><td>23</td><td>Code 128</td><td>37</td><td>GS1 DataBar Omni-Directional</td></tr>
<tr><td>8 / 08</td><td>UPC-E with addon2</td><td>25</td><td>Code 93</td><td>38</td><td>GS1 DataBar Limited</td></tr>
<tr><td>11</td><td>UPC-A with addon5</td><td>26</td><td>Code 11</td><td>39</td><td>GS1 DataBar Expanded</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Parameter Name</th><th>DFS Data Index</th><th>Default Value</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>Symbology</th><th>Value</th><th>Symbology</th><th>Value</th><th>Symbology</th></tr>
<tr><td>12</td><td>UPC-E with addon5</td><td>27</td><td>Telepen</td><td>40</td><td>DataMatrix</td></tr>
<tr><td>13</td><td>Code 39</td><td>29</td><td>Code 39 CPI (aka, "Italian CPI")</td><td>41</td><td>QR Code</td></tr>
<tr><td>15</td><td>Interleaved 2 of 5</td><td>30</td><td>Codablock A</td><td>42</td><td>MaxiCode</td></tr>
<tr><td>16</td><td>Standard 2 of 5</td><td>31</td><td>Codablock F</td><td>74</td><td>AZTEC</td></tr>
</table>
</td></tr>
<tr><td>Symbology Encryption</td><td>0015_0009</td><td>No default</td><td>Symbology encryption. A comma-separated list of (non-negative) decimal codes corresponding to barcode symbologies with data that will be encrypted in 95. messages. Use the same format as 0015_0008 : Symbologies</td></tr>
<tr><td>LED Intensity</td><td>0015-0011</td><td>99</td><td>The percentage of LED illumination intensity. The value must be between 0 - 99.</td></tr>
</table>

5.2.3 BIN Lookup (stb.dat)

This section describes the BIN lookup parameters. STB stands for spin the BIN. This section is organized in the order you would write the BIN parameters. These are the records that configure PIN Encouragement software support.

After a card is swiped, the terminal's internal Spin the BIN (STB) feature searches the look-up tables listed in the BIN ranges and checks them against the account number on the card. If the account is included in the list, the payment type is retrieved from that list also. The application searches the card configuration table (see Card Configuration Table) for card handling information. The RBA may also request the payment from the host. If the payment type is not selected by the host and the STB search fails, the customer is prompted to make the selection by pressing a button on the terminal screen.

The source data is located in the `config.dfs` file under the heading, STB, and filename `stb.dat`.

Parameter Name	DFS Data Index	Default Value	Description
External Spin the BIN Search	0005_0002	0	This parameter enables BIN lookup through another program's lookup table (outside RBA). It enables PIN Encouragement message support. Setting it to "3" enables support for communicating directly to a PIN Encouragement server via Ethernet. There is some limited BIN lookup ability built into the application. • 0 = Disable (default) • 1 = Enable via host (send message as soon as card is swiped) • 2 = Enable via host (send message after receiving 13.x Amount Message from the POS) • 4 = Enable using IBM StorePay method. • 5 = Enable via host (send message after receiving an empty 19.x BIN Lookup Message from the POS).
Spin the BIN Search Table Order	0005_0003	0	This parameter defines whether to search the internal RBA STB database before or after performing the external STB search. The second lookup is only executed if the first lookup fails to identify the card. • 0 = Search external STB table first (default) • 1 = Search internal STB table first
Append Account Tracks in Message 19.x	0005_0004	0	If this parameter is enabled (1, append account tracks), Track 1 and 2 data will be included in the 19.x message. Since track data is not usually required, this option is off by default. • 0 = Disable - do not append account tracks (default) • 1 = Enable - append account tracks

Parameter Name	DFS Data Index	Default Value	Description
BIN Lookup Message 19.x Response Timeout	0005_0005	50	This parameter defines how much time (in 1/10th of a second) to allow the BIN Lookup message to spend searching before it will time out. • 0 = Disable timeout. Wait until either response message or a reset message is received (default) • 1 - 500 = valid values
Include Account Number Check Digit	0005_0006	1	If set to 0, the check digit (last digit) of the account number is stripped before adding it to the PIN Encouragement Request message • 0 = No check digit • 1 = Include check digit (default)
STB Timeout Destination	0005_0007	1	This parameter sets the action if the STB request times out. • 1 - 9 = pinX form is displayed (where X is the number of this setting) • A - P = Assume this payment type
Minimum Clear Digits	0005_0008	6	This parameter sets the minimum leading digits that must be sent when the MSR information is encrypted. • 6 – 9 = Number of leading digits sent when MSR information is encrypted
Delay after Trigger Message	0005_0009	0	This parameter defines the amount of time to delay after receiving the 13.x or 19.x trigger message. Only used when '0005_0002' is set to 2 or 5. Required for slow POS systems. • 0 = disabled • 1 or greater = time in 1/10ths seconds
Append Service Code	0005_0010	0	Append a service code to 19.x: BIN Lookup Messages. • 0 = Do not append service code • 1 = Append service code

5.2.4 BIN Processing (allBins.dat, bin0.dat - bin20.dat)

This section describes the parameters used to determine the bank identification number (BIN) processing information. A BIN identifies the account number of a payment card. When a card is read or entered manually, the terminal's BIN range checking feature searches the look-up tables listed in the BIN ranges and checks them against the account number on the card. If the account is found in the list, the payment type is retrieved as illustrated in the following look-up table example:

Basic BIN Lookup Table

Card	Prefix	Length
AMEX	34, 37	15
Diner's Club	300 - 305, 36	14
Discover	6011, 622126 - 622925, 644 - 649, 65	16
JCB	3528 - 3589	16
MasterCard	51 - 55, 2221 - 2720	16
VISA	4	13, 16

The application also searches the Card Configuration Table for card handling information.

RBA can also request the payment type from the host using the 19.x BIN Lookup Message message, if the 0005_0002 parameter is set in the BIN Lookup (stb.dat) configuration. If the payment type is not selected by the host, the cardholder is prompted to make the selection by pressing a button on the screen.

In the `config.dfs` file, these parameters are listed under the heading **BIN Processing**. The file names in this section are `allbins.dat`, and `bin0.dat` through `bin13.dat`. You can add BIN table entries up to `bin30.dat`.

5.2.4.1 All Bins (`allBins.dat`)

This section describes the parameters listed under filename `allBins.dat`.

allBins.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
Enable BIN Range Checking	0099_0001	0	Specifies whether to use BIN range checking: • 0 = Disable • 1 = Enable (must be set to 1 if using PayPal) Icon Because S1 encryption requires Mod-10, the Mod-10 flag in each `bin(x).dat` file must be set to 1.
Number of BIN Ranges	0099_0002	13	Sets the number of BIN files to search in the `config.dfs` file. When the value is 13, the files `bin1.dat` to `bin13.dat` are searched. File `bin0.dat` contains the default values only. If set to 0, all cards are considered valid, and no BIN range parameters are required.
BIN Length is x Digits	0099_0003	6	Specifies the number of digits in the account number to test. Testing starts from the account's first digit. For example: If the number of digits to test is six, all remaining digits after the sixth digit are not considered.

5.2.4.2 BIN Table Contents

The first part of the DFS Data Index (0100 to 0130) represents a particular card type (e.g., VISA, MasterCard), i.e. one particular BIN table entry.

The second part of the DFS Data Index (0001 to 0006) indicates one of the following parameters for a BIN table entry:

- 0001 = Start BIN range
- 0002 = End BIN range
- 0003 = Minimum account length
- 0004 = Maximum account length
- 0005 = Processing flags

- 0006 = Card sources

For example, the default BIN table entry for Visa, `bin8.dat`, consists of six parameters, 0108_0001 through 0108_0006.

In all cases (except `bin0.dat`, as described below), the meaning of the six parameters is as follows:

DFS Data Index	Example Value	Description
01xx_0001	`400000`	Start of BIN range (lowest BIN number in the range)
01xx_0002	`499999`	End of BIN range (highest BIN number in the range)
01xx_0003	`13`	Minimum account number length
01xx_0004	`16`	Maximum account number length Note, the minimum and maximum account number lengths may be used to distinguish between cards that may have similar prefixes but diverse lengths.

DFS Data Index	Example Value	Description

01xx_0005	`000111`
	`102000000000000000`
	`00000000000000`
	`112100000000000000`
	`00000000000000`
	`122200000000000000`
	`00000000000000`
	`132300000000000000`
	`00000000000000`
		Processing flags. The flags are represented as a string of 134 characters, treated as five fields (the line breaks in the Example Value are only for readability and should not be used in the configuration files).
		The first field consists of six characters as follows:
		• First character = card type; values can be:
		◦ A-P - references a card configuration in the Card Configuration (cards.dat)
		◦ 0-9 - selects one of the payment menus, PAYx.K3Z. (PAY1.K3Z is provided with RBA; merchants can add custom menus if desired.)
		• Second character = reserved
		• Third character = indicates if Mod10 checking is enabled (1 = enable, 0 = disable)
		• Fourth character = whether to prompt for expiration date during manual entry (1 = yes, 0 = no)
		• Fifth character = whether to prompt for CVV during manual entry (1 = yes, 0 = no)
		• Sixth character = how to decode Fleet Card prompting information for this BIN range:
		◦ 0 = Do not decode
		◦ 1 = Decode using VISA Fleet
		◦ 2 = Decode using MasterCard Fleet
		◦ 3 = Decode using Fleet One
		◦ 4 = Decode using Voyager
		◦ 5 = Decode using WEX Legacy
		◦ 6 = Decode using WEX V3
		The remaining four fields specify transaction codes for each payment type, to be included in the 50.x Authorization Request.
		• There are four 32-digit strings, corresponding to the four transaction types: Sale, Void, Return, and Void Return.
		• For each transaction type, the 32-digit string consists of a series of two-digit transaction codes, one code for each of the sixteen card configurations, A-P.
		• Effectively, these four fields convert the payment type (e.g., debit) and the transaction type (e.g., void) into the transaction code.

DFS Data Index	Example Value	Description
		- If the two-digit transaction code is 00, the combination of transaction type and payment type is not allowed for the BIN range. In the Example Value shown, for this BIN range, all four transaction types are supported for card configurations (payment types) A and B, only.
01xx_0006	`MCSHcm`	Card sources included in this BIN range. This is a string containing any or all of these characters: - M = MSR - C = Contactless (MSR or EMV) - S = Smart card (e.g. EMV, WIC, memory) - c = Coupon or key card - m = Mobile - H = Manual (Hand) entry - A = Account message entry - ? = Unknown or invalid card type This field can handle card sources differently. For example, a merchant might want to allow a choice of credit or debit if a card is swiped or inserted, but allow credit only if a card is tapped. In this case, two BIN table entries could be defined with the same BIN ranges, but with MSH as the Card Sources for one, and Ccm as the Card Sources for the other.

5.2.4.3 BIN Table Defaults (`bin0.dat`)

The `bin0.dat` file provides default values that are used when BIN range checking is turned off. It is not used if BIN range checking is enabled (0099_0001 is set to 1).

DFS Data Index	Example Value	Description
0100_0001		Reserved
0100_0002		Reserved
0100_0003	`12`	Minimum account number length
0100_0004	`24`	Maximum account number length

DFS Data Index	Example Value	Description

0100_0005	`000110`
	`102030405060708090A0B0C0D0E0F0G0`
	`112131415161718191A1B1C1D1E1F1G1`
	`122232425262728292A2B2C2D2E2F2G2`
	`132333435363738393A3B3C3D3E3F3G3`
		Processing flag defaults as described under *BIN Table Contents*

0100_0006		Reserved

5.2.4.4 *BIN Table Entries (`bin1.dat` - `bin30.dat`)*

The tables for `bin1.dat` through `bin13.dat` show the default settings for common cards. Custom `binx.dat` files can be included by setting the 0099_0002 parameter appropriately.

PayPal Discover, bin1.dat

DFS Data Index	Default Value	Description
0101_0001	601104	Start of BIN range. Icon Due to the value of parameter 0040_0008 (PayPal/Discover BIN table number), this BIN table number is skipped when doing a BIN lookup with PayPal disabled (parameter 0040_0006 is set to 0). Do not change this BIN table entry to replace it with a different card and assume that it will work simply because PayPal is disabled. Do not reassign bin1 or bin12 to non-payment cards.
0101_0002	601104	End of BIN range.
0101_0003	14	Minimum account length.

DFS Data Index	Default Value	Description
0101_0004	16	Maximum account length.

0101_0005	`G00110`
	`000000000000700000000000000`
	`00000`
	`000000000000710000000000000`
	`00000`
	`000000000000720000000000000`
	`00000`
	`000000000000730000000000000`
	`00000`
		Processing flags.

0101_0006	MCSHcm	Card sources.

Discover 1, bin2.dat

DFS Data Index	Default Value	Description
0102_0001	601100	Start of BIN range.
0102_0002	601199	End of BIN range.
0102_0003	14	Minimum account length.
0102_0004	16	Maximum account length.

0102_0005	`000110`
	`00200000000000000000000000000000`
	`00210000000000000000000000000000`
	`00220000000000000000000000000000`
	`00230000000000000000000000000000`
		Processing flags.

0102_0006	MCSHcm	Card sources.

Discover 2, bin3.dat

DFS Data Index	Default Value	Description
0103_0001	622126	Start of BIN range.
0103_0002	622925	End of BIN range.
0103_0003	14	Minimum account length.
0103_0004	16	Maximum account length.

DFS Data Index	Default Value	Description
0103_0005	000110 00200000000000000000000000000000 00210000000000000000000000000000 00220000000000000000000000000000 00230000000000000000000000000000	Processing flags.
0103_0006	MCSHcm	Card sources.

Discover 3, bin4.dat

DFS Data Index	Default Value	Description
0104_0001	644000	Start of BIN range.
0104_0002	649999	End of BIN range.
0104_0003	14	Minimum account length.
0104_0004	16	Maximum account length.
0104_0005	000110 00200000000000000000000000000000 00210000000000000000000000000000 00220000000000000000000000000000 00230000000000000000000000000000	Processing flagss.
0104_0006	MCSHcm	Card sources.

Discover 4, bin5.dat

DFS Data Index	Default Value	Description
0105_0001	650000	Start of BIN range.
0105_0002	659999	End of BIN range.
0105_0003	14	Minimum account length.
0105_0004	16	Maximum account length.
0105_0005	000110 00200000000000000000000000000000 00210000000000000000000000000000 00220000000000000000000000000000 00230000000000000000000000000000	Processing flags.
0105_0006	MCSHcm	Card sources.

MasterCard 1, bin6.dat

DFS Data Index	Default Value	Description
0106_0001	510000	Start of BIN range.
0106_0002	559999	End of BIN range.
0106_0003	14	Minimum account length.
0106_0004	16	Maximum account length.

0106_0005	`000110`
	`10200000000000000000000000000000`
	`11210000000000000000000000000000`
	`12220000000000000000000000000000`
	`13230000000000000000000000000000`
		Processing flags.

0106_0006	MCSHcm	Card sources.

MasterCard 2, bin7.dat

DFS Data Index	Default Value	Description
0107_0001	222100	Start of BIN range.
0107_0002	272099	End of BIN range.
0107_0003	16	Minimum account length.
0107_0004	19	Maximum account length.

0107_0005	`000110`
	`10200000000000000000000000000000`
	`11210000000000000000000000000000`
	`12220000000000000000000000000000`
	`13230000000000000000000000000000`
		Processing flags.

0107_0006	MCSHcm	Card sources.

VISA, bin8.dat

DFS Data Index	Default Value	Description
0108_0001	400000	Start of BIN range.
0108_0002	499999	End of BIN range.
0108_0003	13	Minimum account length.
0108_0004	16	Maximum account length.

DFS Data Index	Default Value	Description
0108_0005	000110 10200000000000000000000000000000 11210000000000000000000000000000 12220000000000000000000000000000 13230000000000000000000000000000	Processing flags.
0108_0006	MCSHcm	Card sources.

AMEX - Range 1, bin9.dat

DFS Data Index	Default Value	Description
0109_0001	340000	Start of BIN range.
0109_0002	349999	End of BIN range.
0109_0003	14	Minimum account length.
0109_0004	15	Maximum account length.
0109_0005	000110 00200000000000000000000000000000 00210000000000000000000000000000 00220000000000000000000000000000 00230000000000000000000000000000	Processing flags.
0109_0006	MCSHcm	Card sources.

AMEX - Range 2, bin10.dat

DFS Data Index	Default Value	Description
0110_0001	370000	Start of BIN range.
0110_0002	379999	End of BIN range.
0110_0003	14	Minimum account length.
0110_0004	15	Maximum account length.
0110_0005	000110 00200000000000000000000000000000 00210000000000000000000000000000 00220000000000000000000000000000 00230000000000000000000000000000	Processing flags.
0110_0006	MCSHcm	Card sources.

Loyalty Cards, bin11.dat

DFS Data Index	Default Value	Description
0111_0001	700100	Start of BIN range.
0111_0002	700199	End of BIN range.
0111_0003	10	Minimum account length.
0111_0004	20	Maximum account length.

0111_0005	`000110`
	`00000000000000000000000000000000`
	`00000000000000000000000000000000`
	`00000000000000000000000000000000`
	`00000000000000000000000000000000`
		Processing flags.

0111_0006	MCSHcm	Card sources.

PayPal Physical Cards, bin12.dat

DFS Data Index	Default Value	Description
0112_0001	622119	Start of BIN range. Due to the value of parameter 0040_0009 (PayPal/Discover BIN table number), this BIN table number is skipped when doing a BIN lookup with PayPal disabled (parameter 0040_0006 is set to 0). Do not change this BIN table entry to replace it with a different card and assume that it will work simply because PayPal is disabled. Do not reassign bin1 or bin12 to non-payment cards.
0112_0002	622119	End of BIN range.
0112_0003	16	Minimum account length.

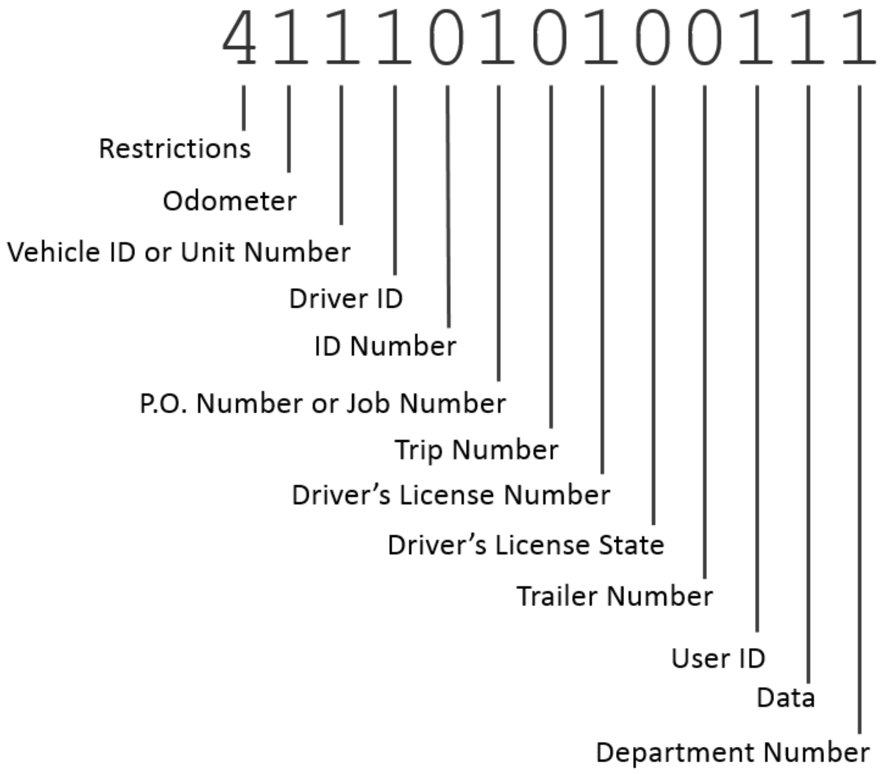
DFS Data Index	Default Value	Description
0112_0004	16	Maximum account length.
0112_0005	G00110 000000000000700000000000000 00000 000000000000710000000000000 00000 000000000000720000000000000 00000 000000000000730000000000000 00000	Processing flags.
0112_0006	MCSHcm	Card sources.

All Other Cards, Customer Definable, bin13.dat

DFS Data Index	Default Value	Description
0113_0001	000000	Start of BIN range.
0113_0002	999999	End of BIN range. There are up to 30 BIN ranges. All BIN ranges can be configured by the customer.
0113_0003	10	Minimum account length.
0113_0004	20	Maximum account length.
0113_0005	000110 102030405060708090A0B0C0D0E0F0 G0 112131415161718191A1B1C1D1E1F1 G1 122232425262728292A2B2C2D2E2F2 G2 132333435363738393A3B3C3D3E3F3 G3	Processing flags.
0113_0006	MCSHcm	Card source.

5.2.4.5 Fleet Prompting

A flag in the BIN Table indicates whether fleet processing is required and, if so, the decoding to use. The results from the fleet card read are stored to variable 430. The contents of the variable appear as a numerical string, as illustrated in the following example. The first flag shows restrictions on the card, while the rest determine whether a specific prompt is required (1) or not (0).



Allowed values for restrictions are:

- 0 = None
- 1 = Fuel only
- 2 = Fuel and maintenance only
- 3 = Fuel and other
- 4 = Fuel and auto
- 5 = Fuel and oil

In this example, the card is restricted to fuel and auto and requires every fleet prompt **except** ID Number, Trip Number, Driver's License State, and Trailer Number.

5.2.5 Card Configuration (cards.dat)

5.2.5.1 Overview

The Card Configuration parameters are used to configure the Retail Base Application (RBA) and to control the data flow, individually per card type. These parameters are found in the `config.dfs` file under the heading Cards and filename `cards.dat`. The RBA currently supports 16 card types which are referenced as type A through type P. The card options are executed by the RBA as listed in columns, starting from the left side and going to the right. This order may be altered by some of the configuration parameters listed in different configuration files such as those found in `mainFlow.dat`. To summarized, the application uses the following:

- Parameter Name - Card Configuration for Cards A through P
- DFS Data Index - 0011_0001 through 0011_0016

For default values refer to Card Configuration Table.

5.2.5.2 Use of the Verify Amount Flag During EMV Transactions

EMV transactions use the Amount Verify flag in the `cards.dat` section of the `config.dfs` file, which provides a means of suppressing the amount verification prompt during the EMV transaction flow. In situations where the customer wants to prompt the cardholder for amount verification, this eliminates a duplicate prompt. Additionally, the transaction amount can be displayed on the PIN Entry or Signature screen. When the cardholder enters a PIN or signs, approval of the amount is implied. If there is no card verification method, then there is no screen for implied approval in such cases. The Verify Amount flag works as follows:

- 0 = Do not verify amount
- 1 = Always verify amount
- 2 = Verify amount if cashback

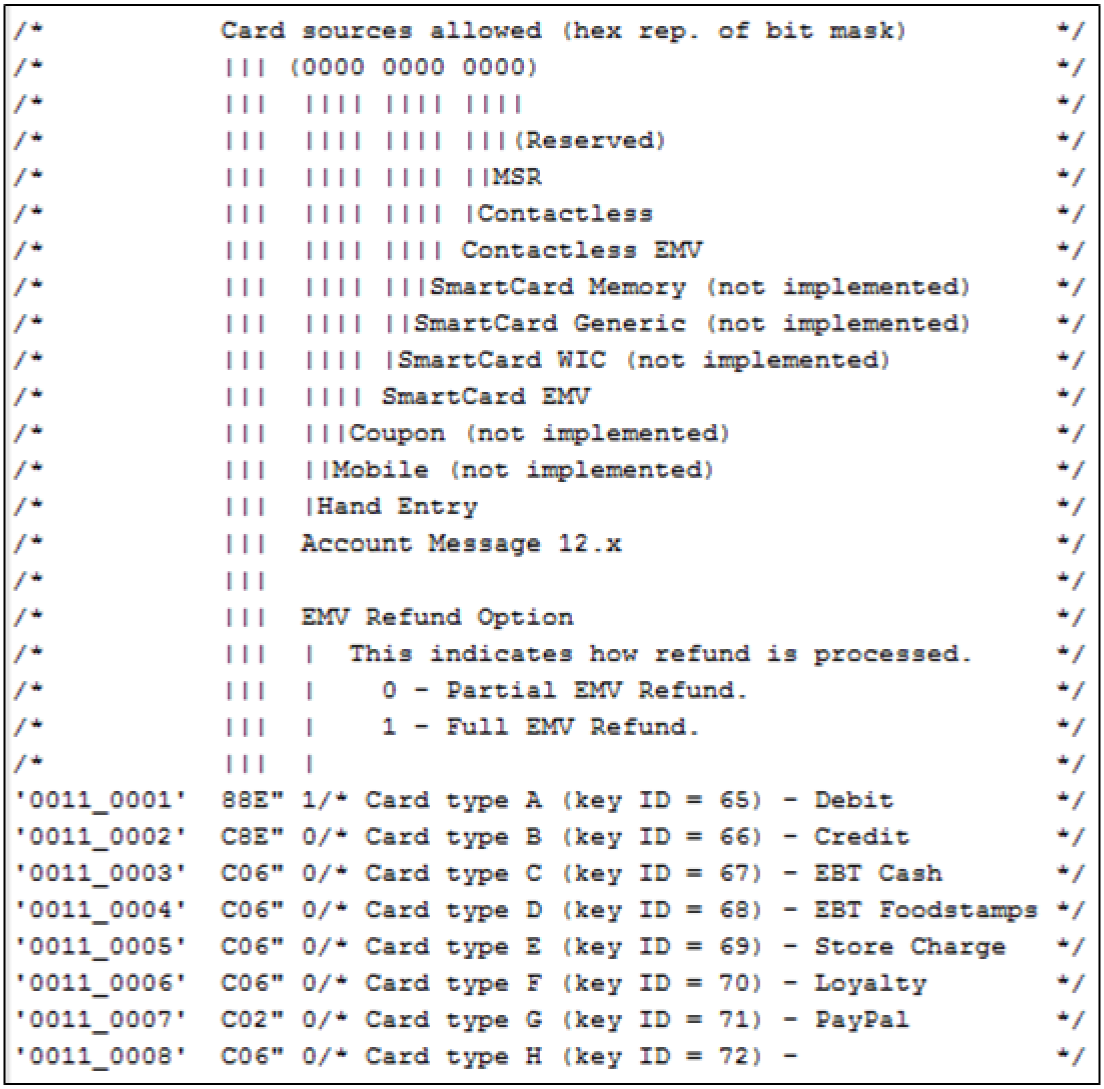
If the merchant elects to use their own prompt for amount verification, the Amount OK screen is suppressed by setting the Verify Amount flag to 0.

5.2.5.3 Card Configuration and Options

Each record specifies the processing for a different card type. Most card types are defined as payment cards. These are configured to support debit, credit, and other payment type transactions. Some cards may be configured as non-payment type cards, such as loyalty or ID cards; these cards are not part of the payment transaction.

Each card type has a key ID. When a payment menu form is created and you would like to display a button to select this payment type, the ID that the button returns should be the key ID value. As an example, the key ID ASCII value for a Debit card is 65. Refer to the following extract from the config.dfs file for card configuration for more card configuration parameters and key IDs.

> All of the entries in `cards.dat` are configurable.



```
/*           Card sources allowed (hex rep. of bit mask)        */
/*           ||| (0000 0000 0000)                               */
/*           |||  |||| |||| ||||                                */
/*           |||  |||| |||| |||(Reserved)                       */
/*           |||  |||| |||| ||MSR                               */
/*           |||  |||| |||| |Contactless                        */
/*           |||  |||| |||| Contactless EMV                     */
/*           |||  |||| |||SmartCard Memory (not implemented)    */
/*           |||  |||| ||SmartCard Generic (not implemented)    */
/*           |||  |||| |SmartCard WIC (not implemented)         */
/*           |||  |||| SmartCard EMV                            */
/*           |||  |||Coupon (not implemented)                   */
/*           |||  ||Mobile (not implemented)                    */
/*           |||  |Hand Entry                                   */
/*           |||  Account Message 12.x                          */
/*           |||                                                */
/*           |||  EMV Refund Option                             */
/*           |||  |  This indicates how refund is processed.    */
/*           |||  |    0 - Partial EMV Refund.                  */
/*           |||  |    1 - Full EMV Refund.                     */
/*           |||  |                                             */
'0011_0001'  88E" 1/* Card type A (key ID = 65) - Debit          */
'0011_0002'  C8E" 0/* Card type B (key ID = 66) - Credit         */
'0011_0003'  C06" 0/* Card type C (key ID = 67) - EBT Cash       */
'0011_0004'  C06" 0/* Card type D (key ID = 68) - EBT Foodstamps */
'0011_0005'  C06" 0/* Card type E (key ID = 69) - Store Charge   */
'0011_0006'  C06" 0/* Card type F (key ID = 70) - Loyalty        */
'0011_0007'  C02" 0/* Card type G (key ID = 71) - PayPal         */
'0011_0008'  C06" 0/* Card type H (key ID = 72) -                */
```

Extract of Card Configuration in config.dfs File

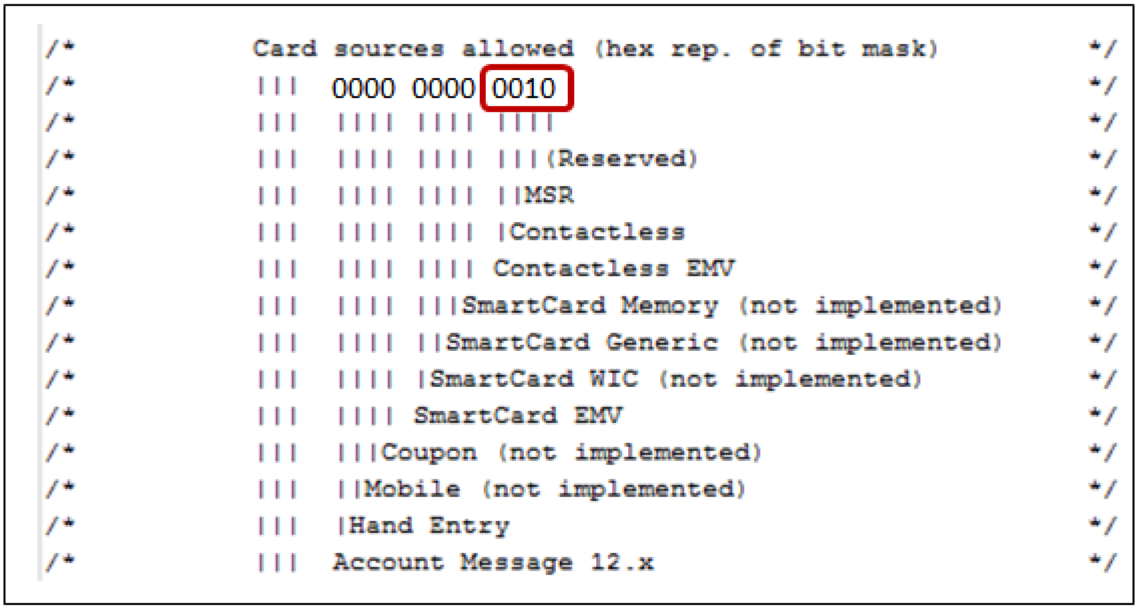
Card configuration parameters 0011_0009 (key ID 73) through 0011_0016 (key ID 80) are customer definable.

5.2.5.4 Converting Binary to Hexadecimal for Card Sources Allowed

Any combination of the allowable card sources for assignment to particular card types can be configured in the `config.dfs` file. This section describes how to convert the 12 binary digits that make up the various card sources allowed to three hexadecimal digits for use in the `config.dfs` file.

Binary-to-Hex Conversions Example 1: MSR Only

In this example, only MSR is to be configured. The binary representation for Card Sources Allowed will be 0000 0000 0010 in this case. A 1 in any position indicates Allowed, while a 0 indicates Not Allowed. Since hexadecimal is base-16 and binary is base-2, one set of four binary digits converts to one hexadecimal digit. So three four-digit binary sets converts to a three-digit hexadecimal number. Lets look at the following example:



```
/*          Card sources allowed (hex rep. of bit mask)          */
/*          |||  0000 0000 0010                                   */
/*          |||  |||| |||| ||||                                   */
/*          |||  |||| |||| |||(Reserved)                          */
/*          |||  |||| |||| ||MSR                                  */
/*          |||  |||| |||| |Contactless                           */
/*          |||  |||| |||| Contactless EMV                        */
/*          |||  |||| |||SmartCard Memory (not implemented)       */
/*          |||  |||| ||SmartCard Generic (not implemented)       */
/*          |||  |||| |SmartCard WIC (not implemented)            */
/*          |||  |||| SmartCard EMV                               */
/*          |||  |||Coupon (not implemented)                      */
/*          |||  ||Mobile (not implemented)                       */
/*          |||  |Hand Entry                                      */
/*          |||  Account Message 12.x                             */
```

Binary-to-Hex Conversions Example 1: MSR Only

In this example, the lower four bits are 0010. Referring to the following conversion table, the hexadecimal value is 2. With all zeros for the upper eight bits, the hexadecimal value for Card Sources Allowed is 002.

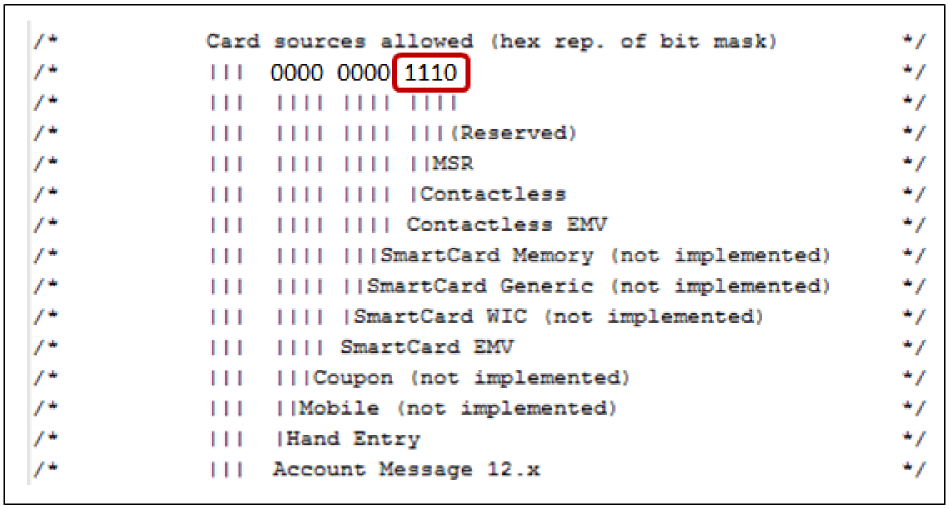
Key: Table of Hex-to-Binary Equivalents

Hex	Binary Equivalent
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011

Hex	Binary Equivalent
C	1100
D	1101
E	1110
F	1111

Binary-to-Hex Conversions Example 2: MSR, Contactless, and Contactless EMV

In this next example we allow MSR, Contactless, and Contactless EMV. Set the bits accordingly as shown:



```
/*          Card sources allowed (hex rep. of bit mask)          */
/*          |||  0000 0000 1110                                  */
/*          |||  |||| |||| ||||                                  */
/*          |||  |||| |||| |||(Reserved)                         */
/*          |||  |||| |||| ||MSR                                 */
/*          |||  |||| |||| |Contactless                          */
/*          |||  |||| |||| Contactless EMV                       */
/*          |||  |||| |||SmartCard Memory (not implemented)      */
/*          |||  |||| ||SmartCard Generic (not implemented)      */
/*          |||  |||| |SmartCard WIC (not implemented)           */
/*          |||  |||| SmartCard EMV                              */
/*          |||  |||Coupon (not implemented)                     */
/*          |||  ||Mobile (not implemented)                      */
/*          |||  |Hand Entry                                     */
/*          |||  Account Message 12.x                            */
```

Binary-to-Hex Conversions Example 2: MSR, Contactless, and Contactless EMV

In this example, the hexadecimal value for the lower four bits is E, using the above conversion table with a binary value of 1110. With all zeros for the upper eight bits, the hexadecimal value for Card Sources Allowed is 00E.

5.2.5.5 Card Configuration Table

The columns in the card configuration table are:

Card Configuration Table

Parameter	Position from Left	Description
Enable	1	Specifies whether this particular card type (e.g., A for Debit) is allowed (enabled). • 0 = Enabled. • 1 = Disabled.
Card Type	2	Specifies if the account number from the swiped card is used for tendering (payment for purchase) or for information (discounts, loyalty). • 0 = Payment card. • 1 = Non-payment card (loyalty card, rewards card, points card, advantage card, or club card type).
Required Track	3	Specifies which card MSR track must be available for this type of card. • 1 = Track 1 required. • 2 = Track 2 required. • 3 = Use Track 1 if read, else use Track 2. • 4 = Use Track 2 if read, else use Track 1. • 5 = Require both tracks (Both tracks will be in the 50.x Authorization Request message).
Display Show Card to Cashier Timeout	4	Controls the display of the prompt, "Show card to cashier." • 0 = Do not show. • Other than 0 = Time to Display (in 1/10th of a second).
PIN Prompt	5	PIN prompt. • 0 = the RBA does not prompt the customer to enter a PIN. • >0 = parameter is treated as a prompt ID for the `SECURPROMPT.xml` file. The string pointed to by the index is displayed during PIN entry.

Parameter	Position from Left	Description
Cash Back Limit	6	Cash back limit in cents (e.g., enter 10000 for $100.00). The limit also serves to enable cash back entry. • 0 = Cash back entry not allowed.
Verify Cash Back	7	When Cash Back Limit is a positive number, the Verify Cash Back value indicates whether to display a prompt to confirm the cash back selection. • 0 = Don't Verify. • 1 = Verify.
Amount Index	8	This is an index to the purchase amount field in the 13.x Amount Message. When the 13.x message has multiple fields, the amount that the index points to is used in the 50.x request. • Index 1 points to the first amount field in the 13.x message • Index 2 points to the second field, and so on. This allows you to specify the appropriate field for each card type.
Verify Amount	9	Indicates whether to display a message to confirm the purchase amount. • 0 = Do not verify. • 1 = Always verify. • 2 =Verify if Cash Back Limit is greater than 0.
Signature Capture	10	Indicates whether a signature is required on credit transactions. • 0 = No signature. • 1 = Signature required after transaction is approved. • 2= Signature required before approval.
Signature Threshold	11	Sets the minimum transaction value for which a signature is required. • The Signature Capture parameter value must be either 1 or 2 for this parameter to be valid.

Parameter	Position from Left	Description
Index # for Card Description	12	Represents the `PROMPT.xml` prompt ID. Text from the corresponding prompt ID is displayed for two seconds on the terminal screen to show the selected payment type.
Check Expiration Date	13	Compare the expiration date on the card with the date set in the terminal. If the card is expired, display an error and ask for a new card. The time and date of the terminal must be set properly using the 28.x Set Variable Request message.
On PIN Entry Cancel	14	This tells the RBA what should happen when the cancel button is pressed on the current transaction's PIN entry screen. • "-" = Restart PIN entry. • 0 = Cancel the payment and start over. • 1 - 9 = Loads the payment menu 1 - 9 respectively. • A - P selects the specific payment type.
Allow Partial Payment Buttons on the Amount Verification Screen	15	If the amount verification form has a partial payment button on it, • setting this entry to 0 removes the button for this payment type. • setting this entry to 1 allows the button.
Encryption Index	16	Encryption index. • D = use the index specified by parameter '0006_0008'. • 0 - 9 = use the specified index (e.g., 0, 1, 2, ..., or 9).
Prompt for Expiration Date for Manual Entry	17	This Boolean flag tells the RBA whether or not to prompt for manual entry of Expiration Date, and can be made applicable to a particular card type.

Parameter	Position from Left	Description
Prompt for CVV for Manual Entry	18	This Boolean flag tells the RBA whether or not to prompt for CVV, and can be made applicable to a particular card type. Examples for Prompt for Expiration Date for Manual Entry, and Prompt for CVV for Manual Entry: A card type of 'credit' can be configured to always prompt for CVV and Expiration Date, but 'EBT Cash' can be configured to not prompt for either of those. Yet another type can be configured to prompt for one but not the other.
Card Sources Allowed (hexadecimal representation of bit mask)	19	This setting specifies which card sources are enabled for a particular card type. Examples for Card Sources Allowed: A card type of 'debit' can be configured to not allow for manual card entry, and another card type, such as 'Store Gift Card' can be configured to not allow contactless. In the card table there are placeholders for about ten (10) card sources for this field, but only manual and contactless are currently affected by this field. Other card sources (like various flavors of Smart Card) have placeholders.
EMV Refund Option	20	The setting of this parameter determines how an EMV refund is processed. • 0 = Non-EMV refund. • 1 = Full EMV refund.

5.2.6 Cash Back Configuration (cashback.dat)

This section describes the parameters used to configure the cash back options. These parameters are found in the `config.dfs` file under the heading, Cash Back, and filename `cashback.dat`.

If selected in the configuration, each cash selection must be confirmed or rejected by the cardholder.

When the cash back screen is allowed, the cardholder is prompted with the cashback option.

- If the cardholder presses NO, the RBA skips over the Cash Back process and continues to the next process.
- If the cardholder presses YES, the cash selection screen displays.

The cardholder may select fast cash, such as $20 or $40, or may tap the OTHER button and enter a specific amount. Entered cash amount is checked against the lowest maximum cash back limit. That is, the global maximum cash back limit (Cash Back section of `config.dfs`, '0002_0001') or the per-card maximum cash back limit (Cash Back Limit column in Card Configuration Table). As a note, the cash back button will not be displayed on Amount Verification screen of the iUN (unattended terminal) since this terminal is limited to only two buttons.

The option of entering the cash back amount is controlled individually for each card type. The choices are listed in the cards.dat file, under a parameter called Cash Back Limit.

- If the value of Cash Back Limit is 0, the cash back screen is disabled.
- If the Cash Back Limit is greater than 0, the cash back screen is displayed by the RBA data flow. When the Cash Back Limit value is greater than 0, that value has two meanings: it enables the cash back screen to be displayed, and it limits the amount of cash that the cardholder can request.

Other functions of the Cash Back entry are controlled by the configuration switches common to all card types, which are listed in the Cash Back section of `config.dfs`.

cashback.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
Cash Back Limit	0002_0001	99999	This record specifies a global cash back limit for all payment types in cents. Maximum value is 99999 ($999.99).
Initial Cash Back Screen	0002_0002	0	This parameter determines whether the user will see buttons showing predefined cash back amounts and a button labeled Other for manual entry. If enabled, the user must type in the desired cash back amount manually. This allows customers to enter any amount up to the cash back limit in dollars. • 0 = Use fast cash back keys. Allow an “Other” button that user can select for manual entry of cash back amount. (Default). • 1 = Manual entry only. Do not use fast cash back keys.

Parameter Name	DFS Data Index	Default Value	Description
Amount for Fast Cash Back Key 1	0002_0004	2000	Cash back amount assigned to soft key #1. Amount must be in cents (e.g., enter 2000 for $20.00).
Amount for Fast Cash Back Key 2	0002_0005	4000	Cash back amount assigned to soft key #2. Amount must be in cents (e.g., enter 4000 for $40.00).
Amount for Fast Cash Back Key 3	0002_0006	8000	Cash back amount assigned to soft key #3. Amount must be in cents (e.g., enter 8000 for $80.00).
Amount for Fast Cash Back Key 4	0002_0007	10000	Cash back amount assigned to soft key #4. Amount must be in cents (e.g., enter 10000 for $100.00).
Use Cash Back	0002_0008	0	The Cash Back Increments parameter can only be used if the Cash Back Selection parameter ('0002_0002') is set to 1. Cash back increments are only used after the cardholder has selected the cash back option OTHER. • 0 = Disable (default) • 1 = Enable. The amount entered must be a multiple of the amount specified in the Cash Back Increment Amount parameter ('0002_0009').
Cash Back Increment Amount	0002_0009	1000	This parameter can only be used if the Use Cash Back Increments parameter is enabled. This parameter specifies the increment amount for cash back in cents. If this parameter is enabled, customers will only be allowed to receive cash back using the increments set by this parameter. For example, if the increment is $20, customers will not be able to receive $30 cash back.

Parameter Name	DFS Data Index	Default Value	Description
Cash Back Flow	0002_0010	1	This parameter specifies when cash back will be prompted for: • 0 = Before PIN entry • 1 = After PIN entry (default) • 2 = Cash back option is offered with Amount Verification screen. The cash back button is not available on the Amount Verification screen for the iUN since this terminal is limited to only two buttons. When selecting option '2'; following PIN entry, if there is no amount then the terminal will display the "Please wait for cashier" screen, not the cash back screen. Once there is an amount, the amount verification screen with the cash back option will then be displayed.
Cancel Destination	0002_0011	0	This parameter specifies where to go if the cancel button is pressed: • 0 = Restart Transaction • 1 = Payment Menu • 2 = Cashback Screen
On Cashback Incorrect	0002_0012	1	This parameter specifies what to do when the customer says that the cashback amount is incorrect: • 0 = Return to cashback amount screen • 1 = Return to cashback Yes/No screen • 2 = Restart transaction • 3 = Return to payment selection screen

Parameter Name	DFS Data Index	Default Value	Description
Cashback Other Amount	0002_0013	0	This parameter determines if the amount entered is in dollars or cents: • 0 = Dollars and cents • 1 = Dollars

5.2.7 Compatibility Flags (compat.dat)

This section is for customers who coded their POS system to work with features of legacy terminals to select the old or new functionality.

These parameters are found in the `config.dfs` file under the heading, Compatibility Flags, and filename `compat.dat`.

compat.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
11.x Status Response Format	0013_0001	1	Specifies the format of the status message response. • 0 = <STX><11.><number 2 bytes> <text, up to 32 char><ETX> • 1 = <STX><11.><number 2 bytes> <text, up to 32 char><FS> <ETX>
11.x Status Response Format	0013_0002	0	When 11.x is received in the offline mode, respond with: • 0 = <STX><11.><00><text><ETX> (response with status) • 1 = <STX><00.><offline code><ETX> (response with off-line message)

Parameter Name	DFS Data Index	Default Value	Description
Treat a Message as an ACK	0013_0003	0	Assume an ACK was received if a message is received in response to a sent message: • 0 = Wait for an ACK • 1 = Treat a received message as an ACK
Display "Cancelled Transaction" Screen	0013_0004	0	Display transaction cancelled screen: • 0 = Do not display • 1 – 255 = Display time in 1/10th of a second
Numeric Payment Types	0013_0005	0	Responses 1 - 8 in 04.x and 19.x select: • 0 = Payment types A – H • 1 = Payment menus `pay1.icg` - `pay8.icq`
Line Item Compression	0013_0006	1	Use smart compression on the line display: • 0 = Truncate text if the data is too long for the display • 1 = Use smart compression to make text fit on the line display
Send Reset at End of a Transaction	0013_0007	0	Determines when to send a reset message at the end of a transaction. Valid if the parameter, 0007_0023, is set to *reset at the end of a transaction.* • 0 = Do not send • 1 = Send on approved • 2 = Send on decline • 3 = Always send

Parameter Name	DFS Data Index	Default Value	Description
Reset on Decline	0013_0008	1	Sets the application to reset after a decline transaction: • 0 = Do not reset • 1 = Reset
Option to Add Destination Field to Reset Message	0013_0009	0	Sets whether to add a field to the reset message telling the POS the new destination in the application flow after the reset message is received: • 0 = Do not add field • 1 = Add field
Delay ACK until Message is Processed	0013_0011	0	Sends ACKs to messages as soon as a message is received or delay until the message is processed. • 0 = Normal • 1 = Delay ACK
Suppress Response to 28.x Set Variable Message	0013_0012	0	This setting turns off the response message sent to the POS when a Set Variable message is received. • 0 = Send Response • 1 = Suppress
Add Source Field to 23.x: Card Read Request Message	0013_0014	0	Include the source field to the 23.x Card Read Request (On-Demand) message. • 0 = Do not include source (compatible with previous versions) • 1 = Add source of card data to message
Go Online After a Successful EFT Download	0013_0015	1	Controls whether to go online after a successful EFT download. • 0 = Start in offline mode • 1 = Start in online mode

Parameter Name	DFS Data Index	Default Value	Description
Field Size for RAM and Flash Memory Size	0013_0016	1	This setting sets the field size for the terminals RAM and Flash memory. • 0 = 4-byte fields • 1 = Variable size fields The default value of 1 allows the 07.x Unit Data Request message response to return the correct amount of RAM and flash when the resulting value exceeds 9999 kilobytes.
Send PIN Entry Message when Cancel Transaction	0013_0017	0	Send 31.1 when PIN entry is cancelled during the transaction. • 0 = Disable • 1 = Enable
Use alternate maximum Tailgate packet size	0013_0018	1	Use alternate maximum Tailgate packet size. • 0 = Disable (247 byte max) • 1 = Enable (240 byte max)
Reject incoming connection requests if connected	0013_0019	0	Reject incoming connection requests if connected. • 0 = Allow connection requests from the same host. • 1 = Reject all connection requests while connected.
Host communication inactivity timeout (Ethernet)	0013_0020	0	Host communication inactivity timeout (Ethernet only). • 0 = Disable timeout • 1 - 9 = Invalid. Changed to 10 seconds • > 10 = Inactivity timeout in seconds

Parameter Name	DFS Data Index	Default Value	Description
12.x Account Message Response	0013_0021	1	12.x Account Message Response. • 0 = Do not send 12.x response. • 1 = Send 12.x response.
Cancel Key Handling	0013_0022	0	Configures the default behavior for the <Cancel> key. • 0 = Functions as input entry Cancel • 1 = Functions as input entry Clear If no digits/characters were entered, the <Cancel> key cancels the clear-text or PIN entry. If at least one digit/character was entered, the <Cancel> key clears the entered digits/characters and restarts clear-text or PIN entry. This is applicable to non-iUN terminals only.
Decommission an unattended terminal	0013_0024		Decommission an unattended terminal. Values include: • 0 = The terminal exits RBA if it is not commissioned • 1 = The 07.x Unit Data and 08.x Health Stats messages return *Decommissioned* for the serial number. RBA responds with *00.9999* to all other messages. If the terminal is started in maintenance mode and not commissioned, it exits RBA.

5.2.8 Contactless Reader Configuration (cless.dat)

This section describes the configuration parameters for the internal contactless card reader.

When contactless is enabled using the 412 variable with the 28.x Set Variable Request message, contactless mode is enabled only until the terminal is rebooted. It is disabled following a reboot of the terminal. To enable contactless mode permanently (following terminal reboot), use the 0008_0001 configuration parameter with the 60.x Configuration Write message. The 0008_0001 configuration parameter defines whether the contactless card reader is enabled and the supported mode, such as key card or EMV.

Note: To retain the value following reboot, a 00.x Offline Message or 01.x Online Message must be sent after the configuration is changed via the 60.x message.

Refer to the following table for contactless configuration parameters.

Contactless Configuration Parameters

Parameter Name	DFS Data Index	Default Value	Description
Contactless Reader Mode	0008_0001	9	Defines whether the contactless card reader in the terminal is enabled and which supported mode is selected. • 0 = Disabled • 1 = Payment (PPSE) only • 8 = Key Card only • 9 = EMV only. A check for Amount is performed, and if: a. No amount is set, the 23.x response is 23.A, indicating that Amount was not set, and the contactless reader is not enabled b. An amount is set, no response is sent until the card read request is complete After the contactless mode setting is changed, the terminal resets the transaction, and the screen is refreshed.
Beep When Contactless Card is Read	0008_0003	1	Whether to sound a tone when a valid contactless card read is received: • 0 = Do not sound a tone • 1 = Sound a tone To completely disable a beep at card read, 0007_0014 must also be set to 0.

Parameter Name	DFS Data Index	Default Value	Description
Contactless Both Tracks Indicator	0008_0005	b	Defines the value used in the **Account Data Source** Field in the **Authorization Request** message. This parameter applies when the account information source is both tracks of a card read by the Contactless reader.
Contactless Track 1 Indicator	0008_0006	h	Defines the value used in the **Account Data Source** Field in the **Authorization Request** message. This parameter applies when the account information source is Track 1 of a card read by the Contactless reader.
Contactless Track 2 Indicator	0008_0007	d	Defines the value used in the **Account Data Source** Field in the **Authorization Request** message. This parameter applies when the account information source is Track 2 of a card read by the Contactless reader.
Bad Read Error Display	0008_0008	30	Defines the display duration for a Bad read error. • 0 = Disabled. • >0 = Display duration in 1/10 seconds.
Contactless Event Delay	0008_0010	7	Specifies the amount of time that the terminal must detect a contactless card before it registers the event. This is used to keep the terminal from logging contactless events when swiping a contactless card through the MSR. • 0 = no delay • 1 - 65000 = time in 1/10 seconds
Contactless Floor Limit	0008_0011	200	Defines the contactless floor limit.
Contactless Suspend Steps	0008_0012	(blank)	Contactless suspend steps: • c = Contactless online PIN entry • d = Contactless message display • e = Contactless re-tap required • f = Contactless get data before transaction result

Parameter Name	DFS Data Index	Default Value	Description
Contactless Status Update Steps	0008_0013	(blank)	Contactless status update steps: • a = Card tapped • b = Contactless collision detected • c = Contactless online PIN entry • d = Contactless message display • e = Contactless re-tap required • f = Contactless get data before transaction result
Contactless Suspend Timer	0008_0014	0	Contactless Suspend Timer. • 0 = Timer disabled • >0 = Suspend time in 1/10 seconds.
PayPass Kernel Version	0008_0015	2	PayPass Kernel Version. • 2 = PayPass 2 • 3 = PayPass 3
ExpressPay Kernel Version	0008_0016	3	ExpressPay Kernel Version. • 2 = ExpressPay 2 • 3 = ExpressPay 3
Contactless Card Tap Timeout	0008_0017	60	Configurable timeout for contactless card tap. The set timeout is applied each time the contactless card reader is enabled. • Timeout in seconds.
Contactless transaction type 1	0008_0018	22	Debit contactless void sale transaction type.
Contactless transaction type 2	0008_0019	02	Debit contactless void return transaction type.
Contactless transaction type 3	0008_0020	00	Credit contactless void sale transaction type.
Contactless transaction type 4	0008_0021	00	Credit contactless void return transaction type.

Parameter Name	DFS Data Index	Default Value	Description
Enable Magnetic Stripe-Only support	0008_0022	0	Enable limiting contactless transactions to magnetic stripe (MSD) only. This allows for supporting contactless transactions without changing terminal capabilities in XML configuration. Recommended for use with 0008_0001 = 9, and can be used in lieu of 0008_0001 = 1. • 0 = Disabled. EMV and magnetic stripe contactless transactions both allowed • 1 = Enabled. Magnetic stripe-only contactless transactions allowed, EMV contactless disallowed Requires a reboot when resetting to 0 to take effect.

5.2.9 EMV Flags (emv.dat)

This section describes the flags in the `emv.dat` file used for EMV transactions.

emv.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
EMV Transactions Supported	0019_0001	0	Determines whether the POS supports EMV transactions: • 0 = The POS does not support EMV transactions • 1 = The POS supports EMV transactions

Parameter Name	DFS Data Index	Default Value	Description
Language Auto-Select During EMV Transaction	0019_0002	1	Determines whether the terminal auto-selects the language during an EMV transaction: • 0 = the terminal offers language selection • 1 = the terminal auto-selects the language during an EMV transaction. The language selected is either: ◦ The language specified as the card's preferred language and supported by the The terminal, if there is a match ◦ The default language (RBA variable 409)

Parameter Name	DFS Data Index	Default Value	Description
Application Auto-Select during EMV Transaction	0019_0003	0	Determines whether the terminal auto-selects the application during an EMV transaction: • 0 = The terminal prompts cardholder to select application via menu during an EMV transaction • 1 = The terminal prompts cardholder to select application via confirmation during an EMV transaction starting with the highest-priority application(s) • 2 = The terminal prompts cardholder to select application via menu during an EMV transaction, but does not prompt to confirm the selection • 3 = The terminal prompts cardholder to select application via confirmation during an EMV transaction starting with the highest-priority application(s) but auto-selects the lowest-priority application without subsequent cardholder confirmation if all other applications are declined

Parameter Name	DFS Data Index	Default Value	Description
Asynchronous Transmission of Status Message	0019_0004	0	Determines whether the terminal should asynchronously transmit the Status message to the POS on change of status: • 0 = The terminal does not asynchronously transmit the Status message to the POS on change of status • 1 = The terminal asynchronously transmits the Status message to the POS on change of status
Load EMV Test Environment	0019_0005	1	Determines whether the terminal loads the EMV test and production environments: • 0 = Load the production environment only • 1 = Load the test environment and production environment
Wait After Authorization Confirmation Sent to POS	0019_0006	0	Determines whether the terminal should wait after the Authorization Confirmation message has been sent to the POS. Typically, the terminal pauses for a signature request if it has not already been received after the confirmation message: • 0 = No wait after the confirmation message is assumed in EMV transactions • 1 = Wait after the confirmation message is assumed in the EMV transaction
Interac Application Selection	0019_0007	1	Canada only. • 0 = Disabled • 1 = Enabled

Parameter Name	DFS Data Index	Default Value	Description
Domestic VISA Debit Application Selection	0019_0008	1	Canada only. • 0 = Disabled • 1 = Enabled
Allow CVM modification by POS	0019_0009	0	Allow CVM modification by the POS: • 0 = Do not allow CVM modification by POS (QPS, EPS, etc.) • 1 = Allow CVM modification by the POS
Configurgation file for contact EMV to load at boot time	0019_0010	EMVCLESS.XML	This parameter can be overridden by the EMV 33.08.x Set Variables Message. The name and path of the last file loaded can be retrieved by the `600` variable. If left blank, `EMVCONTACT.XML` is loaded. The source folder for this file is determined by 0091_0031.
Configuration file for contactless EMV to load at boot time	0019_0011	EMVCLESS.XML	This parameter can be overridden by the EMV 33.08.x message. The name and path of the last file loaded can be retrieved by the `601` variable. If left blank, `EMVCLESS.XML` is loaded. The source folder for this file is determined by 0091_0031.
EMV Transaction Response Message	0019_0012	1	Indicates whether the EMV transaction response message is enabled: • 0 = EMV transaction response message is disabled • 1 = EMV transaction response message is enabled
Authorization Response Timeout	0019_0013	700	Timeout for EMV 33.04.x Authorization Response Message: • 0 = Disabled • >0 = Time in tenth seconds

Parameter Name	DFS Data Index	Default Value	Description
MSR Fallback on Power On Failure	0019_0014	0	Indicates whether the application should check for a power-on failure condition for possible fallback to MSR for unattended terminals with combo readers. • 0 = Disabled • 1 = Enabled
No Candidate Match MSR Fallback	0019_0015	0	Indicates whether the application should check for no matching application ID condition for possible fallback to MSR for unattended for terminals with combo readers: • 0 = Disabled • 1 = Enabled
Debit Contact Void Sale Transaction Type	0019_0016	22	Debit contact void sale transaction type
Debit Contact Void Return Transaction Type	0019_0017	2	Debit contact void return transaction type
Credit Contact Void Sale Transaction Type	0019_0018	0	Credit contact void sale transaction type
Credit Contact Void Return Transaction Type	0019_0019	0	Credit contact void return transaction type
Enable External AID selection	0019_0020	0	Enable or disable EMV 33.11.x External AID Selection Notification messages from being sent: • 0 = Disabled • 1 = Enable
FastQuickChipEnable	0019_0021	0	Enable Fast Quick-Chip support: • 0 = Disable • 1 = Enable To enable Fast Quick-Chip for an AID, you must also set the Fast Quick-Chip bit in emvaid.dat. Not all card brands support Fast Quick-Chip.

Parameter Name	DFS Data Index	Default Value	Description
FastQuickChipSuspendSteps	0019_0022	(blank)	Refer to the Transaction Step List for suspend steps.
FastQuickChipStatusUpdateSteps	0019_0023	(blank)	Refer to the Transaction Step List for status update steps.

5.2.10 EMV AID Parameters (emvaid.dat)

This section describes the parameters in the `emvaid.dat` file used for EMV AID selection.

The `emvaid.dat` file can set options for individual AIDs, as shown in the following example. When processing a card with an AID that is not listed in the table, default values are used.

```
'0021_0001' "A0000000031010 55 0 02 0 0 1 0 0 0 1 0 0 0 0" /* VSDC */
'0021_0002' "A0000000041010 55 0 01 0 0 0 0 0 0 0 0 0 0 0" /* MasterCard */
'0021_0003' "A0000002771010 55 0 03 0 1 0 0 0 1 0 0 0 0 0" /* Interac */
'0021_0004' "A0000000032010 55 0 02 0 0 0 0 0 0 0 0 0 0 0" /* VISA */
'0021_0005' "A00000002501   55 0 04 0 0 0 0 0 0 0 0 0 0 0" /* AMEX */
'0021_0006' "A0000001523010 55 0 05 0 0 0 0 0 1 1 1 0 0 0" /* Discover */
'0021_0007' "A0000000980840 55 0 02 1 0 0 0 0 0 1 0 0 0 0" /* VISA US Common Debit */
'0021_0008' "A0000000042203 55 0 01 1 0 0 0 0 1 1 0 0 0 0" /* MasterCard US Common Debit */
'0021_0009' "A0000001524010 55 0 05 1 0 0 0 0 1 1 1 0 0 0" /* Discover US Common Debit */
'0021_0010' "A0000006200620 55 0 06 1 0 0 0 0 0 1 0 0 0 0" /* DNA US Common Debit */
'0021_0011' "A0000003330101 55 1 07 0 0 0 0 0 0 0 0 0 0 0" /* UnionPay */
'0021_0012' "A0000000651010 55 0 08 0 0 0 0 0 0 0 0 0 0 0" /* JCB */
'0021_0013' " " /* */
'0021_0014' " " /* */
'0021_0015' " " /* */
'0021_0016' " " /* */
```

`emvaid` Parameters

The parameters in the `emvaid` table entries are described below:

Card Configuration Table

Parameter	Position from Left	Description
AID	1	The Application ID being configured. Specifies whether this particular card type (such as A for Debit) is allowed (enabled). • 0 = Enabled • 1 = Disabled
Tag 8A value for bad PIN	2	Deprecated/Not supported. RBA does not use tag 8A to identify an *invalid online PIN* response from the processor. The POS should interpret tag 8A as defined by the processor. If the value indicates a bad online PIN, the POS can instruct RBA how to proceed using tag D1011.
Allow PIN bypass	3	The PIN Bypass flag allows or disallows cardholders to end EMV PIN entry without entering digits. The transaction proceeds without a PIN. See notes below. This flag applies to contact EMV cards: • 0 = PIN bypass not allowed (default for all AIDs) • 1 = Enter key bypasses PIN with no digits entered • 2 = Cancel key bypasses PIN with no digits entered
AID Brand	4	Identifies the card brand associated with this AID. This is an index into the `emvbrand` table; for example, the 0021_0002 parameter's AID Brand value of 01 refers to `emvbrand.dat` parameter 0022_0001.

Parameter	Position from Left	Description
US Common Debit AID	5	Values are: • 0 = AID is a non-debit or global debit AID • 1 = AID is a US Common Debit AID The merchant's preference for US Common Debit AID, Global AID, or both, can be configured in the `emvbrand` table. Only the following AIDs are considered US Common Debit AIDs: • DNA Shared Debit AID – A0 00 00 06 20 06 20 • Discover US Common Debit AID–A0 00 00 01 52 40 10 • MasterCard US Common Debit AID–A0 00 00 00 04 22 03 • Visa US Common Debit AID –A0 00 00 00 98 08 40
Force cashback	6	Values are: • 0 = Determine whether to prompt for cash back based on the card's Application Usage Control (AUC) bits • 1 = Force prompting for cash back, regardless of the AUC bits For further details on cash back, see Enabling EMV Cash Back.
PAN consistency check	7	Test whether the Primary Account Number (PAN) in tag 57 (Track 2 Equivalent Data) is consistent with tag 5A (Primary Account Number). This check applies to contactless cards only. • 0 = Disable check for consistency • 1 = Enable check for consistency. If the check fails, the transaction ends with tag D1010 set to the error code "T2CF", Track 2 Consistency Check Failed.

Parameter	Position from Left	Description
Allow contactless PIN bypass	8	The PIN Bypass flag allows or disallows cardholders to end EMV PIN entry without entering digits. The transaction proceeds without a PIN. See notes below. This flag applies to contactless EMV cards. Valid values for the flag are: • 0 = PIN bypass not allowed (default for all AIDs) • 1 = Enter key bypasses PIN with no digits entered • 2 = Cancel key bypasses PIN with no digits entered
		The following flags allow the merchant to determine whether cash back is allowed for this AID, depending on the Cardholder Verification Method (CVM), and whether PIN Bypass is used in the case of PIN CVMs. For further details on cash back, see Enabling EMV Cash Back.
PIN bypass cashback	9	• 0 = Do NOT allow cash back for PIN bypass • 1 = Allow cash back for PIN bypass
Offline PIN cashback	10	• 0 = Do NOT allow cash back for offline PIN CVM • 1 = Allow cash back for offline PIN CVM
Online PIN cashback	11	• 0 = Do NOT allow cash back for online PIN CVM • 1 = Allow cash back for online PIN CVM
Signature cashback	12	• 0 = Do NOT allow cash back for signature CVM • 1 = Allow cash back for signature CVM Note: Discover cards support signature cashback.
No CVM cashback	13	• 0 = Do NOT allow cash back for No CVM • 1 = Allow cash back for No CVM Note: No known cards support No CVM cash back.
Reserved	14	Reserved, set to 0.

Parameter	Position from Left	Description
Fast Quick Chip	15	Fast Quick Chip is allowed for this AID. The parameter, 0019_0021, must also be enabled in emv.dat to support Fast Quick Chip on the terminal. • 0 = Do NOT allow Fast Quick Chip for this AID • 1 = Allow Fast Quick Chip for this AID

5.2.10.1 *PIN-Bypass Guidelines*

- For PIN Bypass, the terminal sets tag T95 (Terminal Verification Results) Byte 3 Bit 4 = 8 to indicate PIN entry is required, PIN pad present, but PIN was not entered.
- The PIN Bypass feature also requires tag 9F840A (Support Bypass PIN Entry) to be set to 01 in `EMVCONTACT.XML`.
- PIN entry can begin in one of the following ways:
  - Automatically during the EMV flow
  - On demand with the 31.x message

5.2.10.2 *Pin-Bypass Process Examples*

In the following examples, the Allow PIN Bypass flag is set to 1:

Standard Flow

1. The cardholder bypasses PIN entry during EMV flow.
2. The flow proceeds (using another CVM if appropriate).

On Demand

1. The POS sends the 31.x message.
2. The terminal displays the PIN entry screen.
3. The cardholder presses Enter without entering any digits.
4. RBA responds with 31.40x0D, where `0x0D` is the binary value for the Enter key.

The card's rules may decline the transaction if PIN is not entered.

5.2.11 EMV Brand Parameters (emvbrand.dat)

This section describes the `emvbrand.dat` parameters. The AID Brand value for each `emvaid.dat` item corresponds to a parameter in `emvbrand.dat`. For example, the 0021_0002 parameter's AID Brand value of 01 refers to `emvbrand.dat` parameter 0022_0001.

emvbrand.dat Parameters

DFS Data Index	EMV Brand	US Common/Global Debit AID Preference	Card
0022_0001	01	Default value is 0.	MasterCard
0022_0002	02	Default value is 0.	VISA
0022_0003	03	Default value is 0.	Interac
0022_0004	04	Default value is 0.	AMEX
0022_0005	05	Default value is 0.	Discover
0022_0006	06	Default value is 0.	DNA (US Debit)
0022_0007	07	Default value is 0.	UnionPay
0022_0008	08	Default value is 0.	JCB
0022_0009	09	Default value is empty.	
0022_0010	10	Default value is empty.	

The field "US Common/Global Debit AID preference" configures whether the RBA automatically prefers global or US common debit AIDs.

This setting applies to both contact and contactless EMV cards that have two application IDs for the same card brand, both tied to the same funding source, where one is the US Common Debit AID for that brand.

Values are as follows:

- 0 = do not prefer US Common or Global Debit AID (default)
- 1 = prefer US Common Debit AID
- 2 = prefer Global Debit AID

In cases 1 or 2, only the preferred AID will be included in the candidate list for selection or confirmation. The other AID will be removed from the candidate list.

To prefer the US common debit AIDs over global debit AIDs, set each brand preference to 1.

5.2.12 Form Files (forms.dat)

The first 18 forms are the standard forms described in the Appendix G: Forms chapter in this document, while the additional forms can be custom forms developed specifically for each user.

Each form is identified in the `forms.dat` portion of `config.dfs`.

See Form Contents and Descriptions for more information on each form's appearance and contents.

Note
The PayPal versions of the forms noted, below, include a PayPal button in place of the **Enter Card** button.

forms.dat Parameters

Parameter (Form) Name	DFS Data Index	Default Value	Description
Offline	0030_0001	OFFLINE.K3Z	This is the form that the terminal will display when it is offline. The PayPal version of this form is `PPOFFLINE.K3Z`.
Message	0030_0002	MSG.K3Z	This is the form that the terminal uses to display a message while waiting for the STB response.
Language Selection	0030_0003	LANG.K3Z	This is the form that the terminal will display to prompt the user to select the language desired.
Card Swipe	0030_0004	SWIPE.K3Z	This is the form that the terminal will display to prompt the cardholder to swipe his magnetic stripe card. The PayPal version of this form is `PSWIPE.K3Z` for Card Swipe with PayPal.

Parameter (Form) Name	DFS Data Index	Default Value	Description
Card Swipe with Language Selection	0030_0005	LSWIPE.K3Z	This is the form that the terminal will display to prompt the cardholder to select a language and to swipe his magnetic stripe card. Displayed if Combine Language Swipe Screens parameter ('0007_0004') is set to 1, <combine screens>. The PayPal version of this form is `PLSWIPE.K3Z` for Card Swipe with PayPal and Language Selection .
Payment Selection	0030_0006	PAY%d.K3Z	This is the template for the form name to use when prompting the customer for payment type (credit, debit, etc.). This template must include the characters "%d". These are replaced with the menu number currently displayed. Menu 1 is always displayed first.
Cash Back Selection	0030_0007	CASHB.K3Z	This form can be used to prompt the customer for the cashback amount or can ask the customer if cashback is desired. If it does not ask for an amount, the form defined in '0030_0008' is used to prompt for an amount.
Cash Back Selection without No	0030_0008	CASHBA.K3Z	This is the form that the terminal will display to prompt the user to select a cash back amount from several choices.
Cash Back Verification	0030_0009	CASHBV.K3Z	This is the form used for verification of cashback amounts.
Amount Verification	0030_0010	AMTV.K3Z	This is the form that the terminal will display to prompt the user to confirm the purchase amount (i.e., Amount OK? $25.99).

Parameter (Form) Name	DFS Data Index	Default Value	Description
Post-Sign	0030_0011	POSTSIGN.K3Z	This is the form that the terminal will display to prompt the user to sign his name. This form follows the `PRESIGN.K3Z` form once signature has been initiated.
Terms and Conditions	0030_0012	TC.K3Z	This is the form the terminal uses when displaying a terms and conditions screen.
Terms and Conditions Signature	0030_0013	TCSIGN.K3Z	This is the form the terminal uses when displaying a terms and conditions screen with a signature field.
Card Swipe On Demand	0030_0014	COD.K3Z	This is a form that the terminal displays to prompt the user to swipe his magnetic stripe card.
PIN Entry	0030_0015	PIN%c.K3Z	This form is used for PIN entry. The RBA looks for a PIN form based on the payment type first. For example, if payment type A, debit, is selected, `pina.K3Z` is used. If `pina.K3Z` is not found, `pin.K3Z` is used. The "%c" is dropped to create the second form name.
Numeric Input	0030_0016	INPUT.K3Z	This form is used for numeric input only.
Alphanumeric Entry	0030_0017	ALPHA.K3Z	This form includes an on-screen keyboard and is used for alphanumeric text entry.
Cash Back Other	0030_0018	CASHBO.K3Z	This is the form used for manual entry of cashback.
Advertising	0030_0019	ADS.K3Z	This is the form that the terminal uses to display advertising screens.

Parameter (Form) Name	DFS Data Index	Default Value	Description
Input Entry	0030_0020	INPUT.K3Z	This form is used during clear text entry, such as cash back or the input request message (see 21.x Numeric Input Request Message (On-Demand)). This form is also used to support the PIN entry with optional credit selection option as described in the Enter PIN or Press Green for Credit forms section.
Contactless Card Swipe	0030_0021	CSWIPE.K3Z	This is the form that the terminal will display to prompt the cardholder to tap his contactless card on the terminal. The PayPal version of this form is `CPSWIPE.K3Z` for Contactless Card Swipe with PayPal Selection.
Contactless Card Swipe with Language Selection	0030_0022	CLSWIPE.K3Z	This is the form that the terminal will display to prompt the cardholder to select a language and to tap his contactless card on the terminal. Displayed if Combine Language Swipe Screens parameter ('0007_0004') is set to 1, <combine screens>. The PayPal version of this form is `CPLSWIPE.K3Z` for Contactless Card Swipe with PayPal and Language Selection.
Contactless Card Read Request	0030_0023	CCOD.K3Z	This is a form that the terminal displays to prompt the user to tap his contactless card on the terminal.

Parameter (Form) Name	DFS Data Index	Default Value	Description
Approved/Disapproved	0030_0024	APPDAPP.K3Z	This new screen appears after the approval / disapproval state. Replaces the '0030_0002' `MSG.K3Z` form that used to be used at this point.
Waiting for STB Response	0030_0025	MSG.K3Z	Waiting for STB (Spin the BIN) response.
Survey Swipe	0030_0026	SURSWIPE.K3Z	This is the form that the terminal will display to prompt the customer to either select a button in response to the displayed survey question, or to swipe their card for payment. This form is used with the 40.x and '40.0' survey messages.
PayPal Data Input (On-Demand form)	0030_0027	PPALINP.HTM	Used with PayPal, this is the form used by the cardholder to input a ten-digit numeric code (phone number) to the terminal. This form also used for all input except the PayPal PIN. This form is in .HTM file format (not .K3Z).
PayPal PIN Entry	0030_0028	PPALPCAN.HTM	Requests the cardholder’s PayPal PIN for PayPal authorization. This form is in .HTM file format (not .K3Z).
PayPal Please Wait	0030_0029	PPWAIT.K3Z	Requests that the PayPal cardholder wait for approval or denial.
Remove Inserted Card	0030_0030	REMOVE.K3Z	Prompts the cardholder to remove a card once it has been inserted. Specific to the iUN-series terminals.

Parameter (Form) Name	DFS Data Index	Default Value	Description
EMV Select Account	0030_0031	EACCOUNT.K3Z	EMV select account.
EMV Select Language	0030_0032	ELANG.K3Z	EMV select language.
Smart Card (EMV) Confirm Application	0030_0033	ECONFIRM.K3Z	EMV confirmation.
Menu	0030_0034	MENU.K3Z	Menu.
Message Screen	0030_0035	MSGTHICK.K3Z	Message screen.
Smart Card (EMV) and Swipe	0030_0036	ESWIPE.K3Z	SMC and Swipe.
Smart Card (EMV) and Swipe with Language Selection	0030_0037	ELSWIPE.K3Z	SMC and swipe with language selection.
Contactless Smart Card (EMV) and Swipe	0030_0038	CESWIPE.K3Z	Contactless SMC and swipe.
Contactless Smart Card (EMV) and Swipe with Language Selection	0030_0039	CELSWIPE.K3Z	Contactless SMC and swipe with language selection.
Pre-Sign	0030_0040	PRESIGN.K3Z	Pre-sign signature (enables Cancel). For On-Demand signing, parameter '0030_0041' is used for setting the form. The On-Demand form is not replaced with the Signature form since the "Cancel" button and keypad key remain function through the signature process.
Signature (On-Demand)	0030_0041	SIGN.K3Z	On-demand signature (may enable Cancel).
Smart Card Insert	0030_0042	SMCINSERT.K3Z	Smart Card Insert.
Smart Card Message	0030_0043	SMCMSG.K3Z	Smart Card Message.
Smart Card Update	0030_0044	SMCUPDATE.K3Z	Smart Card Update.

Parameter (Form) Name	DFS Data Index	Default Value	Description
Generic Entry Form	0030_0045	CLESS.K3Z	Generic contactless entry form.

Files 0031 – 0034 are reserved for Prompt Files (when using the 60.x Configuration Write and 61.x Configuration Read messages).

For contactless-related parameters, parameter '0008_0001' in `cless.dat` must be set to a value of '1' to enable the contactless feature.

5.2.13 MAC Entry (mac.dat)

This section describes the MAC entry parameters located in the mac.dat file.

mac.dat Parameter

Parameter Name	DFS Data Index	Default Value	Description
MAC Key Index	0016_0001	1	This is the MAC key index. • The MAC key index has a value from '0' to '9'.

5.2.14 Main Flow (mainFlow.dat)

This section describes the parameters that control the main flow of data. These parameters are found in the `config.dfs` file under the heading, Main Flow, and file name `mainFlow.dat`.

<table>
<thead>
<tr><th>Parameter Name</th><th>DFS Data Index</th><th>Default Value</th><th>Description</th></tr>
</thead>
<tbody>
<tr><td>Duration to Display Results</td><td>0007_0001</td><td>30</td><td>Used in conjunction with parameters 0007_0022, 0007_0023, and 0007_0037 through 0007_0043 to enable individual control of the display duration for the results of the messages listed in the following table:
<table>
<thead><tr><th>Parameter</th><th>Message</th></tr></thead>
<tbody>
<tr><td>0007_0037</td><td>20.x Signature Message (on-demand)</td></tr>
<tr><td>0007_0038</td><td>21.x Numeric Input Request Message (on-demand)</td></tr>
<tr><td>0007_0039</td><td>23.x Card Read Request (on-demand)</td></tr>
<tr><td>0007_0040</td><td>24.x Form Entry Request (on-demand)</td></tr>
<tr><td>0007_0041</td><td>25.x Terms and Conditions Request (on-demand)</td></tr>
<tr><td>0007_0042</td><td>27.x Alpha Input Message (on-demand)</td></tr>
<tr><td>0007_0043</td><td>31.x PIN Entry Messages (on-demand)</td></tr>
</tbody>
</table>
Parameter value is assigned as follows:
• 0 = Do not display results
• 1 = Display results with no timeout
• 2 = Use individual flags (0007_0037 through 0007_0043)
• >2 = Display results duration in 1/10 seconds
To enable configuration parameters 0007_0037 through 0007_0043 to control display duration for these messages, this parameter must be set to 2.</td></tr>
<tr><td>Default Language</td><td>0007_0002</td><td>1</td><td>The default language.
• 1 = English
• 2 = Spanish (Espanol)
The value must be less than or equal to the number of supported languages defined in the Number of Supported Languages parameter 0007_0003.</td></tr>
<tr><td>Number of Supported Languages</td><td>0007_0003</td><td>2</td><td>Number of languages the customer can choose from.
• Maximum value = 3
If more than one language is specified, the Select Language prompt is displayed at the beginning of a transaction.</td></tr>
</tbody>
</table>

Parameter Name	DFS Data Index	Default Value	Description
Combine Language with Card Swipe Screens	0007_0004	1	Is enabled only when the 0007_0009 Customer Action parameter is set to 0. The terminal displays a form that combines the Slide Card and Select Language prompts, which allows the cardholder the option of selecting the language before swiping the card. • 0 = Separate screens (uses forms `lang.K3Z` and `swipe.K3Z`) • 1 = Combine screens (default, uses form `lswipe.K3Z`)
Customer Action (CA)	0007_0005	0	Specifies the first action for the customer: • 0 = Slide card first, followed by payment selection (default) • 1 = Select payment type first, followed by card swipe
On Incorrect Total	0007_0006	0	Specifies where to return in the transaction if the total amount was incorrect and the cash back option is available. • 0 = Return to Wait for Amount screen (default) • 1 = Reset transaction, return to beginning • 2 = Return to cash back entry screen ◦ If the cashback process was not used during the transaction, RBA returns to waiting for the amount message from host.
Clear Line Display on Reset	0007_0007	1	Specifies whether the line display should be cleared following a hard reset message. If the display is not cleared on a reset, the POS must send a 15.8 Soft Reset message to clear the screen. For more details on clearing the line display, refer to the 10.x Hard Reset Message. • 0 = Do not clear display • 1 = Clear display (default)
Show Payment Type Timer	0007_0008	20	Specifies if the selected payment type should be briefly displayed after it is selected. Text for each payment type is listed in the file `PROMPT.XML`. • 0 = Do not display • >0 = Time to display in 1/10 seconds

Parameter Name	DFS Data Index	Default Value	Description
Reset Response	0007_0009	2	Specifies if a 10.x Hard Reset Message should be sent to the POS in response to a reset message from the POS. - 0 = Do not send (default). - 1 = Send. - 2 = Send only if amount has been received.
Online Message Action	0007_0010	1	Specifies whether the online message starts a new payment transaction or displays advertising until a reset is received. - 0 = Go to advertising. - 1 = Start a transaction (default).
Splash Screen Application Name	0007_0011		Specifies a custom application name to display on the application splash screen. Maximum of 25 characters.
Splash Screen Version Number	0007_0012		Specifies a custom application version number to display on the splash screen. Maximum of 25 characters.
Splash Screen Custom Part Number	0007_0013		Custom part number for splash screen.

Parameter Name	DFS Data Index	Default Value	Description
Beep Volume	0007_0014	25	Specifies the volume of the beep. Set this parameter via the 60.x Configuration Write message. • Enter a value of 0 (no sound) to 100 (high volume). This can be configured even when 0008_0003 = 0. This parameter is supported on the following terminals: • iSC250 • iSC350 • iSC480 • iWL250 • iSMP
Alternate On Demand Mode	0007_0015	0	Turns on or off the alternate demand mode to enable or disable the 34.x Save and Restore State Messages. • 0 = Automatically save and restore state (Compatible mode) ◦ The current state is saved when an on-demand message is received, and restored when the command is completed • 1 = New mode ◦ The state is saved using the 33.x message. The application does not return to the saved state. The state is restored using the 34.x message.

Parameter Name	DFS Data Index	Default Value	Description
Terms and Conditions Form: Accept Button	0007_0019	0	Determines when the cardholder sees and presses the Accept button for the Terms and Conditions form. • 0 = Show <Accept> and <Decline> buttons regardless of current position in the text • 1 = Show both <Accept> and <Decline> buttons regardless of current position in the text ◦ Allow the <Accept> button only at the bottom of the text ◦ Allow the <Decline> button regardless of the current position in the text • 2 = Show the <Accept> button only at the bottom of the text. Show the <Decline> button regardless of the current position in the text • 3 = Show both <Accept> and <Decline> buttons only at the bottom of the text
Terms and Conditions Form: Hide Disabled Direction Buttons	0007_0021	0	Toggles an option to hide the <Up> button at the top of the text, and hide the <Down> button at the bottom. • 0 = Always show both directional buttons ◦ Allow only the <Down> button when at the top of the text ◦ Allow only the <Up> button when at the bottom of the text ◦ Allow both the <Up> button and <Down> button when in the middle of the text • 1 = Show and allow enabled buttons only ◦ Show and allow the <Up> button when not at the top of the text ◦ Show and allow the <Down> button when not at the bottom of the text
Time to Display Approval/ Decline Message	0007_0022	50	Sets the amount of time the approval or decline message will display: • 0 = Do not display • 1 = Display until a reset is received • 2 - 65000 = Time to display in 1/10 second **Note:** If 0007_0022 is 1 or 2, 0007_0001 must be set to 2.

Parameter Name	DFS Data Index	Default Value	Description
After Approval/ Decline Display	0007_0023	1	Determines the next action following the approval or decline message: • 0 = Reset • 1 = Go to advertising. Parameter 0010_0001 must be set to 1 or 3 • 2 = Wait for reset. Parameter 0007_0022 must be set to 0 or 1
Wait to Send Authorization Request	0007_0024	0	Determines when the authorization request will be sent: • 0 = Send Authorization Request as soon as ready • 1 = Wait for a 50.x Authorization Request message from the POS
# of Lines in Scrolling Receipt Buffer	0007_0025	16	Determines the number of lines to use for the scrolling receipt buffer. • Minimum value is 16
Min # of Digits Required for Keyed Card	0007_0026	13	Minimum number of digits required for a keyed card. • Allowable values are 1 – 24
Max # of Digits Required for Keyed Card	0007_0027	20	Maximum number of digits required for a keyed card. • Allowable values are 1 – 24
Automatic On- Demand Function Cancel	0007_0028	1	Determines whether to allow an on-demand message to automatically cancel any currently running on-demand function. • 0 = Do not automatically cancel running on-demand functions • 1 = Automatically cancel running on-demand functions.

Parameter Name	DFS Data Index	Default Value	Description
Display Enter Card Prompt	0007_0029	0	Controls the display of the Enter Card button at the bottom of the Card Swipe screen. - 0 = Do not display. - 1 = Display button and prompt for card number, expiration date, and CVV. - 2 = Display button and prompt for card number and expiration date (no CVV). - 3 = Display button and prompt for card number and CVV (no expiration date). - 4 = Display button and prompt for card number (no expiration date, no CVV). - 5 = Do not display button. Enable dynamic manual entry for forced mnual entry by sending 23.[FS][FS][FS]H. - 6 = Display button. Enable dynamic manual-entry process using 19.x messages. When using TDES encryption and manually entering data, the PAN, expiration date, and CVV are all required. This parameter must be set to 0 or 1 when using this encryption mode. When the terminal is configured for payment selection before swipe (0007_0005 = 1), the Expiration Date and CVV entry are controlled based on card type, rather than the global flag (0007_0029). See 0011_xxxx parameters in `config.dfs` for details on card type settings.
CVV and/or Expiration Date not entered for manual card entry	0007_0030	0	Sets the manual card entry if not prompted to enter the CVV and/or expiration date. - 0 = Leave blank. - 1 = Use ASCII zeros as placeholders in track data. When any P2PE type is enabled (e.g., parameter 0091_0001 is a value other than 0), this parameter (0007_0030) is ignored, and zeros are used as placeholders for expiration date and CVV data. EPS encryption leaves the expiration date and CVV blank if they are not entered manually.

Parameter Name	DFS Data Index	Default Value	Description
Backlight	0007_0031	0	Backlight Flag. • 0 = Backlight is disabled. • 1 = Backlight is enabled.
Bluetooth Unpairing	0007_0032	(empty)	Bluetooth unpairing string • = Bluetooth unpairing is disabled from the application. Must unpair via the Telium Manager. This value is otherwise equal to a customized string that enables Bluetooth unpairing when the string is scanned with the barcode scanner.
Bluetooth Pairing	0007_0033	0	Bluetooth pair method. • 0 = Not configured (default) • 1 = iOS • 2 = Standard
Inactivity Timeout	0007_0034	15	Inactivity timeout for portable terminals in minutes. At the specified duration, the terminal goes into sleep mode. • 0 = Disabled • 1 - 200 = Number of minutes of inactivity before the terminal goes into sleep mode. This parameter is ignored if contactless and/or smart card readers are active.

Parameter Name	DFS Data Index	Default Value	Description
Automatic Power Down	0007_0035	60	Automatic power down for portable terminals in minutes. At the specified duration, the terminal powers down. - 0 = Disabled - 1 - 200 = Number of minutes of inactivity before powering down a battery-powered terminal This parameter is ignored if contactless and/or smart card readers are active. **Note** For iSMP4 terminals, when the battery level is less than five percent, and the battery is not on the charging base, the terminal powers off. An alarm sounds every five seconds for the last minute before powering off, prompting the user to place the terminal on a charger. See Parameter 830, Battery Power %, in 28.x Set Variable Request. If the terminal is in sleep mode, applications are idle, and the terminal cannot power off automatically, but when the charge reaches zero percent, the terminal powers off.
Display Backlight Intensity	0007_0036	70	Display backlight intensity 0-100%. - 0 = Off - 1 - 100 = brightness level (%) Very low levels can cause the backlight to appear to be off, depending on the terminal model. For the iPP350, values higher than 50 have the same effect as a value of 50.

Parameter Name	DFS Data Index	Default Value	Description
20.x Signature Request Message Result Display Duration	0007_0037	30	Duration to display result of 20.x Signature Message (on-demand). Used when parameter 0007_0001 is set to 2. • 0 = Do not show. • 1 = Show result but no timeout. • >1 = Duration in 1/10 seconds. Set display timeout for on-demand message result individually with parameters 0007_0037 to 0007_0043. Used when parameter 0007_0001 is set to 2.
21.x Numeric Input Request Message Result Display Duration	0007_0038	30	Duration to display result of 21.x Numeric Input Request Message (on-demand). Used when parameter 0007_0001 is set to 2. • 0 = Do not show • 1 = Show result but no timeout • >1 = Duration in 1/10 seconds
23.x Card Read Request Result Display Duration	0007_0039	30	Duration to display result of 23.x Card Read Request (on-demand). Used when parameter 0007_0001 is set to 2. • 0 = Do not show • 1 = Show result but no timeout • >1 = Duration in 1/10 seconds
24.x Form Entry Request Result Display Duration	0007_0040	30	Duration to display result of 24.x Form Entry Request (on-demand). Used when parameter 0007_0001 is set to 2. • 0 = Do not show • 1 = Show result but no timeout • >1 = Duration in 1/10 seconds
25.x Terms and Conditions Request Result Display Duration	0007_0041	30	Duration to display result of 25.x Terms and Conditions Request (on-demand). This is only used when parameter 0007_0001 is set to 2. • 0 = Do not show • 1 = Show result but no timeout • >1 = Duration in 1/10 seconds

Parameter Name	DFS Data Index	Default Value	Description
27.x Input Message Result Display Duration	0007_0042	30	Duration to display result of 27.x Alpha Input Message (on-demand). Used when parameter 0007_0001 is set to 2. • 0 = Do not show. • 1 = Show result but no timeout • >1 = Duration in 1/10 seconds
31.x PIN Entry Message Result Display Duration	0007_0043	30	Duration to display result of 31.x PIN Entry Messages (on-demand). Used when parameter 0007_0001 is set to 2. • 0 = Do not show. • 1 = Show result but no timeout. • >1 = Duration in 1/10 seconds.
Terminal Country	0007_0044	0	Terminal country: • 0 = USA. • 1 = Canada.
Enable 24 Hour Reboot	0007_0045	0	Automatic reboot after 24 hours' continuous terminal run time: • 0 = Disable. • 1 = Enable. PCI v4 terminals treat this parameter as always enabled, and will automatically reboot 24 hours after their previous reboot. After this feature is enabled, the unit must be returned to Ingenico to have it disabled.

Parameter Name	**DFS Data Index**	**Default Value**	**Description**
Daily Reboot Time	0007_0046	0	Daily reboot time in 24-hour format. This value is the time in each 24-hour period when the terminal reboots automatically. • 0 = No daily reboot. • >0 = Daily reboot time in 24-hout HHMM format. If the terminal is not in idle mode (offline) when the scheduled reboot time is reached, the terminal reboots when it enters the next idle state.
Status Message Configuration	0007_0047	1	Is configured when the 09.x Card Status Message is sent. This setting is used when 0019_0001 (EMV Support) and 0020_0001 (WIC Support) are both set to 1 only. • 0 = Disabled: 09.x message is never sent. • 1 = Limited: 09.x message is only sent before/after smartcard transactions (default for backwards compatibility) • 2 = Verbose: 09.x message is always sent during normal transaction flow.
Status Message Configuration	0007_0048	1	Enables tokenization support. • 0 = Off • 1 = Type 1 Tokenization (KME and On-Guard encryption only) • x = Not supported

Parameter Name	DFS Data Index	Default Value	Description
Display Please Hand Card to Cashier Prompt	0007_0049	1	Enables the display of Please Hand Card to Cashier prompt after three consecutive bad swipes. • 0 = Do not display. • 1 = Display button and prompt for card number, expiration date, and CVV. • 2 = Display button and prompt for card number and expiration date (no CVV). • 3 = Display button and prompt for card number and CVV (no expiration date). • 4 = Display button and prompt for card number (no expiration date, no CVV). When using TDES encryption and manually entering data, the PAN, expiration date, and CVV are required. Must be set to 0 or 1. When the terminal is configured for payment selection before swipe (0007_0005 = 1), the Expiration Date and CVV entry are controlled based on card type rather than the global flag (0007_0029). See 0011_xxxx parameters in `config.dfs` for details on card type settings.
Display Please Swipe with Enter Card Prompt	0007_0050	1	Terminal sends 09.x response to POS with a **max bad swipe** error. Enter Card prompt is displayed with Please Swipe prompt after three consecutive bad swipes. • 0 = Disable • 1 = Enable

Parameter Name	DFS Data Index	Default Value	Description
Display serial numbers for iUC, iUP, and iUR in 07.x and 08.x messages	0007_0051	0	Sends 07.x and 08.x responses to the POS with the terminal serial number. If it is an iUN, each component (iUC, iUP, and iUR) returns their serial number: • 0 = Give iUP serial number • 1 = Give serial number for each terminal If a peripheral device is not connected at boot time, the iUP250 reboots three times before loading RBA. This parameter must be set to 1 to show the serial numbers of the peripheral devices in the 07.x message. If a device is disconnected, its serial number is blank.
Media volume	0007_0052	255	The volume at which media (video or audio files) play. • 0 = Media muted • 1-255 = Media volume
F button behavior	0007_0053	0	Action taken by pressing the **F** button four times at the Offline screen: • 0 = Open communications menu • 1 = Disable feature (legacy) • 2 = Bluetooth. If the terminal is not Bluetooth-enabled, changes this setting to 1 (disabled)
Backlight timer shutoff	0007_0054	120 seconds	Duration of time until the terminal backlight is turned off in seconds.
Cancel button as Stop button	0007_0055	0	Cancel button works as a Stop button on the iUC285 COD/CCOD form. • 0 = Disable • 1 = Enable
Enable iSMP4 current charge setting	0007_0056	0	Enable the ISMP4 to set current charge settings when used with a Wi-Case in serial mode. • 0 = Disable • 1 = Enable

5.2.15 MSR Card Swipe Options (msr.dat)

These parameters, used to set the swipe options for the magnetic stripe reader (MSR), are found in `config.dfs` under the heading, Terminal Local MSR Card Swipe Options, and filename `msr.dat`.

msr.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
Bad Swipes Allowed Before Displaying Assistance Message	0003_0001	3	This parameter specifies how many times a customer can swipe a bad card before the Ask for Assistance prompt or Hand Card to Cashier prompt is displayed. • 0 = Disables the display of the Ask for Assistance or Hand Card to Cashier prompt. • 1 – 65,000 = Defines the number of card swipes allowed before the Ask for Assistance or Hand Card to Cashier prompt displays (default is 3 swipes). After any bad card swipe, the following prompt is displayed for 3 seconds: Card Read Error, Please Try Again.
Ask for Assistance Duration	0003_0002	30	This parameter specifies whether to display the Ask for Assistance prompt or Hand Card to Cashier prompt after the maximum amount of bad card swipes has been reached. • 0 = Off, do not display message • 1 - 5,000 = Duration of display in 1/10th of a second

Parameter Name	DFS Data Index	Default Value	Description
After Max Bad Card Swipes Reset Transaction	0003_0003	1	After the maximum amount of bad card swipes has been reached, this parameter specifies whether to reset the transaction or prompt the customer to reswipe their card. • 0 = Reset transaction • 1 = Prompt for card swipe (default)
Assumed Payment Track	0003_0004	2	The track selected in this parameter is copied to the payment track variable when a card is swiped. Once the payment is selected, the track may be changed. • 0 = Don't set variable until payment is selected • 1 = Assume Track 1 • 2 = Assume Track 2 (default) • 3 = Require Track 1 • 4 = Require Track 2 • 5 = Require both tracks and return read error if either/both not read. Track 1 is stored to RBA variable 405.
Reformat Name Field in Track 1 if in Form Last/First	0003_0005	0	Changes the formatting of the name field based on the value of the parameter: • 0 = Do not modify the name field • 1 = The name field is searched for the / character. If found, the text before and the text following the / character are swapped. The / character is replaced with a space. For example, the name Williams/ Fred is changed to Fred Williams.

Parameter Name	DFS Data Index	Default Value	Description
Bad Read Error Timeout	0003_0007	30	Timeout for display of “bad read” error. • 0 = Disabled • 1 - 65000 = time in 1/10ths of a second
Enable MSR During Bad Read Error	0003_0008	0	Enable MSR when displaying card read error. • 0 = Disable • 1 = Enable
Beep After Card Read	0003_0009	1	Determines whether to sound a “beep” when a card is read. • 0 = Disable • 1 = Enable
Append Track 3	0003_0010	0	Append Track 3 to 23.x card read response. • 0 = Send only Track 1 and Track 2 • 1 = Send Track 1, Track 2 and Track 3
Number of Readable Digits	0003_0016	4	Number of non-obscured digits in displayable account number.
Enable MSR Lights	0003_0017	1	Turn on MSR lights when MSR is enabled. • 0 = Disable • 1 = Enable
Bad read delay for unattended	0003_0018	8	Seconds allowed between an invalid swipe and "bad read" error. iuR only.

5.2.16 PayPal Configuration (paypal.dat)

This section describes the parameters used to configure PayPal authorization support, and for testing efforts prior to production. These parameters are found in the `config.dfs` file under the heading, PayPal Config, and filename `paypal.dat`.

Info
For an overview of PayPal configuration needs, including minimum production requirements, PayPal validation flow, calculating GMT offset, and related forms, see Appendix C. PayPal Overview.

paypal.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
Your Key Name	0040_0002	"GENERIC_T.PEM"	This parameter specifies your key file name. This must be set to enable PayPal payment (see also '0040_0006'). • Your Key Name This parameter supports a maximum of 15 characters.
PayPal Payment Type/ Enable/Disable	0040_0006	0	This parameter specifies the PayPal payment type. Single digit, 7 (e.g., '0011_0007'). This parameter also defines whether or not PayPal support is enabled. By default, PayPal is disabled. • 0 = Disable PayPal Support (default) • 1 or greater = Enable PayPal Support
PayPal/Discover BIN table number	0040_0008	1	This number must match the BIN table entry.
PayPal BIN table number (non-Discover)	0040_0009	11	This number must match the BIN table entry.
Send PayPal Preauthorization message	0040_0010	1	Specifies whether or not to send the POS a preauthorization message. • 0 = Do not send message • 1 = Send preauthorization message as soon as possible • 2 = Send preauthorization after POS sends 52.x message and data is available

5.2.17 PIN Entry (pin.dat)

This section describes the parameters used to configure PIN entry. These parameters are found in the `config.dfs` file under the heading, PIN Entry, and filename `pin.dat`.

pin.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
Overall Timeout for PIN Entry in Seconds	0006_0001	0	Specifies how long to wait for the customer to enter a PIN before timing out. • 0 = 60 seconds (maximum value) • 1 - 600 = Time in 1/10 of a second
First Key Timeout for PIN Entry in Seconds	0006_0002	0	Specifies how long to wait for the customer to enter the first digit of a PIN before terminating PIN entry. • 0 = 60 seconds • 100 - 600 = Time in 1/10 of a second
Between Keys Timeout for PIN Entry in Seconds	0006_0003	0	Specifies how long to wait for the customer after one digit of a PIN has been entered and before the next digit before timing out. • 0 = 60 seconds • 20 – 600 = Time in 1/10 of a second
0-Length PINs	0006_0004	0	Allows 0 length PINs in 31.x message. • 0 = A valid PIN must be entered • 1 = Either an empty or valid PIN must be entered
Display Timeout for Assistance Message	0006_0006	30	Specifies how long to display the *Ask for Assistance* message before timing out. • 0 - 65,000 = Time in 1/10 of a second
PIN Encryption Method	0006_0007	1	Specifies which encryption method to use. The environment index must be specified in the Encryption Environment Index parameter (0006_0008). • 0 = Master/Session • 1 = DUKPT (default)
Encryption Environment Index	0006_0008	0	Specifies a Master/Session key to use for encrypting the PIN information. • 0 - 9 indicates the key index to use

Parameter Name	DFS Data Index	Default Value	Description
Minimum Number of Digits for PIN Entry	0006_0011	4	Sets the minimum number of digits allowed during PIN entry. • 4 - 12 are the only valid values. Must be less than or equal to 0006_0012.
Maximum Number of Digits for PIN Entry	0006_0012	12	Sets the maximum number of digits allowed during PIN entry. • 4 - 12 are the only valid values. Must be greater than or equal to 0006_0011.

Parameter Name	DFS Data Index	Default Value	Description
Enable PIN Bypass for MSR	0006_0013	0	Enables PIN Bypass for MSR to emulate EMV. • 0 = ◦ The <Cancel> key implements cards.dat:: **On PIN Entry Cancel**. If the <Cancel> key is pressed, the flag will contain one of the following values: ▪ 0 = Cancel transaction. ▪ "-" = Restart PIN entry. ▪ 1 - 9 = Load payX form. ▪ A - P = Select card type. ◦ The <Enter> key either: ▪ accepts PIN if minimum number of PIN digits (4) entered or ▪ beeps and continues PIN entry if less than the minimum number of digits entered. • 1 = ◦ The <Cancel> key cancels the transaction. ◦ The <Enter> key implements cards.dat:: **On PIN Entry Cancel**. The <Cancel> key will now cancel the transaction. If the <Enter> key is pressed, the flag will contain one of the following values: ▪ 0 = Cancel transaction. ▪ "-" = Restart PIN entry. ▪ 1 - 9 = Load payX form.

Parameter Name	DFS Data Index	Default Value	Description
			▪ A - P = Select card type. The minimum number of digits must be entered (4 by default). If attempting to submit fewer than the minimum, the terminal beeps and continues PIN entry.
Send PIN key press message	0006_0014	0	Determines whether to send a 31.A Pin Entry Response each time a button is pressed during PIN entry. • 0 = Send no message per key press. • 1 = 31.A PIN key press messages sent for each PIN key press.

5.2.18 Changes to security.dat or secbin.dat

The following policy applies to all MSR encryption methods.

If your organization requires changes to either the `security.dat` or `secbin.dat` file, you must follow the procedure to obtain approval and signature from Ingenico prior to your implementation. See Signing Requirements for .DAT File Changes for more information.

> **Info**
> You will be unable to implement your changes to the `security.dat` and `secbin.dat` files without a signed .PGZ file from Ingenico. If you implement your changes prior to receipt of the new .PGZ file, your Telium terminals may appear to run properly, however, your terminals will actually be running as previously configured, without your changes. See the process diagram on the following page for approval process information.

> **Info**
> It is highly recommended that you use the 61.x Configuration Read Message to ensure your changes to this file are applied correctly once implemented.

Contact your Ingenico Account Manager with any questions you may have about the signing process.

5.2.19 Security BIN (secbin.dat)

This section describes the security parameters in the `secbin.dat` file. If your organization requires changes to the `secbin.dat` file, you must follow the procedure to obtain approval and signature from Ingenico before your implementation. See Signing Requirements for .DAT File Changes.

It is highly recommended that you use the 61.x Configuration Read Message to ensure your changes to this file are applied correctly.

Info
This policy applies to all P2PE encryption methods.

Note
The start and end BINs are compared for the full the length of the entry.

The first parameter (0092_0001) enables the Security BIN Table itself. The default setting is 0 = Off/Disabled. Use 1 = On/Enabled to enable the table.

Parameter 0092_0001 functions as follows:

- If set to 0, then all card data is encrypted, if parameter 0091_0001 is not 0.
- If set to 1, and if parameter 0091_0001 is not 0, then encryption of card data matching a BIN entry in `secbin.dat` depends on the encryption flag setting for the matching BIN.
- If set to 1, and if parameter 0091_0001 is not 0 and the card data does not match a BIN entry in `secbin.dat`, then the card data is encrypted.

The following table describes the BIN information for various card types. One to eight digits can be specified for the start and end of each BIN range.

Default `secbin.dat` Parameters

Parameter Name	DFS Data Index	Start BIN Range	End BIN Range	Min PAN Length	Max PAN Length	Encrypt	Repair Invalid Tracks
Discover 1 card	0092_0002	60110000	60119999	14	16	1 (Yes)	0
Discover 2 card	0092_0003	62212600	62292599	14	16	1 (Yes)	0
Discover 3 card	0092_0004	64400000	64999999	14	16	1 (Yes)	0
Discover 4 card	0092_0005	65000000	65999999	14	16	1 (Yes)	0
MasterCard 1 card	0092_0006	51000000	55999999	14	16	1 (Yes)	3
MasterCard 2 card	0092_0007	22210000	27209999	14	16	1 (Yes)	3

Parameter Name	DFS Data Index	Start BIN Range	End BIN Range	Min PAN Length	Max PAN Length	Encrypt	Repair Invalid Tracks
VISA card	0092_0008	40000000	49999999	13	16	1 (Yes)	3
AMEX – Range 1	0092_0009	34000000	34999999	14	15	1 (Yes)	0
AMEX – Range 2	0092_0010	37000000	37999999	14	15	1 (Yes)	0
JCB	0092_0011	35280000	35899999	14	16	1 (Yes)	0
All other cards	0092_0012	00000000	99999999	10	20	0 (No)	0

The Repair Invalid Tracks flag accepts the following values:

- 0 = Do not attempt to repair invalid track data; return as-is
- 1 = Repair track 1 data only
- 2 = Repair track 2 data only
- 3 = Repair either or both tracks 1 and 2 as needed

Additional BIN ranges can be specified using 0092_0013 through 0092_0081. When assigning additional BIN ranges to these values, follow the same format as the preceding `secbin.dat` parameters.

5.2.20 Security Parameters (security.dat)

This section describes the security parameters in the `security.dat` file.

Security Parameters in the security.dat File

Parameter Name	DFS Data Index	Default Value	Description
Enable Track Data Encryption	0091_0001	0	Encrypt cardholder data sent in messages. • 0 = Disable • 1 = Magtek MagneSafe™ POS • 2 = On-Guard (IngeCrypt) - see note 1 • 3 = EPS • 4 = Voltage TEP1 (Cannot be used with TailGate) • 5 = Voltage TEP2 (Cannot be used with TailGate) • 6 = Voltage TEP4 • 7 = Monetra CardShield • 8 = Mercury Payment Systems (MPS) • 9 = RSA-OAEP • 10 = TransArmor • 11 = TDES DUKPT Generic • 12 = S1 • 14 = TDES DUKPT for NCR/Retalix • 15 = Voltage TEP1x • 16 = Voltage TEP2x **Note 1** To set the value to 2 for On-Guard, a separate `E2ECFG.PGN` file must be loaded to the terminal; it cannot be updated with `security.PGZ`.
Track Data Encryption Key Index	0091_0002	4	Encryption key index for encrypting track data in messages. Applies to all P2PE types with injected keys.

Parameter Name	DFS Data Index	Default Value	Description
Unmasked leading digits	0091_0003	6	Number of leading digits not masked (displayed in the clear) when an encryption method is set to encrypt the account number. Applies to all P2PE types except RSA-OAEP and TransArmor. • Maximum = 6.
Unmasked trailing digits	0091_0004	4	Number of trailing digits not masked (displayed in the clear) when an encryption method is set to encrypt the account number. Applies to all P2PE types except RSA-OAEP and TransArmor. • Maximum = 4.
Max Number of Transactions with Same Key	0091_0005	0	Maximum number of transactions with the same key. This parameter is mutually exclusive of parameter 0091_0006, which has precedence. Applies only to Voltage encryption types. • 0 = Do not change keys based on transaction count • 1 -65000 = Change key after this many transactions with the same key

Parameter Name	DFS Data Index	Default Value	Description
Periodically Change Keys	0091_0006	0	Periodically change keys (Requires setting the terminal date and time). This parameter has precedence over parameters 0091_0005 and 0091_0007. All letters must be entered in UPPER CASE. Applies only to Voltage encryption types. • 0 = Disabled • D = Daily • SU = Change every Sunday • MO = Change every Monday • TU = Change every Tuesday • WE = Change every Wednesday • TH = Change every Thursday • FR = Change every Friday • SA = Change every Saturday • 01-31 = Change on the XX day of the month
Preserve Keys During Power Failure	0091_0007	0	Preserve keys during power failure. This parameter defaults to 1 if parameter 0091_0006 is not 0. Applies only to Voltage encryption types. • 0 = A new key is generated at power up • 1 = Keys are saved when generated and restored at power up

Parameter Name	DFS Data Index	Default Value	Description
Append ETB to Authorization Request (50.x) Message	0091_0008	0	Append Encryption Transmission Block (ETB) to the 50.x Authorization Request message. Applies only to Voltage encryption types. • 0 = Do not append. • 1 = Append.
Identity String	0091_0009	id@sample.com	Identity String provided by authorizer. Applies only to Voltage encryption types. id@sample.com is sample data and not for production.
Identity State	0091_0010	*	Identity State. Applies only to Voltage encryption types. • Use format MMDDYYYY. • If set to * the devices current date will be used. Be sure to set the date and time via the 28.x Set Variable Request message.
Parameter Data Encoded in base64	0091_0011		Parameter data must be obtained from your authorizer. Length varies based on the encryption type enabled by 0091_0001. Applies only to Voltage encryption types.

Parameter Name	DFS Data Index	Default Value	Description
Masking Character	0091_0012	0	Defines the character to use for masking the PAN. For 0091_0001 only 0 (zero) and * (asterisk) are valid. Applies to all P2PE types except RSA-OAEP and TransArmor. The default value of 0 is hard-coded for RSA-OAEP/TransArmor.
Public Key encoded in Base64 for RSA-OAEP and TransArmor	0091_0013	(392-character string)	Contains either: • An RSA public key encoded in Base64 • The filename of a PEM file containing an RSA public key. The PEM file must be loaded to the terminal in a signed PGZ file. If containing a filename, the setting for 0091_0014 is ignored. The application supports the following key lengths: • 1024 bits • 2048 bits • 3072 bits Otherwise, this parameter is made up of a 2048-bit modulus and an exponent (normally 65537). This Public Key should be encoded in ASN.1 Base64 format, which will result in a 392-character string value for this parameter.

Parameter Name	DFS Data Index	Default Value	Description
Exponent Value for RSA-OAEP/ TransArmor	0091_0014	010001	Applies to RSA-OAEP and TransArmor. Specify this value as the default value = 010001. Ignored if 0091_0013 specifies a PEM filename. • Overrides the exponent value from the public key in parameter 0091_0013. • Is in hexadecimal format and should be set to the default value (where 010001 hex = 65537 decimal) • Might need to be changed. Check with your key authority to confirm.
TransArmor Key ID	0091_0015	12345678901	Identifies the public key. Its length should be 11-bytes.

Parameter Name	DFS Data Index	Default Value	Description
TransArmor Encryption Target	0091_0016	2	Selects the preferred encryption target: • 1 = Use Track 1 for encryption. • 2 = Use Track 2 for encryption. • 3 = Reserved for manual entry. Do not use! • 4 = Track 1 is preferred. ◦ Use Track 2 if Track 1 unavailable. • 5 = Track 2 is preferred. ◦ Use Track 1 if Track 2 unavailable. This parameter is ignored for RSA-OAEP encryption.
Length of encrypted CVV returned by Voltage	0091_0017	8	Defines the length of the encrypted CVV for Voltage TEP1 and TEP2 encryption types for manual entry only: • Valid lengths are 7 – 23 digits. If an invalid length is assigned, the length will revert to the default value of eight digits.

Parameter Name	DFS Data Index	Default Value	Description
Terminal Authentication	0091_0018	0	Terminal communication authentication selection. - 0 = Disabled. - No additional application-layer authentication. - 1 = On-connection. - Authentication required after the device connects (e.g., Bluetooth connect). Refer to 93.x Terminal Authentication Messages for a description of the device communication authentication process.
Generic Message Encryption	0091_0019	9	Optional encryption type for barcode data and/or keypad input data. Only RSA is supported. The key information for this encryption is specified in parameters 0091_0020 through 0091_0022. - 0 = Disable. - 9 = RSA encryption.

Parameter Name	DFS Data Index	Default Value	Description
Parameter Data Encoded in Base64	0091_0020	(392-character string)	Applies to RSA only. This parameter contains either: • An RSA public key encoded in Base64 • The filename of a PEM file containing an RSA public key. The PEM file must be loaded to the terminal in a signed PGZ file. If containing a filename, the setting for 0091_0021 is ignored. The application supports the following key lengths: • 1024 bits • 2048 bits • 3072 bits
Exponent for Message Encryption Key	0091_0021	010001	Applies to RSA only. This value is the hexadecimal equivalent of 65537. Do not change this value. It is ignored if 0091_0020 specifies a PEM filename.
Public Key ID	0091_0022	12345678901	Applies to RSA only. This identifies the message encryption key to the decryption system. Its length should be 11 bytes.

<table>
<tr><th>Parameter Name</th><th>DFS Data Index</th><th>Default Value</th><th>Description</th></tr>
<tr><td>RBA Escape Sequence</td><td>0091_0023</td><td>GYRYG</td><td>Supported by iUN terminals only. The sequence of keys used to exit the application from the boot screen. The sequence of keys must be exactly five characters. If an invalid sequence is entered, the sequence defaults to **2634G**.
<table>
<tr><th>Entry</th><th>Key Used for Entry</th></tr>
<tr><td>0 - 9</td><td>Number key</td></tr>
<tr><td>G</td><td>Green or Enter key</td></tr>
<tr><td>Y</td><td>Yellow or Clear key</td></tr>
<tr><td>R</td><td>Red or Cancel key</td></tr>
<tr><td>U</td><td>Up key</td></tr>
<tr><td>D</td><td>Down key</td></tr>
<tr><td>F</td><td>F or * key (key below 7 key)</td></tr>
<tr><td>.</td><td>. or # key (key below 9 key)</td></tr>
<tr><td>a</td><td>F1 key</td></tr>
<tr><td>b</td><td>F2 key</td></tr>
<tr><td>c</td><td>F3 key</td></tr>
<tr><td>d</td><td>F4 key</td></tr>
</table></td></tr>
<tr><td>Input Messages Encryption</td><td>0091_0026</td><td>0</td><td>Specifies whether keypad input data should be encrypted when Generic Message Encryption (parameter 0091_0019) is enabled. Input messages encryption (uses 0091_0020 key).
• 0 = Disabled.
• 1 = Enabled.</td></tr>
</table>

Parameter Name	DFS Data Index	Default Value	Description
Barcode Messages Encryption	0091_0027	1	Specifies whether barcode data should be encrypted when Generic Message Encryption (parameter 0091_0019) is enabled. Barcode messages encryption (uses 0091_0020 key). • 0 = Disabled. • 1 = Enabled.
Drivers License Encryption	0091_0028	0	Applies to RSA and TransArmor only. Drivers license encryption. • 0 = Disabled. • 1 = Enabled.
Non-Standard Card Encryption	0091_0029	0	Applies to all P2PE encryption types. Non-standard card encryption. • 0 = Disabled. • 1 = Enabled.
Block 12.x Account Messages when Encrypting MSR Data.	0091_0030	0	Flag used to indicate to RBA that 12.x: Account Messages are to be blocked when encrypting cardholder data. • 0 = Allow 12.x messages. • 1 = Ignore 12.x messages.
EMV Configuration XML File Type	0091_0031	0	This parameter determines the XML file type for EMV configuration. • 0 = XML files are unsigned and stored in the HOST directory. • 1 = XML files are signed and stored in the System directory.

Parameter Name	DFS Data Index	Default Value	Description
Public Key for Signature Verification	0091_0032		RSA-OAEP public key file containing the public key to be used for signature verification of encryption public keys for dynamic update of RSA-OAEP keys. If this parameter is set, then parameters 0091_0013 and 0091_0014 will be ignored. • The full file name must be specified here. • The file must exist in the RBA application directory.
Public Key for Data Encryption	0091_0033		RSA-OAEP public key file containing the public key to be used for data encryption. The full file name without path must be specified here. • This parameter is not configurable. • It is set during application execution to preserve the current encryption configuration across reboots. This parameter is directly set using the 90.7 Select RSA-OAEP Public Key Request Message.
SSL Protocol Version Identifier	0091_0034	1	SSL protocol version identifier. This setting is checked when a customer has enabled SSL on the terminal and uploaded the correct `server.pgz` file. • 0 = TLS version 1.1 • 1 = TLS version 1.2 RBA is no longer supporting SSLv3.

Parameter Name	DFS Data Index	Default Value	Description
Target Track to Encrypt	0091_0035	2	Indicates the target track to encrypt for TDES DUKPT encryption for NCR/Retalix. • 1 = Track 1. • 2 = Track 2. • 3 = Track 3. • 4 = All available tracks.
Security Application Protection	0091_0036	0	Enable/disable security application protection. • 0 = disabled • 1 = enabled
TransArmor include sentinels	0091_0037	0	Applies to TransArmor only. Specifies if encrypted tracks should contain sentinels. • 0 = include sentinels • 1 = dont include sentinels
Ncr/Retalix Encryption Suppress Data	0091_0038	1	TDES DUKPT NCR/RETALIX encryption suppress data. If enabled the following data will be suppressed from encrypted for MSR transactions • Track 1 and Track 3 • Cardholder Name Values: • 0 = disabled • 1 = enabled

Parameter Name	DFS Data Index	Default Value	Description
Masking of manual PAN Entry, Expiry date and CVV	0091_0039	0	Applies to all P2PE encryption types. Enable or disable masking of PAN, Expiry Date and CVV for manual entry. • 0 = Disabled (all) • 1 = Enabled (all) • 14 = Mask manual PAN entry. • 15 = Mask expiry date • 16 = Mask CVV Masking is no longer dependent on parameters 0005_0002 or 0091_0001. For unmasked values, RBA uses specifiers: • 11 for PAN • 10 for Expiry date • 12 for CVV
	0091_0040	0	Reserved
Maximum TDES transactions with the same key	0091_0041	0	0=Default, key changes after 10 transactions; 1-9=Change key after this many transactions with the same key

Parameter Name	DFS Data Index	Default Value	Description
Use AES encryption with Diffie-Helman key exchange	0091_0042	0	Values: • 0 = Do not encrypt • 1 = Use AES encryption with Diffie-Helman key exchange If enabled, the terminal: • Only accepts the following messages until AES encryption is started: ◦ 07.x Unit Data Request ◦ 08.x Health Stat ◦ 22.x Application ID Request ◦ 62.x File Write ◦ 97.x Reboot ◦ 90.8 messages to initiate AES encryption (see Using 90.x P2PE Data Messages with AES Encryption) • Responds to all other messages received with a 90.839 message, indicating the terminal is waiting on a key exchange. If the terminal sends a 90.839 message: ◦ Stop processing encrypted messages. ◦ Restart AES key exchange.
	0091_0043		Reserved
Hash algorithm for RSA-OAEP	0091_0044	0	Applies to RSA-OAEP and TransArmor encryption; 0=SHA-1; 1=SHA-256

If your organization requires changes to the `security.dat` file, obtain approval and signature from Ingenico prior to your implementation. See the section, Signing Requirements for .DAT File Changes.

Use the 61.x Configuration Read Message to ensure your changes to this file are applied correctly.

This policy applies to all P2PE encryption methods.

5.2.21 Signature Items (sig.dat)

Signature capture is available on the following Ingenico terminals:

- iSC250
- iSC350
- iSC480

This section describes the parameters used to configure signature capture options. These parameters are found in the `config.dfs` file under the heading, Signature, and filename `sig.dat`. Refer to the following table for a description of the signature capture parameters.

sig.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
Max Time Allowed for Signature	0009_0001	0	Specifies how long to wait for the customer to complete a signature before timing out. The time count starts with the first screen touch. • 0 = Timeout disabled • 1 - 65,000 = Timeout in 1/10 of a second
Send Message When Signature Ready	0009_0002	0	Specifies whether the terminal will send a message to the cash register when the customer completes a signature. The signature is then available for download. • 0 = Do not send a message; the POS must poll to determine when a signature is available. • 1 = Send 20.x Signature Message (On-Demand) signature request to the POS.

Parameter Name	DFS Data Index	Default Value	Description
Max Time to Allow Before Signing (pen down timeout)	0009_0004	0	Specifies how long to wait for a customer to begin the signature before timing out. • 0 = Timeout disabled • 1 - 65,000 = Timeout in 1/10 of a second
Signature encoding Format	0009_0005	9	This parameter determines the signature encoding. • 9 = Three-byte ASCII • 11 = Ingenico iSCxxx terminals will encode the signature data in the Legacy format and encode using base64. • 12 = Enhanced L format
Save State on Signature Capture Request (20.x message)	0009_0006	0	Specifies what state the terminal returns to based on the set value: • 0 = Terminate transaction • 1 = Return to the state the terminal was in when the message was received. See also sections On-Demand Transaction Process, Communication Messages, and 20.x: Signature RESPONSE Message (On-Demand) for additional details.
Minimum Acceptable Signature Size	0009_0007	50 bytes	Specifies the smallest acceptable signature in bytes. If a signature is below this threshold, it is deleted, and a new signature is requested.

Parameter Name	DFS Data Index	Default Value	Description
Display Signature Until Download	0009_0008	0	Displays signature until download starts. • 0 = Disable • 1 = Enable
Number of Bytes in Signature Block	0009_0012	200	Specifies the number of bytes in the signature block. • 1 - 1,000 = Number of bytes
Automatic Signature Acceptance Timeout	0009_0013	0	Specifies amount of time to wait for the customer to press OK after signing before automatically accepting the signature. • 0 = Disabled • 1 - 65,000 = Timeout in 1/10 of a second
Maximum Allowable Signature Size	0009_0014	750	Maximum signature size in bytes. Cannot exceed maximum value of 0009_0012 times 10 signature blocks, rounded down to closest amount divisible by 3.
Message Response Type	0009_0015	0	Message response to send when the signature is ready. Applies only when the Send message when signature ready (0010_0002) is set to 1. • 0 = No message body • 1 = 20.0x where x is the number is signature blocks.
Scaling Signature Factor	0009_0016	0	Scaling signature factor. 3 Byte ASCII SIG_BIN_2 signature format only. Accepts values 0 - 5.

5.2.22 Status Messages (status.dat)

This section describes the parameters used to configure the text portion of the response to a 11.x: Status Message request message. The response is used by the host and may be displayed on the POS screen. These parameters are located in the config.dfs file under the heading "Text Portion of Status Messages," and file name status.dat.

Important Notice
The DFS data indexes for status.dat have been changed to accommodate increasing BIN table entries. Any code that reads/writes status.dat parameters must be updated to use the new '0097_xxxx' DFS data indexes associated with status.dat. It is not necessary to generate a new status.dat file, although config.dfs files should be updated with the new DFS data indexes for documentation purposes.

Status.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
Lane Closed	0097_0001	"Lane Closed"	Closed.
Slide Card	0097_0002	"Slide Card"	Slide Card.
Processing Account Number	0097_0003	"Processing..."	Processing STB.
Select Payment	0097_0004	"Select Payment"	Select Payment.
Enter PIN	0097_0005	"Enter PIN"	Enter PIN.
Cash Back	0097_0006	"Cash back"	Cash Back.
Please Wait	0097_0007	"Please Wait"	Wait for amount.
Void OK?	0097_0008	"Void OK?"	Void OK.
Return OK?	0097_0009	"Return OK?"	Return OK,
Void Return OK?	0097_0010	"Void Return OK?"	Void Return OK.
Amount OK?	0097_0011	"Amount OK?"	Amount OK.
Processing Authorization	0097_0012	"Processing …"	Authorizing.
Please Sign	0097_0013	"Please Sign"	Sign.
Signature Accepted	0097_0014	"Signature Accepted"	Signature complete.
Error	0097_0015	"Error"	Error.
Input	0097_0016	"Input"	Input.
Input Accepted	0097_0017	"Input Accepted"	Input done.
Select Language	0097_0018	"Select Language"	Select language.

Parameter Name	DFS Data Index	Default Value	Description
Advertising	0097_0019	“Advertising”	Advertising via 30.x: Advertising Message (On-Demand) message.
Offline Advertising	0097_0020	“Offline Advertising”	Via the 00.x: Offline Message.
Menu	0097_0021	"Menu"	Menu.
Textbox	0097_0022	"Textbox"	Textbox.
EMV	0097_0023	"EMV"	EMV.
WIC	0097_0024	"WIC"	WIC.
SMC	0097_0025	"SMC"	SMC.
Idle	0097_0026	"Idle"	Idle.

5.2.23 Store/Lane Information (store.dat)

This section describes the parameters needed for your merchant identification information. This information is used in response to a 50.x: Authorization Request/Response pair. The source data is located in the `config.dfs` file under the heading, Store/Lane Information, and filename `store.dat`.

store.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
Acquirer BIN	0004_0001	123456	This parameter is the 6-digit acquiring bank number.
Merchant Number	0004_0002	789012345678	This parameter is the 12-digit merchant ID number.
Store Number	0004_0003	9012	This parameter is the 4-digit store ID number.
Terminal Number	0004_0004	3456	This parameter is the 4-digit terminal ID number.
Merchant Category	0004_0005	7890	This parameter is the 4-digit merchant industry classification.
Merchant County Code	0004_0006	123	This parameter is the 3-digit country code.
Merchant Zip Code	0004_0007	45678	This parameter is the 5-digit merchant Zip Code.

Parameter Name	DFS Data Index	Default Value	Description
Time Zone Difference	0004_0008	900	This parameter is the 3-digit time difference from GMT time zone.
Index Code	0004_0009	0	This parameter is the 1-digit index code, which should always be zero (0).
Terminal Serial Number	0004_0010	Blank	This parameter can override the serial number that was burned into the terminal at the time of manufacture. It is used when the repair facility sends out a new terminal to replace one that had been damaged, and the replacement terminal needs to have the same serial number as the original. It is an 8-digit serial number. Leave blank to use hardware serial number.
Message Status Code	0004_0011	@	This parameter is the 1-digit message status code.
Starting Transaction Number	0004_0012	1	This parameter is the first transaction number that will be started when the terminal boots.
TID	0004_0013	12345678	TID. Used if TransArmor is enabled.

5.2.24 User-Defined Variables (var.dat)

This section describes default values for user-defined variables.

var.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
User Variable	0014_0001 to 0014_0025	" " The default value is a single-space character.	Default values of the user-defined variables as set by the 28.x Set Variable Request.

5.2.25 WIC Parameters (wic.dat)

This section describes the WIC (Women, Infants, and Children) parameters located in the `wic.dat` file.

wic.dat Parameter

Parameter Name	DFS Data Index	Default Value	Description
WIC Support	0020_0001	0	POS supports use of WIC benefits: • 0 = The POS does not support WIC • 1 = The POS does supports WIC
Mother Key Slot	0020_0002	6	WIC Internal Key Index for Mother Key Slot.
Reverse Mother Key Slot	0020_0003	7	WIC Internal Key Index for Reverse Mother Key Slot.
SC5 Key Slot	0020_0004	8	WIC Internal Key Index for SC5 Key Slot.
SC6 Key Slot	0020_0005	9	WIC Internal Key Index for SC6 Key Slot.

5.3 Format Specifiers

This section explains the display attributes and provides examples on how to use the attributes in an FS string using the **Clear Text Entry** control.

Ingenico terminal format specifiers and their corresponding ID numbers are defined in the `SECURPROMPT.XML` files.

A format specifier allows you to customize the display of key data entered by the user during the clear text entry process. By default, the format specifier is an empty string, and numbers are displayed on the screen with no additional formatting.

FS strings contain display attributes that tell the terminal how to display the data on the screen. Display attributes are separated within the FS string by the percent sign (%).

- The percent character ('%') might be displayed as a fixed, hidden, or overwrite character by repeating it twice (e.g., "%o100%f%%").

There are two kinds of display attributes: general and specific.

- General attributes apply to the entire data-entry process.
- Specific attributes apply to one or more display positions used by the data-entry process.

The maximum number of format specifiers is 150.

5.3.1 General Attributes

General attributes follow the format:

```
%[General attribute][Data]
```

The general attributes are explained in the table below.

> MAX is the maximum number of display positions available within the data entry field.

General Attributes

Attribute Name	Description	Notes
m	Minimum characters	The `'%m'` attribute specifies the minimum number of digits to be entered by the user. The value following this attribute is interpreted as the minimum number of digits. If the `'%m'` attribute is not defined in the format specifier, the default value for the `'%m'` attribute will be used, which is zero (0). The range for this attribute is 0 – MAX. If ENTER is pressed before typing the minimum number of digits, an "invalid" beep will indicate that the entered input has not been accepted. If the number specified cannot be accommodated, the '%m' attribute is adjusted internally by the RBA.
M	Maximum characters	The `'%M'` attribute specifies the maximum number of digits to be entered by the user. The value following this attribute is interpreted as the maximum number of digits. If the `'%M'` attribute is not defined in the format specifier, the total number of specified overwrite characters (`'%o'`) in the format specifier string will be used. The range for this attribute is 1 – MAX. If digits are pressed after the maximum number has been reached, an "invalid" beep indicates that those digits will not be displayed on the screen or recorded. If the number specified cannot be accommodated, the '%M' attribute is adjusted internally.
p	Password protection character	The `'%p'` attribute password-protects and changes the appearance of all characters entered by the user. The ASCII character following this attribute is interpreted as the password character and is displayed on the screen in place of the characters entered by the user. For example, when the user enters a PIN, the `'%p'` attribute can specify that asterisks appear on the screen instead of numbers.

Attribute Name	Description	Notes
P	Password protection delay	The `'%P'` attribute password-protects and changes the appearance of all characters entered by the user on a delayed basis. The ASCII character following this attribute is interpreted as the password character and is displayed on the screen in place of the characters entered by the user as the next character is entered or after one second of time passes. In other words, you will see the last character entered for up to one second. As an example, when entering a PIN of `'1 2 3 4'`, if the `'%P'` attribute has been specified that asterisks appear on the screen instead of numbers, an asterisk will replace the number 1 as the number 2 is entered, an asterisk will replace the number 2 as the number 3 is entered, etc. As an additional precaution, if a number 1 is entered and one second of time passes without another number also being entered, the number just entered will be replaced by an asterisk irrespective of the '%P' setting.
z	Leading zeroes recognition	The '%z' attribute forces UIA to recognize leading zeros, which are otherwise ignored by default.

5.3.2 Specific Attributes

Specific attributes follow the format:

```
%[Specific attribute][Display string]
```

The display string is what will appear on the terminal screen. The following table describes the specific attributes.

Specific Attributes

Specific Attribute	Description	Notes
f	Fixed characters	The `‘%f’` attribute defines the corresponding positions to be displayed at all times. The `‘%f’` attribute cannot be modified during the data entry process.
h	Hidden characters	The `‘%h’` attribute causes the specific display positions to show only when each hidden character from the right is passed by the shifting text (text being entered by user). From that moment on, these positions are fixed and cannot be modified for the rest of the data entry process.

Specific Attribute	Description	Notes
o	Overwriting characters	The `‘%o’` attribute defines the corresponding positions to be displayed at the beginning of the data entry process, but allows shifting text to overwrite them. If the total number of specified overwrite characters (`‘%o’`) in the format specifier string is less than the maximum number of digits that the user can enter, then an additional number of overwrite characters (equal to the difference in the total number of specified overwrite characters and the maximum number of digits) are added as blank spaces into the format specifier. Any additional white space overwrite characters are appended after all other overwrite/fixed characters.
s	Shifting characters	The `‘%s’` attribute defines the corresponding positions to be displayed at the beginning of the data entry process, and then shifted one position at a time for each digit entered (or cleared) when the first one from the right is passed by shifting text. Shift characters are currently only implemented for default direction ‘%dr’ and are ignored (e.g., not included in the display) for ‘%dl’.
dl	Direction left	The ‘%dl’ attribute defines the direction in which characters are added to a text field. When ‘%dl’ is set, characters are added to the left side of the screen, and the text fills the screen from left to right.
dr	Direction right	The ‘%dr’ attribute defines the direction in which characters are added to a text field. When ‘%dr’ is set, characters are added to the right side of the screen, and the text fills the screen from right to left. `‘%dr’` is set by default if neither `‘%dl’` nor `‘%dr’` is specified in the format specifier string.

5.3.3 Using Multiple Format Specifier Attributes

If more than one of each of the following format specifier attributes are present in the format specifier string, then the last (i.e., first from the right end of the format specifier) of each specific attribute is used and all other identical attributes are ignored:

- `‘%m’` minimum number of digits to enter

- ‘%M’ maximum number of digits to enter
- ‘%d’ direction to enter digits

If more than one of each of the following format specifier attributes is present in the format specifier string, then the first (e.g., from the left end of the format specifier) of each specific attribute is used and all other identical attributes are ignored:

- ‘%s’ shift characters

5.3.4 Unknown Format Specifiers

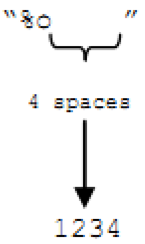
Unknown format specifier characters are ignored. In the following examples, the format specifier characters enclosed in top and bottom asterisks are ignored:

Ignored Specifiers

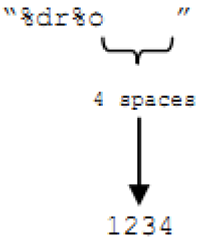
Format Specifier Key	Description
******* " 123% %o %f.%o " *******	Initial unknown characters before first known format specifier attribute are ignored
******* " 123%% %o %f.%o " *******	Initial unknown characters before first known format specifier attribute are ignored including escaped delimiter character
**** "%o %q% %f " ****	All characters following an unknown format specifier attribute but before next known format specifier attribute are ignored
**** "%o %q%%%f " ****	All characters following an unknown format specifier attribute, but before next known format specifier attribute, are ignored, including escaped delimiter character

When a format specifier or prompt is unavailable, an text box is displayed indicating an error. When RBA encounters a missing format specifier, the message “missing specifier” is displayed x. A missing format specifier indicates the format specifier was invalid. Examples of invalid format specifiers are non-numeric (like “%fhello”) or negative minimum or maximum values. Another example is “%m%M4”, where no value is provided to the minimum attribute.

5.3.5 Examples of Format Specifiers

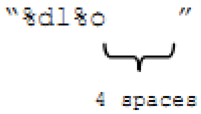


or



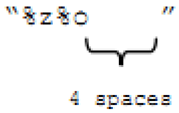
Input Example for Overwrite Format Specifier (see Note)

Key Pressed	Display	Explanation
(none)	“ ”	
Clear	“ ”	Long/bad beep because no digits are entered to clear.
0	“ ”	Long/bad beep because leading zeros not specified (i.e., no ‘%z’).
1	“ 1”	
2	“ 12”	
3	“ 123”	
4	“1234”	
5	“1234”	Long/bad beep because maximum digits entered.
Clear	“ 123”	
5	“1235”	
Clear	“ 123”	
Clear	“ 12”	
Clear	“ 1”	
Clear	“ ”	
Clear	“ ”	Long/bad beep because no digits remaining to clear.

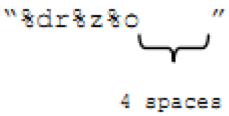


Input Example for Overwrite Format Specifier (Direction Left)

Key Pressed	Display	Explanation
(none)	“ ”	
Clear	“ ”	Long/bad beep because no digits are entered to clear.
1	“1 ”	
2	“12 ”	
3	“123 ”	
4	“1234”	
5	“1234”	Long/bad beep because maximum digits entered.

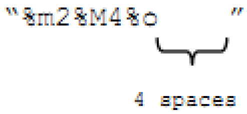


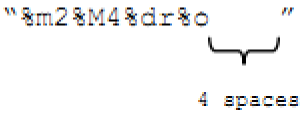
or



Input Example for Overwrite and Leading Zeroes Recognition Format Specifiers (see Note)

Key Pressed	Display	Explanation
(none)	“ ”	
Clear	“ ”	Long/bad beep because no digits are entered to clear.
0	“ 0”	
1	“ 01”	
2	“ 012”	
3	“0123”	
4	“0123”	Long/bad beep because maximum digits entered.

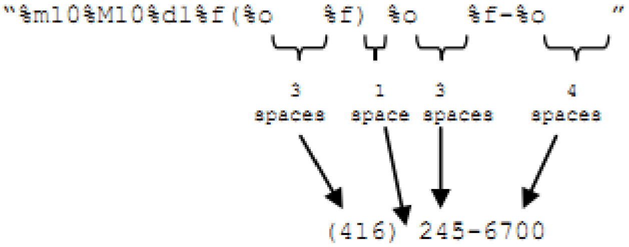




or

Input Example for Overwrite, Maximum, and Minimum Characters Format Specifiers (see Note)

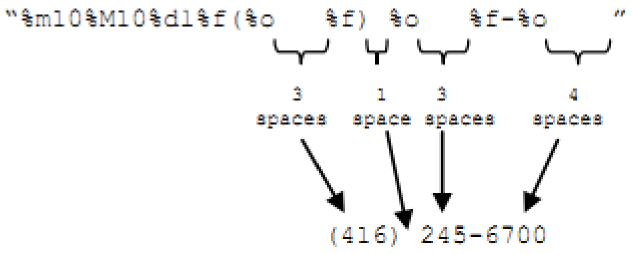
Key Pressed	Display	Explanation
(none)	“ ”	
Clear	“ ”	Long/bad beep because no digits are entered to clear.
Enter	“ ”	
1	“ 1”	
Enter	“ 1”	Silently ignored because minimum digits not entered.
2	“ 12”	
3	“ 123”	
4	“1234”	
5	“1234”	Long/bad beep because maximum digits entered.
Clear	“ 123”	
Enter	“ 123”	Silently accepted because minimum digits entered.



Using Specifiers to Prompt for a US Phone Number

Key Pressed	Display	Explanation
(none)	“( ) - ”	

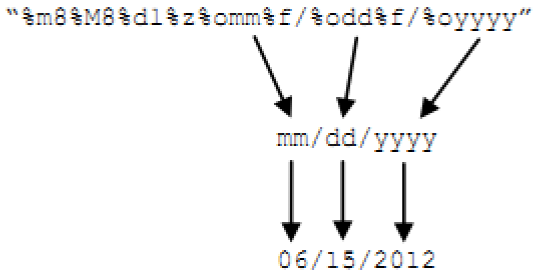
Key Pressed	Display	Explanation
Clear	“( ) - ”	Long/bad beep because no digits are entered to clear.
4	“(4) - ”	
1	“(41) - ”	
6	“(416) - ”	
2	“(416) 2 - ”	
4	“(416) 24 - ”	
5	“(416) 245- ”	
Enter	“(416) 245- ”	Silently ignored because minimum digits not entered.
6	“(416) 245-6 ”	
7	“(416) 245-67 ”	
0	“(416) 245-670 ”	
0	“(416) 245-6700”	
1	“(416) 245-6700”	Long/bad beep because maximum digits entered.
Enter	“(416) 245-6700”	Silently accepted because minimum digits entered.



Using Specifiers to Prompt for a Social Security Number

Key Pressed	Display	Explanation
(none)	“ - - ”	
Clear	“ - - ”	Long/bad beep because no digits are entered to clear.

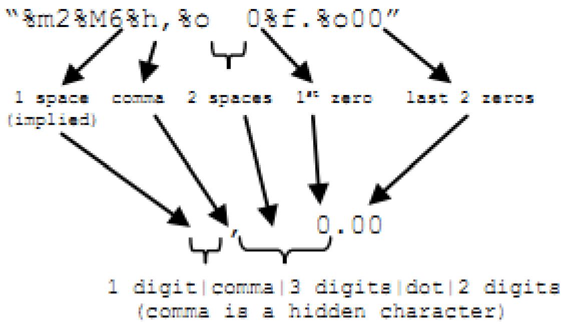
Key Pressed	Display	Explanation
1	“1 - - ”	
2	“12 - - ”	
3	“123- - ”	
4	“123-4 - ”	
5	“123-45- ”	
Enter	“123-45- ”	Silently ignored because minimum digits not entered.
6	“123-45-6 ”	
7	“123-45-67 ”	
8	“123-45-678 ”	
9	“123-45-6789”	
0	“123-45-6789”	Long/bad beep because maximum digits entered.
Enter	“123-45-6789”	Silently accepted because minimum digits entered.



Using Specifiers to Prompt for a Date

Key Pressed	Display	Explanation
(none)	“mm/dd/yyyy”	
Clear	“mm/dd/yyyy”	Long/bad beep because no digits are entered to clear.
0	“0m/dd/yyyy”	

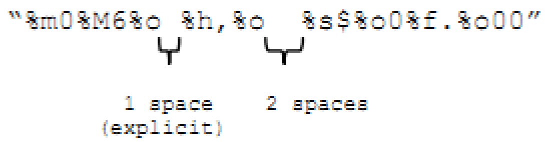
Key Pressed	Display	Explanation
6	“06/dd/yyyy”	
1	“06/1d/yyyy”	
5	“06/15/yyyy”	
2	“06/15/2yyy”	
0	“06/15/20yy”	
Enter	“06/15/20yy”	Silently ignored because minimum digits not entered.
1	“06/15/201y”	
2	“06/15/2012”	
0	“06/15/2012”	Long/bad beep because maximum digits entered.
Enter	“06/15/2012”	Silently accepted because minimum digits entered.



Using Specifiers to Prompt for a Dollar Amount

Key Pressed	Display	Explanation
(none)	“ 0.00”	
1	“ 0.01”	
2	“ 0.12”	
3	“ 1.23”	
4	“ 12.34”	

Key Pressed	Display	Explanation
5	“ 123.45”	
6	“1,234.56”	The hidden comma appears.
7	“1,234.56”	Long/bad beep because maximum digits entered.



```
“%m0%M6%o %h,%o  %s$%o0%f.%o00”
```

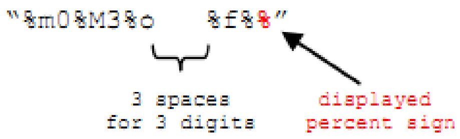
Using Specifiers to Prompt for a Dollar Amount, Method 2

Key Pressed	Display	Explanation
(none)	“ 0.00”	
5	“ 0.05”	
1	“ 0.51”	
2	“ 5.12”	
3	“ 51.23”	
9	“ 512.39”	
7	“5,123.97”	The hidden comma appears.
6	“5,123.97”	Long/bad beep because maximum digits entered.

Note
Including the Direction Right specifier invokes the same behavior as default settings, and so the two given examples for the same sample scenario behave identically.

The following format specifiers have been updated/corrected:

Updated/Corrected Format Specifiers

Format Specifier Key	Description
`“%dl%o” / “%dl%z%o”`	Zero digits may be entered for these two format specifiers, unless sent down in a 21.x message with the ‘%m’ attribute set.
 "%m0%M3%o %f%%" 3 spaces for 3 digits displayed percent sign	Three (3) digits may be entered to the left of the displayed percent sign.

5.3.6 Clear-Text Key Press Input Support

On iUN terminals, clear-text input key presses may be sent to the POS when RBA variable `805` is enabled, as illustrated in the following table:

Enabling Clear-Text Input Key Presses

Variable 805 Value	Description
0	Clear-text input keypress events and messages are disabled.
1	Clear-text input keypress events and messages are enabled. Input using asterisk (*) and pound (#) keys is supported.
2	Clear-text input keypress events and messages are enabled. Input using asterisk (*) and pound (#) keys is not supported and can cause errors.

Asterisk (*) and pound (#) keys are also supported if a form independently enables them, even if variable `805` is not set to '1'.

At startup, RBA variable `805` is not defined (blank), but behaves as though it is disabled `(805` = '0').

When enabled, key press events are returned in the '21.A' Numeric Input Request (On-Demand) messages. The character following the 'A' in the '21.A' message will match the key pressed, as described in the below table:

21.x Request per Clear-Text Key Press

Key Pressed	21.x Request Sent	Notes
<0>	`21.A0`	
<1>	`21.A1`	
<2>	`21.A2`	

Key Pressed	21.x Request Sent	Notes
<3>	`21.A3`	
<4>	`21.A4`	
<5>	`21.A5`	
<6>	`21.A6`	
<7>	`21.A7`	
<8>	`21.A8`	
<9>	`21.A9`	
<*>	`21.A*`	Only when enabled.
<#>	`21.A#`	Only when enabled.
<CLEAR>	`21.A=`	
<ENTER>	`21.0[input]`	[input] = Clear text input, including possible `NULL` input (clear text input left empty).
<CANCEL>	`21.1`	

5.4 Prompts

RBA has the ability to display prompts in up to three languages. Prompts are stored in the files `PROMPT.xml`, `SECURPROMPT.xml`, `CUSTPROMPT.xml`, and `TC1.xml`. Each prompt is assigned a number, which is then used by forms that need to reference that prompt. For example, the text element in the form `swipe.K3Z` contains the text "`&lt;?ivPROMPT3?&gt;`". This instructs the RBA to load Prompt 3 from the current language's prompt file. Prompt 3 should, in the proper language, instruct the customer to swipe a card.

To comply with PCI-DSS requirements, `PROMPT.xml` and `CUSTPROMPT.xml` are subject to security restrictions that prohibit the display of prompts containing character combinations representing the words "PIN" and "NIP."

Info

Updates to any of the prompts' `*.XML` files will only take effect after the terminal is rebooted by a 97.x message sent by the POS.

The maximum number of prompts is 400 and PIN prompts are limited to 50.

5.4.1 Line Breaks in Prompts

When composing messages for the terminal to display, only certain line break formats will be recognized and function correctly. Examples of both are given in the table below:

Format	Break Type	Sample Prompt in .xml File	Terminal Displays
\n	Line feed	`Please wait\nDo not remove card`	Please wait Do not remove card
&lt;br&gt	Line break	`Please wait&lt;br&gtDo not remove card`	Please wait Do not remove card

> Any extra backslashes (\) will appear in a prompt's displayed text.

5.4.2 Specifying Different Prompt Text for Different Devices

To specify a prompt to display only on a specific device use the following attributes to specify the prompt inside the <prompt .../> tag...

iSC350="..."
iPP250="..."

...and so on.

These attributes override the `message="..."` and `shortmessage="..."` attributes when the prompt is loaded on the specified device.

If the device-specific prompt attribute is not present, then the message="..." attribute will be loaded or the shortmessage="..." attribute will be loaded depending on those attributes being present and the device type.

5.4.2.1 Example

The following example displays identically on all terminals except the iWL250:

```
<Prompt id="309" message="Authorizing... Please wait" shortmessage="Authorizing\nPlease
wait" iWL250="Authorizing\nPlease wait"/>
```

The iWL250 displays "Please wait" on a new line, regardless of whether `message` or `shortmessage` would be otherwise called.

5.4.3 Custom Prompts (CUSTPROMPT.xml)

Ingenico may provide custom financial application prompts in English, Spanish and French in an optional `CUSTPROMPT.xml` file based on customer requests. Each customer's `CUSTPROMPT.xml` file is unique. The `CUSTPROMPT.xml` file must be packaged as a signed PGZ file before it can be loaded on an Ingenico terminal.

Prompt text display properties (such as font type, font color or text justification) are described by the text element used to call the prompt in a given form.

To comply with PCI-DSS requirements, `CUSTPROMPT.xml` is subject to security restrictions that prohibit the display of prompts containing character combinations representing the words “PIN” and “NIP.”

5.4.4 Security Prompts (SECURPROMPT.xml)

The `SECURPROMPT.xml` file includes prompts in English, Spanish, and French for use during cardholder input. Though it can be customized, the `SECURPROMPT.xml` file must be signed by Ingenico before being loaded on an Ingenico terminal.

Each prompts XML file is divided into sections containing several different types of prompts; for example, prompts in the PIN section are used exclusively during PIN entry.

Some prompts, such as index 14, Please enter your PIN, are called by the application by default during standard flow. Any index in the PIN section of `SECURPROMPT.xml` file can be called on demand for PIN entry.

The same rules apply to the Clear section for user input:

- These prompts are available during user clear-text input (such as entering a cashback amount or ZIP code) only
- Some prompts are called by the application during standard flow automatically, but the whole section is available on demand for clear-text entry.

Refer to `SECURPROMPT.xml` for prompt information.

5.4.5 Transaction Prompts (PROMPT.xml)

Prompts and other display strings are also found in the `PROMPT.xml` file. RBA is delivered with prompts in English, Spanish, and French.

Prompt text display properties (such as font type, font color or text justification) are described by the text element used to call the prompt in a given form.

To comply with PCI-DSS requirements, `PROMPT.xml` is subject to security restrictions that prohibit displaying prompts containing character combinations representing the words *PIN* and *NIP*. The following table provides descriptions and default values for prompts.

These prompts are called by 24.x Form Entry Request (On-Demand) and 70.x Update Form Element Message. PROMPT.xml also contains Button IDs and Images.

Prompt Descriptions and Default Values

Prompt ID	Default Value	Description
1	• Please select language • Por favor seleccione idioma • Choisissez la langue SVP	If more than one language is specified in the Language Count parameter, this prompt is displayed at the beginning of each transaction. A button for each available language is displayed for cardholder selection.

Prompt ID	Default Value	Description
2	• Processing... please wait • Procesando… favor de esperar • En traitement... Un moment SVP	Is displayed while the application uses the BIN lookup function to select the payment type for the card.
3	• Please slide card • Por favor deslice su tarjeta • Glissez la carte SVP	Prompts the cardholder to swipe a payment card.
4	• Expired card. Please use another. • Tarjeta expirada. Favor de usar otra. • Carte expirée	Is displayed when date checking is enabled, and the cardholder used a card with an expiration date before today's date. Another form of payment must be tendered to complete the transaction.
5	• Card read error. Try again. • Error de lectura de tarjeta. Intente de nuevo. • Erreur de lecture. Réessayez	Informs the cardholder that terminal could not read the card that was swiped.
6	• Please select payment type • Seleccione tipo de pago • Choisissez le type de paiement	Prompts the cardholder to select the payment type.
7	• Please wait for the cashier • Por favor espere por el cajero(a) • Attendez le cassier SVP	Is displayed when waiting for the purchase amount from the POS.
8	• Cashback correct? $&lt;?ivCASHBACK?&gt; • Cashback correcto? $&lt;?ivCASHBACK?&gt; • Retrait d'argent? $&lt;?ivCASHBACK?&gt;	Prompts the cardholder to verify the cashback amount.

Prompt ID	Default Value	Description
9	• Not a sale. Cashback cancelled. • No es una venta. Cashback cancelado. • Retrait d'argent annulé	Informs the cardholder that the selected cashback amount is ignored due to a transaction type limitation. This parameter applies to void, return, and void return transactions.
10	• Amount OK? $&lt;?ivTOTAL?&gt; • Cantidad correcta? $&lt;?ivTOTAL?&gt; • Montant OK? $&lt;?ivTOTAL?&gt;	Prompts the cardholder to verify the purchase amount.
11	• Processing... please wait • Procesando… favor de esperar • En traitement... Un moment SVP	Is displayed while the transaction is being authorized.
12	• Invalid card for payment type • Tarjeta no permitida para ese tipo de pago • Carte erronée pour ce type de paiement	Informs the cardholder that the swiped card is not an accepted form of payment.
13	• Register ivTerminal? • Register ivTerminal? • Register ivTerminal?	
14	• PIN must be 4 to 12 digits • PIN debe ser 4 a 12 dígitos • Un NIP va de 4 à 12 chiffres	Informs the cardholder that the PIN entered was not within the required four to 12 digit range.
15	• Cashback? • Cashback? • Retrait d'argent?	Asks the cardholder if he or she wants cash back.

Prompt ID	Default Value	Description
17	• Cashback limit is $&lt;?ivCB_MAX_TEXT?&gt; • Limite de cashback es $&lt;?ivCB_MAX_TEXT?&gt; • Retrait d'argent limité à $&lt;?ivCB_MAX_TEXT?&gt;	Informs the cardholder that the amount requested is greater than the maximum allowed.
18	• Invalid amount: $&lt;?ivCASHBACK?&gt; • Cantidad no permitida: $&lt;?ivCASHBACK?&gt; • Montant erroné: $&lt;?ivCASHBACK?&gt;	Informs the cardholder that the amount requested is not a multiple of the cash back increment amount (0002_0008).
20	• Amount must be less than $&lt;?ivAMOUNT? &gt; • Cantidad debe ser menos que $&lt;?ivAMOUNT?&gt; • Montant dépasse le total $&lt;?ivAMOUNT?&gt;	Is displayed if the cardholder enters a payment amount greater than the purchase amount.
21	• Approved • Approbation • Approuvé	Informs the cardholder that the transaction is approved. If the authorization message includes a prompt, the prompt from the authorization message is used.
22	• Declined • Denegado • Refusé	Informs the cardholder that the transactionis declined. If the authorization message includes a prompt, the prompt from the authorization message is used.
23	• Transaction cancelled • Transacción cancelada • Transaction Annulée	Informs the cardholder that the transaction is cancelled.
24	• Invalid payment type • Tipo de pago no permitido • Le type nul de Paiement	Informs the cardholder that the card is an invalid payment type.

Prompt ID	Default Value	Description
25	• Card not accepted • Tarjeta no aceptada • Cette carte n'est pas acceptée	Informs the cardholder that the card is not accepted.
26	• Encrypting... please wait... • Codificando… favor de esperar… • Chiffrement... patientez SVP...	Is displayed while a PIN is being encrypted after entry.
27	• CPEM test card read • Lectura de tarjeta de prueba CPEM • Lecture de la carte test CPEM	Indicates a CPEM test card read.
28	• Please select benefit type • Por favor seleccione tipo de beneficio • Choisissez le type d'allocation SVP	This is displayed while waiting for the user to select the EBT message type.
32	• Card read cancelled • Lectura de tarjeta cancelada • Lecture de la carte annulée	Is displayed when the cash register cancels the card swipe on demand.
33	• Input cancelled • Entrada cancelada • Entrée annulée	Informs the cardholder that the POS has cancelled the request to read a card.
34	• Signature cancelled • Firma cancelada • Signature annulée	Is displayed when the cash register cancels the signature request.
35	• Please show card to cashier • Por favor muestre su tarjeta al cajero(a) • Montrez carte au cassier SVP	Asks the cardholder to hand the payment card to the cashier so the signature or account number can be verified.

Prompt ID	Default Value	Description
36	• Void OK? $&lt;?ivTOTAL?&gt; • Confirma anulación? $&lt;?ivTOTAL?&gt; • Annulation OK? $&lt;?ivTOTAL?&gt;	Asks the cardholder to verify the void amount.
37	• Return OK? $&lt;?ivTOTAL?&gt; • Confirma devolución? $&lt;?ivTOTAL?&gt; • Remboursement OK? $&lt;?ivTOTAL?&gt;	Asks the cardholder to verify the return amount.
38	• Void return OK? $&lt;?ivTOTAL?&gt; • Confirma anulación de devolución? $&lt;?ivTOTAL?&gt; • Annuler remboursement OK? $&lt;?ivTOTAL?&gt;	Asks the cardholder to verify the void return amount.
42	• PLEASE INSERT YOUR SMART CARD QUICKLY OR SELECT A LANGUAGE • Spanish 42 • INSERER VOTRE CARTE RAPIDEMENT OU SELECTIONNER LANGUE	
77	• Unable to process transaction • Spanish 77 • Incapable de Traiter Transaction	
90	• Please wait... • Favor de esperar… • Un moment SVP…	Is displayed after a signature is entered, if the RBA is configured to wait for the cashier to verify the signature if '0009_0008' = 1(Display Signature Until Download) is enabled.

Prompt ID	Default Value	Description
91	• Unable to authorize • Incapaz de autorizar • Incapable d'autorise	Is displayed if the terminal is unable to send an authorization request to the POS.
92	• Signature accepted • Firma aceptada • Signature acceptée	Informs the cardholder that the signature was accepted.
93	• Input accepted • Entrada aceptada • Entrée acceptée	Informs the cardholder that the extra input just entered was accepted.
94	• Card accepted • Tarjeta aceptada • Carte acceptée	Informs the cardholder that the card swipe was accepted.
95	• Terms accepted • Condiciones aceptadas • Conditions acceptées	Confirms that the cardholder has accepted terms and conditions.
96	• Terms declined • Condiciones negadas • Conditions refusées	Confirms that the cardholder has declined terms and conditions.
97	• - More - • - Mas - • - Plus -	Is displayed to indicate there is more available.
98	• Card read error • Error de lectura de tarjeta • Erreur de lecture	Is displayed when a card read error occurs.
100	• Please insert card • Por favor, inserte la tarjeta • Insérer la carte SVP	

Prompt ID	Default Value	Description
101	• Please remove card quickly • Por favor, retire la tarjeta de forma rápida • Retirer la carte SVP	
102	• Please insert card • Por favor, inserte la tarjeta • Insérer la carte SVP	
103	• Please remove card quickly • Por favor, retire la tarjeta de forma rápida • Retirer la carte SVP	
120	• Accept • Aceptar • Approuvé	Is displayed to confirm the cardholder's acceptance.
121	• Decline • Denegar • Refusé	Is displayed to confirm the cardholder's denial.
126	• Please hand card to cashier • Por favor pase tarjeta al cajero(a) • Donnez carte au cassier SVP	Asks the cardholder to hand the payment card to the cashier. The cashier can enter the card number manually. This prompt is displayed when the terminal is unable to read the card after the specified number of allowed bad-read attempts has been reached (parameter 0003_0001 in `msr.dat`).
127	• Too many PIN entry errors • Demasiados errores de entrada de PIN • Trop d'essais - NIP erroné	Informs the cardholder that the transaction is being cancelled because the cardholder is having trouble entering a valid PIN.
128	• Invalid PIN. Please re-enter • PIN incorrecto. Marque de nuevo • NIP erroné. Réssayez	

Prompt ID	Default Value	Description
130	• Debit • Debito • Débit	Is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or cardholder screen selection.
131	• Credit • Crédito • Crédit	Is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or cardholder screen selection.
132	• EBT Cash • Efectivo EBT • Comptant - EBT	Is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or cardholder screen selection.
133	• EBT Foodstamps • Estampillas EBT • Bon alimentaire - EBT	Is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or cardholder screen selection.
134	• Store Charge • Cargo de la Tienda • Carte magasin	Is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or cardholder screen selection.
135	• Thank you for your loyalty • Gracias por su lealtad • Merci de votre fidélité	Is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or cardholder screen selection.
136	• PayPal • PayPal • PayPal	
137	• EMV • EMV • EMV	
153	• Invalid Account Number • Número de cuenta no válido • Numéro de compte non valide	

Prompt ID	Default Value	Description
154	• Invalid Date • Fecha No Permitida • Date Nulle	Is displayed when an invalid date is entered.
155	• Incorrect Password	
156	• Security Code too small • Código de Seguridad demasiado pequeño • Code de Sécurité trop petit	Is displayed when the user enters a security code that is too short.
163	• Rebooting for iOS mode...	
164	• Please ask for assistance • Por favor pida ayuda • Demandez de l'aide SVP	Is displayed when the terminal is unable to read the card after the specified number of allowed bad-read attempts is reached (parameter 0003_0001 in `msr.dat` file). The cashier can enter the card number manually.
165	• Please sign and tap Ok with pen • Por favor firme y toque OK con el lápiz • Signez avec le stylo SVP	Prompts the cardholder to enter a signature.
166	• Please slide card or Tap • Por favor deslice o toque su tarjeta • Glissez la carte SVP or TAP	Prompts the cardholder to slide or tap the card.
167	• Index key is missing • Clave de índice que falta • Clé d'index introuvable	

Prompt ID	Default Value	Description
168	• Please slide card or choose PayPal • Por favor deslice su tarjeta o elija PayPal • Glissez carte ou choisissez PayPal	
169	• Please slide card, tap or choose PayPal • Por favor deslice o toque su tarjeta o elija PayPal • Glissez carte, touchez ou choisissez PayPal	
170	• Is this amount OK? $&lt;?ivTOTAL?&gt; • ¿Importe OK? $&lt;?ivTOTAL?&gt; • Montant OK? $&lt;?ivTOTAL?&gt;	
171	• Please wait for cashier • Por favor espere por el cajero(a) • Attendez le cassier SVP	
172	• Please wait... • Favor de esperar... • Un moment SVP...	
173	• This Lane Closed • Este Carril Cerrado • Cette Voie Fermée	
174	• BT Pairing Required • Sincronización BT Necesario • Pairage BT Nécessaire	

Prompt ID	Default Value	Description
175	• BT Pairing Mode? • ¿BT Modo Sincronización? • Mode de Pairage BT?	
176	• Scan BT Pairing Code • Escanear BT código de sincronización • Scannez code d'Pairage BT	
177	• BT PIN Type? • ¿BT PIN Type? • BT PIN Type?	
178	• Invalid BT MAC/PIN • Inválido MAC/PIN BT • Invalide MAC/PIN de BT	
179	• Invalid BT MAC • Inválido MAC BT • Invalide MAC de BT	
180	• Invalid/No BT PIN • Inválido/No PIN BT • Invalide/No PIN de BT	

Prompt ID	Default Value	Description
181	• BT Name: &lt;?ivBLUETOOTHDEVICE?&gt;&lt;br&gt;&lt;br&gt;BT Pairing...&lt;br&gt;PIN: Nombre BT: &lt;?ivBLUETOOTHDEVICE?&gt;&lt;br&gt;&lt;br&gt;BT Sincronización...&lt;br&gt;PIN: &lt;?ivBLUETOOTHPIN?&gt; • BT Nom: &lt;?ivBLUETOOTHDEVICE?&gt;&lt;br&gt;&lt;br&gt;Pairage BT...&lt;br&gt;PIN: &lt;?ivBLUETOOTHPIN?&gt;	IOS pairing message.
182	• BT pairing failed • BT Sincronización fallado • Pairage BT n'a pas	
183	• Could not initiate BT pairing • No podía inicie la BT sincronización • N'a pas pu initier Pairage BT	
184	• Signature accepted. Thank you. • Spanish 184 • Signature acceptée. Merci.	

Prompt ID	Default Value	Description
185	• BT Name: &lt;?ivBLUETOOTHDEVICE?&gt;&lt;br&gt;&lt;br&gt;BT Pairing...&lt;br&gt;PIN: Nombre BT: &lt;?ivBLUETOOTHDEVICE?&gt;&lt;br&gt;&lt;br&gt;BT Sincronización...&lt;br&gt;PIN: &lt;?ivBLUETOOTHPIN?&gt; • BT Nom: &lt;?ivBLUETOOTHDEVICE?&gt;&lt;br&gt;&lt;br&gt;Pairage BT...&lt;br&gt;PIN: &lt;?ivBLUETOOTHPIN?&gt;	Standard (non-iOS) pairing message.
190	• Please sign for signature registration. • Spanish 190 • Signer pour l'enregistrement de la signature.	
191	• Please sign above for signature verification. • Spanish 191 • Signer pour vérification de la signature.	
192	• Transaction not&lt;br&gt;started • Spanish 192 • Transaction non\nDémarrée	
200	• Debit is available. Would you like to proceed with debit? • Spanish 200 • Débit est disponible. Voulez-vous procéder à débit	

Prompt ID	Default Value	Description
201	- Would you like to try your check card? - Spanish 201 - Voulez-vous essayer carte bancaire?	
202	- If this is ATM/Debit card, select Yes. - Spanish 202 - Si c'est une carte ATM/Débit, sélectionner Oui	
203	- If this is a Check card, select Yes. - Spanish 203 - Si c'est une carte Banquaire, Sélectionner Oui	
204	- Debit is available. Proceed with Debit? - Spanish 204 - Débit est disponible. Procéder à débit?	
205	- Entry timeout\nTransaction cancelled - Spanish 205 - Dépassement temp entrée\nTransaction annulée	
206	- Just received an unknown message - Spanish 206 - A message inconnue est reçu	

Prompt ID	Default Value	Description
207	• Cancelled by customer • Spanish 207 • Annulée par le client	
208	• Card read error&lt;br&gt;please try again • Spanish 208 • Erreur lecture carte\nSVP Réessayer	
209	• Invalid card type...&lt;br&gt;ask for assistance • Spanish 209 • Carte Invalide\nSVP demandez de l'assistance	
210	• Card not read&lt;br&gt;please ask for assistance • Spanish 210 • Carte non lisible\nSVP demandez de l'assistance	
211	• Please select shopping payment type • Spanish 211 • SVP sélectionnez le type de paiement d'achats	
212	• Entry timeout&lt;br&gt;Transaction cancelled • Spanish 212 • Dépassement temp entrée;\n Transaction annulée	

Pro mpt ID	Default Value	Description
213	• System Parameters • Spanish 213 • Les Paramètres du Système	
214	• Exit • Spanish 214 • Sortir	
215	• Config EFT Levels • Spanish 215 • Config des niveaux EFT	
216	• Manufacture / Device Id • Spanish 216 • Fabrication / Id périphérique	
217	• Config Host Port • Spanish 217 • Config du Port du hote	
218	• Config Test/Debug • Spanish 218 • Config de Test/Debug	
261	• Insert card, swipe quickly\nor select a language • Spanish 261 • Insérer carte, glisser rapidement\nou choisissez une langue	
268	• Please remove card • Spanish 268 • SVP Retirer Carte	

Prompt ID	Default Value	Description
270	• Select language • Spanish 270 • Choisir Langue	
271	• Select application • Spanish 271 • Choisir Application	
272	• Previous • Spanish 272 • Précédent	
273	• Next • Spanish 273 • Suivant	
274	• Confirm application %s • Spanish 274 %s • Confirmer Application %s	
275	• Last pin try • Spanish 275 • Dernier Essai NIP	
276	• PIN try limit exceeded\nRemove card • Spanish 276 • Essais NIP Dépassées\nRetirer Carte	
277	• Card removed\nTransaction cancelled • Spanish 277 • Carte Retiré\nTransaction Annulée	

Prompt ID	Default Value	Description
278	• Transaction cancelled\nRemove card • Spanish 278 • Transaction Annulée\nRetirer Carte	
280	• Authorizing\nPlease wait\nDo not remove card • Spanish 280 • Autorisation\nPatientez SVP\nRetirer pas la carte	
281	• Card blocked\nRemove card • Spanish 281 • Carte Bloquée\nRetirer Carte	
282	• System error #%d\nRemove card • Spanish 282 • Erreur du Système #%d\nRetirer Carte	
283	• Aborted\nRemove card • Spanish 283 • Annulée\nRetirer Carte	
284	• Call your bank • Spanish 284 • Appelez Votre Banque	
285	• Purchase • Spanish 285 • Achat	
286	• Refund • Spanish 286 • Remboursement	

Prompt ID	Default Value	Description
287	• Please confirm • Spanish 287 • Veuillez Confirmer	
288	• Confirmed. Thank you • Spanish 288 • Confirmé - Merci	
289	• Card problem. Remove card • Spanish 289 • Problème Carte\nRetirer Carte	
290	• Do not remove card • Spanish 290 • Ne Pas Retirer Carte	
291	• Card not supported\nRemove card • Spanish 291 • Carte Non Supportée\nRetirer Carte	
292	• Remove card • Spanish 292 • Retirer Carte	
293	• Authorization send failed • Spanish 293 • Envoie Autorisation Echoué	
294	• Authorization response timeout • Spanish 294 • Delai Réponse d'autorisation	

Prompt ID	Default Value	Description
295	- Authorization confirmation failed - Spanish 295 - Confirmation Autorisation a Echoué	
296	- Not accepted\nRemove card - Spanish 296 - Pas Accepté\nRetirer Carte	
297	- PIN OK - Spanish 297 - NIP OK	
298	- Transaction prep send failed - Spanish 298 - Prép Des Opérations Envoi a Echoué	
299	- Cash back - Spanish 299 - Remise En Argent	
300	- Incorrect PIN - Spanish 300 - NIP Incorrect	
301	- Declined - Spanish 301 - Refusé	
302	- Not approved \n Application expired - Spanish 302 - Non Approuvée\nApplication Expirée	

Prompt ID	Default Value	Description
303	• Please wait...Do not remove card • Spanish 303 • Patientez SVP...Ne Pas Retirer La Carte	
304	• Transaction\nchanged\nto\n • Spanish 304 • Opération\nModifiée\nA\n	
305	• Application not supported • Spanish 305 • Application Non Supportée	
306	• Insert card in chip reader • Spanish 306 • Insérer Carte Dans Lecteur de Puce	
307	• Tap failed insert or swipe card • Spanish 307 • Echec Lect S Contact Insérer/ Glisser Carte	
308	• Select account • Spanish 308 • Selectionnez Compte	
309	• Authorizing... Please wait • Spanish 309 • Autorisation\nPatientez SVP	
310	• System Error • Spanish 310 • Erreur du Systeme	

Prompt ID	Default Value	Description
311	• Please swipe or insert card • Spanish 311 • Glissez ou Inserer la carte SVP	
312	• Insert, Swipe or Tap Card • Spanish 312 • Inserer, Glisser ou Tapoter carte	
313	• No amount entered\nTransaction cancelled • Spanish 313 • Aucun montant entré\nTransaction annulée	
314	• Contactless transaction limit exceeded • Spanish 314 • Limite Operation Sans Contact Dépassé	
315	• Approved • Spanish 315 • Approuvée	
350	• Please wait • Espere por favor • Patientez SVP	
351	• Accept changes?\nEnter or Cancel • Aceptar cambiar?\n\nEnter o Cancel • Accepter les changement?\nEnter or Cancel	

Prompt ID	Default Value	Description
352	• Transaction complete • Transaccion\nCompeto • Transaction completée	
353	• Transaction cancelled • Transaccion\nCancelada • Transaction annulée	
354	• WIC update\ncancelled • WIC actualizar\ncancelado • MISE A JOUR WIC\nANNULEE	
355	• Transaction\naccepted • Transaccion\nCancelada • TRANSACTION\nACCEPTEE	
356	• Updating\ncard • Actualizando\ntarjeta • MISE A JOUR\nDE LA CARTE	
357	• Please insert card • Inserte tarjeta • INSERER LA CARTE	
358	• Card problem\nSee journal message • Problema en la tarjeta\nConsulte de mensajes\nen el journal • PROBLEME DE CARTE\nVOIR LE JOURNAL DES MESSAGES	
359	• Card Blocked • Spanish 359 • Carte Bloquée	

Prompt ID	Default Value	Description
360	- Application Blocked - Spanish 360 - Application Bloquée	
375	- Fallback. Please remove card.	
386	- Signature Timeout	

5.4.5.1 Button Text

Prompt	Prompt ID	Default Value
Accept	101	- “Accept” - “Aceptar” - “Approuvé”
Cashback	102	- “Cashback” - “Cashback” - “Cashback”
Clear	103	- “Clear” - “Borrar” - “Claire”
Credit	104	- “Credit” - “Crédito” - “Crédit”
Debit	105	- “Debit” - “Débito” - “Débit”
Decline	106	- “Decline” - “Negada” - “Refusé”

Prompt	Prompt ID	Default Value
EBT Cash	107	• “EBT Cash” • “Efectivo” • “Trésorerie EBT”
EBT Food	108	• “EBT Food” • “Comida” • “Alimentaire EBT”
English	109	• “English” • “English” • “English”
Enter Card	110	• “Enter Card” • “Entre Tarjeta” • “Entrer Carte”
Español	111	• “Español” • “Español” • “Español”
Francais	112	• “Francais” • “Francais” • “Francais”
No	113	• “No” • “No” • “Pas”
Ok	114	• “Ok” • “Ok” • “Ok”
Other	115	• “Other” • “Otro” • “D'autres”

Prompt	Prompt ID	Default Value
Partial Payment	116	• “Partial Payment” • “Pague Menos” • “Paiement Partiel”
Store	117	• “Store” • “Tienda” • “Magasin”
Yes	118	• “Yes” • “Si” • “Oui”

5.4.6 Terms and Conditions (TC1.xml)

The Terms and Conditions verbiage used by RBA can be found in the `TC1.xml` file. RBA is delivered with default Terms and Conditions verbiage in English, Spanish, and French.

Text display properties for the Terms and Conditions verbiage (such as font type, font color or text justification) are described by the text element used to call the text in a given form. The maximum characters allotted for Terms and Conditions is 7500.

6 Host Interface Messages

6.1 Communication Protocol Overview

The terminal connects to the POS using one of the supported communication methods. Communications are asynchronous, seven-bit ASCII. The communication between the POS and the terminal consists of:

- Link-level responses that are used to control the communications and to ensure the messages are exchanged correctly
- Messages that are used to exchange the required information

6.1.1 Link-Level Communications

The link- level responses are the ASCII-defined control characters, ACK and NAK. A link-level response is sent in response to all messages received.

- ACK is a positive acknowledgement that indicates to the station sending a message that the message was received correctly.
- NAK is a negative acknowledgement that indicates to the station sending a message that the message was received but it was incorrect. The sending station then resends the previous message.

The only other response that a station sending a message expects is a timeout while waiting for the ACK or NAK. If a message is sent and no response is received by the sending station within three seconds, this is called a timeout and the sending station resends the previous message.

Link-level responses are one-character messages as defined below:

Link-level response one byte, where:

- Link-level response is the following one-character ASCII code:
    - ACK is hexadecimal 06
    - NAK is hexadecimal 15

6.1.2 Data Message Format

The POS and terminal to exchange information via data messages. Each data message corresponds to the function requested and performed. The messages consist of the required data, enclosed in ASCII-defined control characters that indicate the start and end of the message. The sending station receives a link-level response (ACK or NAK) or a timeout response to all messages sent. The RBA can be configured such that the ACK response is delayed until a message is processed via parameter '0013_0011' in the Compatibility Flags (compat.dat) section of `config.dfs`.

The POS can send a message to the terminal at any time. On many systems, the terminal can send a message to the POS only at specific times when the POS is expecting to receive a message. Other systems allow the terminal to send a message at any time.

Systems that limit terminal messages usually expect to receive a message at the following times:

- In response to an Online Request message
- In response to a Status Request message

- Between the sending of an Amount message and receiving an Authorization Request message

The RBA general message flow contains the following message exchange sequences:

- Startup sequence: The messages between and including the Online Request and Online Response are the startup sequence. This sequence normally occurs only when the POS sales application is started and the POS is opened for sales transactions.
- Transaction sequence: The messages between and including the Amount message and the RESET message are the possible sequence of messages for each transaction. Some subset of the messages would be used for any EFT transaction.
- Shutdown sequence: The Offline message is the shutdown sequence and is sent by the POS when the sales application is stopped and the POS closed.

This section defines the messages that flow between the POS and the terminal. Some of these messages flow to and from the switch and are defined by the VISA Second Generation specification. For consistency, all messages have been defined to fit within the VISA message format.

All messages are checked by the LRC and a positive acknowledgement (ACK) response is sent to all good messages or a negative acknowledgement (NAK) to all bad messages. The maximum allowable message length is 247 bytes.

The basic format of these messages is as follows:

RBA Message Format

Message Fragment	Fragment Size
STX control character	1 byte
Message Identifier	3 bytes
....	
Message data, consisting of multiple fields	n bytes (0 to 241)
....	
ETX control character	1 byte
LRC check character	1 byte

Where:

- STX is the ASCII-defined control character, hex 02
- Message Identifier is three ASCII characters (two digits followed by a decimal) that identify the type message, such as 00.x (Offline Message)
- Message Data is the variable data defined for each message
- ETX is the ASCII-defined control character, hex 03
- LRC check character is a character generated for each message, using the data in the message, and is verified by the receiving station to ensure that the message was received correctly. The LRC is generated by exclusive-OR’ing all characters in the message except the STX but including the ETX (see following example).

This calculation is done by both the sending and receiving station. The sending station appends the LRC character to the message it is sending following the ETX character. The receiving station validates that the LRC is received at the end of the message is the same LRC that it calculated for the message it received.

The following is an example of two messages shown in both ASCII and hex, followed by the LRC calculation:

10.x Hard Reset Message with no Data

ASCII	[STX]10.[ETX][LRC]
Hex	`02 31 30 2E 03 2C`
LRC calculation	`31 01 2F 2C`

12.x Amount Message with a Balance Due of $123.89

ASCII	[STX]13.12389[ETX][LRC]
Hex	`02 31 33 2E 31 32 33 38 39 03 1E`
LRC calculation	`31 02 2C 1D 2F 1C 24 1D 1E`

6.1.3 General Message Flow

The following diagram shows the general message flows between a POS and the terminal.

General Message Flow

Sequence	Message	POS	Terminal
Begin Startup Sequence End Startup Sequence	Online Request	→	
	Online Response	←	
Begin Transaction Sequence End Transaction Sequence	Set Variable Request (receipt)	→	
	* Transaction Type (Sale, Void, Return, etc.)	→	
	* Account Message	→	
	Amount Message	→	
	* Status Request	→	
	* Status Response	←	

Sequence	Message	POS	Terminal
	Authorization Request	←	
	Authorization Response	→	
	* Get Variable Request	→	
	* Get Variable Response	←	
	Reset	→	
Shutdown Sequence	Offline	→	

> **Info**
> An asterisk (*) indicates an optional message. All of these messages will be ACKed, but that is not shown here for simplicity.

The diagram, above, contains the following message exchange sequences:

- Startup sequence: The messages between and including the Online Request and Online Response are the startup sequence. This sequence normally occurs only when the POS sales application is started and the POS is opened for sales transactions.
- Transaction sequence: The messages between and including the Amount message and the Reset message are the possible sequence of messages for each transaction. Some subset of the messages would be used for any EFT transaction.
- Shutdown sequence: The Offline message is the shutdown sequence and is sent by the POS when the sales application is stopped and the POS closed.

6.1.4 Messages Allowed during Waiting for Download

Invalid changes to configuration could cause the terminal to enter a perpetual "Waiting for download" state.

For troubleshooting, the POS can send the following messages from POS to correct invalid configuration parameters or reload applications:

- 01.x Online Message to initiate an EFT download.
    - If the 01.x message does not initiate a download, the terminal will:
        - Respond with a 00.x Offline Message including an error code.
        - Not go online.
- 07.x Unit Data Request to report status.
- 08.x Health Stat to report status.
- 11.x Status Message to report status.
- 62.x: File Write to download configuration or data files.
- 63.x Find File to check for a file.

- 97.x Reboot reboot message.

6.2 Communication Messages

This section describes communication messages, as well as how they are used, when they are executed, and their corresponding configuration settings in `config.dfs`.

Although most RBA messages are executed as part of a configurable sequence (standard flow), some messages can be executed by the host on demand, interrupting the expected operation of the application to perform the new function. The following on-demand messages are supported:

- 20.x Signature Message (On-Demand)
- 21.x Numeric Input Request Message (On-Demand)
- 23.x Card Read Request (On-Demand)
- 24.x Form Entry Request (On-Demand)
- 25.x Terms and Conditions Request (On-Demand)
- 26.x Run Script Request (On-Demand)
- 27.x Alpha Input Message (On-Demand)
- 30.x Advertising Request Message (On-Demand)
- 31.x PIN Entry Messages (On-Demand)
- 34.x Save and Restore State Messages
- 51.x Beep On-Demand Message
- 72.x Audio Play Request

6.2.1 General Rules

- On-demand messages are not nested.
- On-demand messages cannot be executed while another on-demand function is being executed. This behavior can be modified by the onDemandCancel flag in the `mainFlow.dat` file.
- A transaction resumes where it was interrupted when an on-demand function is completed.
- If an on-demand message cannot be executed, a response with a reject status is returned.

6.2.2 Terminal Behavior with Smart Card Inserted

If a smart card is inserted in the terminal when a 00.x Offline message, 01.x Online message or 10.x Hard Reset message is received, the cardholder is prompted to remove the card before the terminal resets all transaction data. The terminal displays the message, *Please remove card*, and continues to beep until the smart card is removed. The POS must wait until the terminal resets the transaction data and transitions to the offline or online state before initiating a transaction.

6.2.3 Limitations for iUC2xx Payment Terminals

The iUC2xx payment terminal features a small display and does not include a keypad or touchscreen; therefore, it does not have PIN-entry capability and no printing or bar code scanning. These terminals use on-demand messages exclusively.

The following messages are not supported by the iUC2xx:

- 01.x Online message

- 12.x Account message
- 17.x Merchant Data Write message
- 19.x BIN Lookup message
- 20.x Signature Request message
- 21.x Numeric Input Request message
- 25.x Terms and Conditions
- 31.x PIN Entry message
- 40.x Survey message
- 50.x Authorization Request/Response message
- 91.x Printer message
- 94.x and 95.x Barcode Configuration messages

> Please note that to adhere to enhanced security standards, dynamic text is not supported. The POS uses prompt indices to display plain-text messages.

6.2.4 00.x Offline Message

6.2.4.1 Overview of the 00.x Offline Message

The 00.x Offline message flows in both directions.

- The POS sends the 00.x Offline Request message to force the terminal into an offline state. The message format will always be '00.0000' when sent by the POS.
- The terminal sends the 00.x Offline Response message to the POS to indicate that it has detected a problem and has entered the offline state. In an offline state, the terminal cannot collect any customer data.

6.2.4.2 00.x Offline Response Message

The 00.x Offline Response message contains a four-character status field which indicates the reason for being offline. Refer to the following '00.x' Offline Response Message Format table for a description of these. Also refer to the following examples.

- The '2000' code may be encountered when an attempt has been made to perform a task offline which cannot be accomplished in an offline state, such as attempting to reset when offline. Resetting can only be accomplished when in online state.
- The '4000' code may be encountered when a parameter file such as a .dat file is missing.
- The '9000' code may be encountered when a key is bad or missing, or for instance, when an attempt is made to generate a Voltage key, but there is an error when generating that key.

In an offline state, the terminal clears all data and remains offline until receiving the 01.x Online Message. Upon receiving the 01.x message, the terminal attempts to go Online. If successful, it then sends an Online Response indicating it has entered the online state. If unsuccessful, an Offline Response message will be returned to the POS indicating the reason for being offline.

The terminal comes up in an offline state following a power on or a PLD (power line disturbance). The RBA will also send an Offline Response message when it encounters an error condition (such as an encryption error) during normal operation; this reply is not solicited by an Offline Request message from the POS. The terminal enters an offline state if it detects an unrecoverable error within itself, if it detects invalid message protocol from the POS, or if it receives an Offline Request message from the POS.

When the RBA receives the Offline Request message, it will only send an Offline Response message if it is already in an offline state.

Refer to the following tables which describe the '00.x' Offline Request message format and '00.x' Offline Response message format.

00.x Offline Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M00_OFFLINE Identifier – ASCII – “00.”
4	4	Decimal	iConnectEFT Constant = P00_REQ_REASON_CODE • Always '0000' for the Offline Request message.
8	1	Constant	ASCII control character – ETX
9	1	Binary	LRC check character.

00.x Offline Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M00_OFFLINE Identifier – ASCII – “00.”
4	4	Decimal	iConnectEFT Constant = P00_RES_REASON_CODE Offline reason code. • 0000 = No errors present. • 2000 = Request not valid. • 4000 = Missing parameter file. • 9000 = MSR encryption error.
8	1	Constant	ASCII control character – ETX
9	1	Binary	LRC check character.

Legacy versions of the RBA Test Application may not display the 00.x response from the terminal when sending a 00.x message to RBA.
While there is a response format designated for the 00.x message, the POS polls the terminal for its online/offline status using the 11.x Status Message.

6.2.4.3 *Message Responses in the Offline State*

The below table illustrates the messages that are functional when prompting a terminal in offline mode:

Message Availability while Offline

Messages Allowed	Example Response	11.x Status
00.x Offline Message	00.0000	11.00LaneClosed[FS]
01.x Online Message	01.00000000	11.01SlideCard[FS]
07.x Unit Data Request	07.INGNAR[FS]iSC350[FS]6115A1000000021[FS]9999[FS]9999[FS]0000[FS]0308[FS]1111[FS]0271[FS]0000[FS]0000	11.00LaneClosed[FS
08.x Health Stat	08.0047[FS]03[FS]02[FS]27[FS]0012[FS]3438[FS]iSC350[FS]6115A1000000021[FS]1111[FS]0271[FS]0308[FS]0270[FS]0000[FS]0000[FS]63896[FS]22066[FS]2010-06-29[FS]0[FS]OK[FS]Retail Base[FS]INGNAR[FS]0000	11.00LaneClosed[FS
11.x Status Message	11.00LaneClosed[FS]	11.00LaneClosed[FS
20.x Signature Message (On-Demand)	Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed[FS] then 11.12
21.x Numeric Input Request Message (On-Demand)	Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed[FS] then 11.12
22.x Application ID Request	Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed[FS]
23.x Card Read Request (On-Demand)	Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed[FS] then 11.12

Messages Allowed	Example Response	11.x Status
24.x Form Entry Request (On-Demand)	Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed[FS] then 11.12
25.x Terms and Conditions Request (On-Demand)	Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed[FS] then 11.12
26.x Run Script Request (On-Demand)	26.3	11.00LaneClosed[FS] then 11.12
27.x Alpha Input Message (On-Demand)	The alpha keyboard appears on the PIN pad before the 11.x request is run. Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.12Input[FS]).	11.00LaneClosed[FS] then 11.12
28.x Set Variable Request	For variable 202 the response would be 28.30000202.	11.00LaneClosed[FS]
29.x Get Variable Request	The response to a 29.00000251 message would be 29.20000251Retail base.	11.00LaneClosed[FS]
30.x Advertising Request Message (On-Demand)	Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed[FS] then 11.15Advertising[FS]
31.x PIN Entry Messages (On-Demand)	The alpha keyboard appears on the PIN pad before the 11.x request is run. Nothing is received in the Testing Tool (until you send a Status Request 11.x, then you get 11.12Input[FS]).	11.00LaneClosed[FS] then 11.12Input[FS]
60.x Configuration Write		
61.x Configuration Read		
62.x File Write		
63.x Find File	The response to a '63./HOST/BIN3.DAT' message would be '63/0187'.	11.00LaneClosed[FS]
64.x Delete File		
70.x Update Form Element Message		
87.x On-Guard and KME Card Read Data		

Messages Allowed	Example Response	11.x Status
90.x P2PE Data Message		
97.x Reboot		

These messages are disallowed in an offline state:

- 03.x Set Session Key Message
- 04.x Set Payment Type Request
- 09.x Card Status Message
- 10.x Hard Reset Message
- 12.x Account Message
- 13.x Amount Message
- 14.x Set Transaction Type
- 15.x Soft Reset Message
- 16.x Contactless Mode Request
- 17.x Merchant Data Write
- 18.x Non-Payment Card Message
- 19.x BIN Lookup
- 33.x EMV messages
- 34.x Save and Restore State Messages
- 40.x Survey Messages
- 41.x Card Read Message
- 50.x Authorization Request
- 51.x Beep On-Demand Message
- 52.x PayPal Preauthorization Message
- 58.x Terminal Discovery Message
- 82.x On-Guard and KME Session Key Injection Command
- 83.x On-Guard and KME Enable Message
- 85.x On-Guard and KME Non-Payment Card Message
- 86.x On-Guard and KME BIN Lookup (PIN Encouragement) Message
- 88.x On-Guard and KME Translate Encrypted Card Data Command
- 89.x On-Guard and KME Register BIN Record Command
- 91.x Print Message
- 93.x Terminal Authentication Messages
- 94.x and 95.x Barcode Configuration Messages

6.2.5 0x and 50.x Authorization Response Message Format

Both the Authorization Response and Alternate Authorization Response message formats are defined by the VISA Second Generation specification.

This message contains the transaction approval code and text, which RBA shows on the display during execution of the Transaction End process. The 0.x message is in the old format, while 50.x is in the new format. Both formats are supported by RBA.

0.x Authorization Response Message Format (Sent from POS)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	1	Constant	iConnectEFT Constant = M00_AUTHORIZATION Message Identifier – ASCII – 0
2	8	Decimal	iConnectEFT Constant = RES_PIN_PAD_SERIAL_NUM Terminal Serial Number.
10	1	Constant	Index Code (always 0).
11	4	Decimal	iConnectEFT Constant = RES_POS_TXN_NUM POS Transaction Number (from Authorization Request).
15	2	Alphanum	iConnectEFT Constant = P50_RES_RESPONSE_CODE Response Code: • A? = Approved. • NP = Re-enter PIN. • E? = Declined. • N? = Declined. All other codes are invalid and are treated as declined. The ? above may be any character.
17	6	Alphanum	iConnectEFT Constant = P50_RES_APPROVAL_CODE Approval Code.
23	6	Decimal	iConnectEFT Constant = P50_RES_TODAYS_DATE_YYMMDD Today’s Date (YYMMDD).
29	Variable	Alphanum	iConnectEFT Constant = P50_RES_PROMPT_INDEX_NUM Prompt index number.
M	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
M + 1	1	Constant	ASCII control character – ETX
M + 2	1	Binary	LRC check character.

50.x Alternate Authorization Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M50_AUTHORIZATION Message Identifier – ASCII – 50.
4	8	Decimal	iConnectEFT Constant = P50_RES_PINPAD_SER_NUM Terminal Serial Number.
12	1	Constant	Index Code (always 0).
13	4	Decimal	iConnectEFT Constant = P50_RES_POS_TXN_NUM POS Transaction Number (from Authorization Request)
17	2	Alphanum	iConnectEFT Constant = P50_RES_RESPONSE_CODE Response Code: • A? = Approved. • NP = Re-enter PIN. • E? = Declined. • N? = Declined. All other codes are invalid and are treated as declined
19	6	Alphanum	iConnectEFT Constant = P50_RES_APPROVAL_CODE Approval Code.
25	6	Decimal	iConnectEFT Constant = P50_RES_TODAYS_DATE_YYMMDD Today's Date (YYMMDD)
31	Variable	Alphanum	iConnectEFT Constant = P50_RES_PROMPT_INDEX_NUM Alpha display message (from POS) or prompt index number (from RBA). (Up to 32 characters.)

Offset	Length	Type	Description
M	1	Constant	ASCII control character – FS
M + 1	1	Constant	ASCII control character – ETX
M + 2	1	Binary	LRC check character.

6.2.6 01.x Online Message

The 01.x Online message flows in both directions as a request/response pair. An 01.x request message is sent by the POS when the terminal is being “Opened” to validate that it has the correct program load and parameter load levels and is ready to process transactions. If these conditions are met, the terminal responds with an Online response message to the POS and is then ready to collect customer data. If the terminal has detected an unrecoverable error and cannot process transactions, or the 01.x request parameter load level is zero and the terminal does not contain a parameter load, then the terminal responds with the proper Offline message and remains in the offline state.

When the terminal is ready, it responds with an Online response identifying the current program load and parameter load versions. Upon sending the Online response, the terminal enters the Slide Card state. At this point, the terminal is ready to accept card data, account selection, and PIN data.

Info
The 01.x message allows the merchant or financial institution to control the changing of the terminal program or parameters. Software versions can be controlled by maintaining a file on the store controller that contains the program load and parameter load version for each terminal.

The 01.x message data consists of two fields: program load version and parameter load version, which indicate the level of program and parameter load that is currently contained in that terminal. The terminal validates that it contains the current levels before accepting the 01.x request. If either level is incorrect, it requests a load of the pieces that are incorrect (see Scenarios, below). The incorrect level in the 01.x request that is forcing the load is assigned to be the level of the new load. Zero (0000) in either field of the 01.x request is a special case indicating to the terminal to use the level it currently contains of that piece. If the terminal does not contain any level for that piece, it returns an Offline message; otherwise, the Online response contains the current level value for that piece.

Info
Sending the 01.x message with a value other than the current EFT level (0000) will trigger a download via TDA. When this occurs, you may observe in the RBA Testing Tool execution of “VarSetByInt() code in procesOnlineReqRespMsg()”. When this occurs, RBA exits and can no longer respond to further 00.x or 01.x messages.

Format Defined

Syntax	Sample
01.XXXXYYYY	01.02081234

Version Fields

Field	Value	Sample
XXXX	Program Load version	0208
YYYY	Parameter Load version	1234

Scenarios

Terminal contains	01.x Message contains	Comparison results in…
01.02071235	01.02081234	A version change for both the Program Load and Parameter Load, resulting in '01.02081234' being loaded to the terminal.
01.02071234	01.02081234	A version change for the Program Load, resulting in '01.02081234' being loaded to the terminal.
01.02071234	01.00000000	Proceeds online and conducts no version changes. The terminal continues with '01.02071234'.

01.x Online Request Message/Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M01_ONLINE_MSG Message Identifier – ASCII – “01.”
4	4	Decimal	iConnectEFT Constant = P01_REQ_APPID for Request or P01_RES_APPID for Response Application ID (“0000” is the response)
8	4	Decimal	iConnectEFT Constant = P01_REQ_PARAMID for Request or P01_RES_PARAMID for Response Parameter ID (“0000” is the response).
12	1	Constant	ASCII control character – ETX
13	1	Binary	LRC check character.

Info
The default value is '0001'.

6.2.7 03.x Set Session Key Message

The 03.x Session Key message sends a new session encryption key to the terminal.

- The 03.x request message data consists of the new session key for data encryption.
- The 03.x response message contains a three-character status code indicating the result of the key injection.
- The only allowable message response to the 03.x request is a 03.x response.

Although this message can be sent any time, it is recommended to send it before a new transaction is started on the terminal. The new session key is stored in flash memory where it remains stored, even in the event of a power loss.

03.x Session Key Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M03_SET_SESSION_KEY Message Identifier – ASCII – 03
4	Variable	Alphanum	iConnectEFT Constant = P03_REQ_SESSION_KEY Session Key (must be between 16 and 32 characters)
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

03.x Session Key Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M03_SET_SESSION_KEY Message Identifier – ASCII – 03
4	16	Decimal	iConnectEFT Constant = P03_RES_SESSION_KEY_STATUS Decimal Status Flag: • 0 = Successful • 1 = Failed

Offset	Length	Type	Description
20	1	Constant	ASCII control character – ETX
21	1	Binary	LRC check character

6.2.8 04.x Set Payment Type Request

The 04.x Set Payment Type Request message flows from the POS to the terminal. This message is used to indicate the payment method if selected by the cashier for the customer. The Force Type parameter in the request message determines whether to always force the payment method or to only force the payment method when it has not been selected by the cardholder. Payment methods are defined in the cards.dat configuration file. A field for the transaction amount is included in this message. A 13.x Amount Message specifying the transaction amount is followed by the 04.x message.

The terminal responds to the POS with a 04.x response message indicating if the Force Type was successful, if it failed, or if the customer changed the payment method. As with the request message, a field for the transaction amount is included in this message.

04.x Set Payment Type Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant = M04_SET_PAYMENT_TYPE. Message Identifier – ASCII – "04."
4	1	Decimal	iConnectEFT Constant = P04_REQ_FORCE_PAYMENT_TYPE. Force Type. This setting determines whether the terminal should force a specific type of payment: • 0 = Unconditional – always force payment type, even if the cardholder selects an alternate payment type. • 1 = Conditional – force payment type unless the cardholder selects a payment type.

Offset	Length	Type	Description
5	1	Decimal	iConnectEFT Constant = P04_REQ_PAYMENT_TYPE. Payment Type (as defined in `cards.dat`): • A = Debit. • B = Credit. • C = EBT Cash. • D = EBT Food Stamps. • E = Store Charge. • F = Loyalty. • G = PayPal. Icon These are only the default settings in cards.dat. Payment type can be configured such that any value A – P refers to a payment type of choice.
6	Variable	Decimal	iConnectEFT Constant = P04_REQ_AMOUNT. Transaction Amount. • 3 to 9 characters. Icon If the transaction amount is zero, or if the transaction amount is less than 3 digits, then preceding zeros will be added to the amount. If the amount exceeds 9 digits, the digits in excess of 9 will be silently ignored. As examples: • 0 amount = '000' sent. • 8 amount = '008' sent. • 78 amount = '078' sent.
M	1	Constant	ASCII control character – ETX.
M + 1	1	Binary	LRC check character.

04.x Set Payment Type Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant = M04_SET_PAYMENT_TYPE. Message Identifier – ASCII – "04."
4	1	Decimal	iConnectEFT Constant = P04_RES_FORCE_PAYMENT_TYPE. Force Type: • 0 = Successful. • 1 = Failed. • 2 = Changed customer selection.
5	1	Decimal	iConnectEFT Constant = P04_RES_PAYMENT_TYPE. Payment Type (as defined in cards.dat): • A = Debit. • B = Credit. • C = EBT Cash. • D = ET Food Stamps. • E = Store Charge. • F = Loyalty. • G = PayPal. Icon These are only the default settings in cards.dat. Payment type can be configured such that any value A – P refers to a payment type of choice.

Offset	Length	Type	Description
6	Variable	Decimal	iConnectEFT Constant = P04_RES_AMOUNT. Transaction Amount. • 3 to 9 characters. Icon If the transaction amount is zero, or if the transaction amount is less than 3 digits, then preceding zeros will be added to the amount. If the amount exceeds 9 digits, the digits in excess of 9 will be silently ignored. As examples: • '0' amount – '000' sent. • '8' amount – '008' sent. • '78' amount – '078' sent.
M	1	Constant	ASCII control character – ETX.
M + 1	1	Binary	LRC check character.

6.2.9 07.x Unit Data Request

The Unit Data Request message flows from the POS to the terminal. It requests the terminal hardware and software configuration.

07.x Unit Data Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M07_UNIT_DATA Message Identifier – ASCII – 07.
4	1	Constant	ASCII control character – ETX
5	1	Binary	LRC check character.

07.x Unit Data Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M07_UNIT_DATA Message Identifier – ASCII – 07.
4	Variable	Alphanum	iConnectEFT Constant = P07_RES_MANUFACTURE Manufacture ID – INGNAR
M	1	Constant	ASCII control character – FS
M+1	Variable	Alphanum	iConnectEFT Constant = P07_RES_DEVICE Terminal ID. Depending on the setting of 0013_0023 in Compatibility Flags (compat.dat), it returns: - The device ID only, such as iPP350 or iSC250 - The full reference, such as ISC250-01T1226A
N	1	Constant	ASCII control character – FS
N+1	Variable	Alphanum	iConnectEFT Constant = P07_RES_UNIT_SERIAL_NUMBER Unit Serial Number. For iUN terminals, when 0007_0051 is set to 1, returns the serials for each component. For example: `80074079[iUP],80047988[iUR],20035018[iUC]` Otherwise, it returns only the iUP serial number. During Alert Irruption condition, ALERT is sent in place of the manufacture serial number. Otherwise, it sends the serial number.
O	1	Constant	ASCII control character – FS
O+1	Variable	Decimal	iConnectEFT Constant = P07_RES_RAM_SIZE RAM size in KB.
P	1	Constant	ASCII control character – FS
P+1	Variable	Decimal	iConnectEFT Constant = P07_RES_FLASH_SIZE Flash memory size in KB.

Offset	Length	Type	Description
Q	1	Constant	ASCII control character – FS
Q+1	4	Decimal	iConnectEFT Constant = P07_RES_DIGITIZER_VERSION Digitizer Version .
Q+5	1	Constant	ASCII control character – FS
Q+6	4	Decimal	iConnectEFT Constant = P07_RES_SECURITY_MODULE_VERSION Security Module Version.
Q+10	1	Constant	ASCII control character – FS
Q+11	Variable	Decimal	iConnectEFT Constant = P07_RES_OS_VERSION OS Version.
R	1	Constant	ASCII control character – FS
R+1	4	Decimal	iConnectEFT Constant = P07_RES_APPLICATION_VERSION Application Version, data format is XXYY.
R+5	1	Constant	ASCII control character – FS
R+6	4	Decimal	iConnectEFT Constant = P07_RES_EFTL_VERSION EFTL Version.
R+10	1	Constant	ASCII control character – FS
R+11	4	Decimal	iConnectEFT Constant = P07_RES_EFTP_VERSION EFTP Version.
R+15	1	Constant	ASCII control character – FS
R+16	Variable	Alphanum	iConnectEFT Constant = P07_RES_MOB_DEV_BATTERY_LEVEL Mobile Terminal Battery Level. • 0-100% for a mobile terminal. • N/A for a non-mobile terminal.
S	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
S+1	Variable	Alphanum	iConnectEFT Constant = P07_RES_MOB_DEV_BATTERY_CHRG_STAT Mobile Terminal Battery Charging State. • CHG = charging state. • DCHG = discharging state. • N/A = not available.
T	1	Constant	ASCII control character – FS
T+1	Variable	Alphanum	iConnectEFT Constant = P07_RES_MANUFACTURING_SERIAL_NUMBER Manufacturing Serial Number
U	1	Constant	ASCII control character – FS
U+1	4	Alphanum	iConnectEFT Constant = P07_RES_EMV_DC_KERNEL_TYPE Kernel type.
U+5	1	Constant	ASCII control character – FS
U+6	4	Alphanum	iConnectEFT Constant = P07_RES_EMV_ENGINE_KERNEL_TYPE Engine kernel type.
U+10	1	Constant	ASCII control character – FS
U+11	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_DISCOVER_KERNEL_TYPE Contactless Discover kernel type.
U+15	1	Constant	ASCII control character – FS
U+16	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_EXPRESSPAY_V3_KERNEL_TYPE Contactless ExpressPay v3 Kernel type.
U+20	1	Constant	ASCII control character – FS
U+21	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_EXPRESSPAY_V2_KERNEL_TYPE Contactless ExpressPay v2 Kernel type.
U+25	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
U+26	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_PAYPASS_V3_KERNEL_TYPE Contactless PayPass v3 kernel type.
U+30	1	Constant	ASCII control character – FS
U+32	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_PAYPASS_V3_APP_TYPE Contactless PayPass v3 application type.
U+35	1	Constant	ASCII control character – FS
U+36	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_VISA_PAYWAVE_KERNEL_TYPE Contactless VISA PayWave kernel type.
U+40	1	Constant	ASCII control character – FS
U+41	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_INTERAC_KERNEL_TYPE Contactless Interac kernel type.
U+45	1	Constant	ASCII control character – FS
U+46	Variable	Alphanum	iConnectEFT Constant = P07_RES_CLESS_INTERFACE_IS_SUPPORTED • YES = Contactless interface supported for transactions. • NONE = Contactless interface not supported for transactions.
V	1	Constant	ASCII control character – ETX
V+1	1	Binary	LRC check character.

The default configuration returns the full size of RAM and Flash memory in kilobytes. See also Compatibility Flags (compat.dat).

6.2.10 08.x Health Stat

The Health Stat message flows from the POS to the terminal. It requests health and usage statistics and identification information from the terminal.

08.x Health Stat Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M08_HEALTH_STAT Message Identifier – ASCII – 08.
4	1	Decimal	iConnectEFT Constant = P08_REQ_REQUEST_TYPE Request Type: • 0 = Retrieve Health Stats. • 1 = Reset then Retrieve Health Stats. • 2 = Battery Life Health Stats.
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character.

Info

IBMEFT downloading is not supported in this release. The following table will display the values of EFTL version and EFTP version as 0000.

Health Stat Response offsets 4 (Number of MSR swipes), offsets 9, 12 and 15 (Number of bad Track 1, 2, 3, reads), and offset 18 (Number of signature totals) reset to 0 (zero) after having sent an 08.1 reset message. All other 08.x responses remain intact for the life of the terminal.

There are two possible response messages when the Health Stat Request message is received. The response messages are generated based on the request type:

- Request type = 0 – return Health Stats (see **Health Stat Response Message Format** in first table below).
- Request type = 1 – reset and then retrieve Health Stats (refer **Health Stat Response Message Format** in first table below).
- Request type = 2 – return Battery Life Health Stats (refer **Health Stat Battery Life Response Message Format** in second table below).

08.x Health Stat Response Message Format (returned in response to **08.0 Retrieve Health Stats** message)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M08_HEALTH_STAT Message Identifier – ASCII – 08.

Offset	Length	Type	Description
4	4	Decimal	iConnectEFT Constant = P08_RES_COUNT_MSR_SWIPES Number of MSR Swipes.
8	1	Constant	ASCII control character – FS (0x1c)
9	2	Decimal	iConnectEFT Constant = P08_RES_COUNT_BAD_TRACK1_READS Number of bad Track 1 reads.
11	1	Constant	ASCII control character – FS (0x1c)
12	2	Decimal	iConnectEFT Constant = P08_RES_COUNT_BAD_TRACK2_READS Number of bad Track 2 reads.
14	1	Constant	ASCII control character – FS (0x1c)
15	2	Decimal	iConnectEFT Constant = P08_RES_COUNT_BAD_TRACK3_READS Number of bad Track 3 reads.
17	1	Constant	ASCII control character – FS (0x1c)
18	4	Decimal	iConnectEFT Constant = P08_RES_COUNT_SIGNATURES Number of signature totals.
22	1	Constant	ASCII control character – FS (0x1c)
23	4	Decimal	iConnectEFT Constant = P08_RES_COUNT_REBOOT Number of reboots.
27	1	Constant	ASCII control character – FS (0x1c)
28	Variable	Alphanum	iConnectEFT Constant = P08_RES_DEVICE_NAME Terminal Name.
M	1	Constant	ASCII control character – FS (0x1c)

Offset	Length	Type	Description
M + 1	Variable	Alphanum	iConnectEFT Constant = P08_RES_SERIAL_NUMBER Unit serial number. For iUN terminals, when 0007_0051 is set to 1, this field returns the serial numbers for each component. For example: `80074079[iUP],80047988[iUR],20035018[iUC]` Otherwise, it returns only the iUP serial number. During Alert Irruption condition, ALERT is sent in place of the manufacture serial number. Otherwise, it sends the serial number.
N	1	Constant	ASCII control character – FS (0x1c)
N + 1	Variable	Decimal	iConnectEFT Constant = P08_RES_OS_VERSION OS version.
O	1	Constant	ASCII control character – FS (0x1c)
O + 1	Variable	Decimal	iConnectEFT Constant = P08_RES_APP_VERSION Application Version, format is MMNN (where MM = Major version, NN = Minor version).
P	1	Constant	ASCII control character – FS (0x1c)
P + 1	Variable	Decimal	iConnectEFT Constant = P08_RES_SECURITY_LIB_VERSION Security library version.
Q	1	Constant	ASCII control character – FS (0x1c)
Q + 1	4	Decimal	iConnectEFT Constant = P08_RES_ TDA_VERSION TDA version.
Q + 5	1	Constant	ASCII control character – FS (0x1c)
Q + 6	4	Decimal	iConnectEFT Constant = P08_RES_EFTL_VERSION EFTL version

Offset	Length	Type	Description
Q + 10	1	Constant	ASCII control character – FS (0x1c)
Q + 11	4	Decimal	iConnectEFT Constant = P08_RES_EFTP_VERSION EFTP version .
Q + 15	1	Constant	ASCII control character – FS (0x1c)
Q + 16	Variable	Decimal	iConnectEFT Constant = P08_RES_RAM_SIZE RAM size in KB.
R	1	Constant	ASCII control character – FS (0x1c)
R + 1	Variable	Alphanum	iConnectEFT Constant = P08_RES_ FLASH_SIZE Flash memory size in KB.
S	1	Constant	ASCII control character – FS (0x1c)
S + 1	Variable	Alphanum	iConnectEFT Constant = P08_RES_MANUFACTURE_DATE Manufacture Date.
T	1	Constant	ASCII control character – FS (0x1c)
T + 1	1	Decimal	iConnectEFT Constant = P08_RES_CPEM_TYPE CPEM type. • 0 = none. • 1 = Internal.
T + 2	1	Constant	ASCII control character – FS (0x1c)

Offset	Length	Type	Description
T + 3	Variable	Alphanum	iConnectEFT Constant = P08_RES_PEN_STATUS Pen Status (see Note). • OK • FAIL • UNKNOWN • UNSUPPORTED DEVICE **Note** Currently this is not supported in Telium terminals and the field will return UNSUPPORTED_DEVICE.
U	1	Constant	ASCII control character – FS (0x1c)
U + 1	Variable	Alphanum	iConnectEFT Constant = P08_RES_APP_NAME Application Name.
W	1	Constant	ASCII control character – FS (0x1c)
W + 1	6	Alphanum	iConnectEFT Constant = P08_RES_MANUFACTURE_ID Manufacture ID – INGNAR
W + 7	1	Constant	ASCII control character – FS (0x1c)
W + 8	Variable	Constant	iConnectEFT Constant = P08_RES_DIGITIZER_VERSION Digitizer Version.
X	1	Constant	ASCII control character – FS (0x1c)
X + 1	Variable	Decimal	iConnectEFT Constant = P08_RES_MANUFACTURING_SERIAL_NUMBER Manufacturing Serial Number. For iUN terminals, when 0007_0051 is set to 1, this field will return the manufacturing serials for each component. For example, `70006383[iUP],70003635[iUC],80014816[iUR]` Otherwise, it will return only the iUPs manufacturing serial serial number.

Offset	Length	Type	Description
Y	1	Constant	ASCII control character – ETX
Y + 1	1	Binary	LRC check character.

Health Stat Battery Life Response Message Format (returned in response to **08.2 Battery Life Health Stats** message)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – 08.
4	4	Decimal	Number of times MSR was enabled.
8	1	Constant	ASCII control character – FS (0x1c)
9	4	Decimal	Total time MSR was enabled.
13	1	Constant	ASCII control character – FS (0x1c)
14	4	Decimal	Average MSR enable time.
18	1	Constant	ASCII control character – FS (0x1c)
19	4	Decimal	Number of times contactless was enabled.
23	1	Constant	ASCII control character – FS (0x1c)
24	4	Decimal	Total time contactless was enabled.
28	1	Constant	ASCII control character – FS (0x1c)
29	4	Decimal	Average contactless enable time.
33	1	Constant	ASCII control character – FS (0x1c)
34	4	Decimal	Number of times smart card was enabled.
38	1	Constant	ASCII control character – FS (0x1c)
39	4	Decimal	Total smart card enabled time.
43	1	Constant	ASCII control character – FS (0x1c)
44	4	Decimal	Average smart card enable time.
48	1	Constant	ASCII control character – FS (0x1c)
49	4	Decimal	Number of times barcode scanner was enabled.

Offset	Length	Type	Description
53	1	Constant	ASCII control character – FS (0x1c)
54	4	Decimal	Total barcode scanner enable time.
58	1	Constant	ASCII control character – FS (0x1c)
59	4	Decimal	Average barcode scanner enable time.
63	1	Constant	ASCII control character – FS (0x1c)
64	4	Decimal	Number of times Bluetooth was searching.
68	1	Constant	ASCII control character – FS (0x1c)
69	4	Decimal	Total Bluetooth search time.
73	1	Constant	ASCII control character – FS (0x1c)
74	4	Decimal	Average Bluetooth search time.
78	1	Constant	ASCII control character – FS (0x1c)
79	4	Decimal	Number of times Bluetooth was connected.
83	1	Constant	ASCII control character – FS (0x1c)
84	4	Decimal	Total time Bluetooth was connected.
88	1	Constant	ASCII control character – FS (0x1c)
89	4	Decimal	Average Bluetooth connection time.
93	1	Constant	ASCII control character – FS (0x1c)
94	4	Decimal	Number of times display backlight was enabled.
98	1	Constant	ASCII control character – FS (0x1c)
99	4	Decimal	Total time display backlight was enabled.
103	1	Constant	ASCII control character – FS (0x1c)
104	4	Decimal	Average backlight enable time.
108	1	Constant	ASCII control character – ETX
109	1	Binary	LRC check character

Although all terminals respond to the 08.2 message, only the battery-powered terminals return meaningful results. All other terminals return zero for each field.

6.2.11 09.x Card Status Message

The 09.x Card Status message flows from the terminal to the POS. This message provides information about the card such as the card type, message version, transaction type, and card status. It is primarily used for smart cards and provides card insertion status (e.g., inserted, removed), card swipe status, and card tapped status.

Anytime a WIC, EMV, or other smart card is inserted to initiate a transaction, the terminal sends the message indicating that the card has been inserted. An invalid card type may include a valid smart card that has been disabled (e.g., disabled EMV card; 0019_0001 = 0).The following table provides example 09.x Card Status messages.

Example 09.x Card Status Messages

Card Type	Action	09.x Card Status Message
EMV	Card inserted.	09.020201I
	Card removed.	09.020201R
WIC	Card inserted.	09.010201I
	Card removed.	09.010201R
Generic Smart Card	Card inserted.	09.000201I
Damaged or Invalid Card	Card inserted.	09.990201P

To avoid interrupting the EMV or WIC transaction flow, 09.x status messages conveying card insertion status (e.g., card inserted, card removed) are only sent before the transaction is started.

09.x Card Status Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – 09.

Offset	Length	Type	Description
4	2	Numeric	Card type: - 00 = Generic smart card (SMC) - 01 = WIC smart card - 02 = EMV smart card - 03 – MSR - 04 – contactless - 98 = Non-EMV card inserted (iUN/iUR devices only) - 99 = Damaged or invalid card 09.x returns status for smart cards, except for card type = 98.
6	2	Numeric	Message version/variant: - 0 = Original message version only for WIC smart cards - 1 = Current message version for WIC and EMV smart cards - 02 = Message version upgraded to two digits
8	2	Numeric	Transaction type: - 01 = Purchase
10	1	Alphanumeric	Card status: - A = No supported kernel for contactless tap - B = Bad MSR swipe (or bad card read) - C = no supported processing for contactless tap - I = SMC inserted - R = SMC removed - P = Unknown problem - S = MSR card swiped - T = Contactless card tapped
11	1	Constant	ASCII control character – ETX
12	1	Binary	LRC check character

6.2.12 09.x Set Allowed Payments Message

The 09.x Set Allowed Payment Message flows from the POS to the terminal. It is used to enable or disable the various payments configured in the terminal. The terminal supports up to sixteen payment types, which are defined in the `config.dfs` file. A payment type is represented by a flag. The array of flags may include a subset of the flags. If the message only includes five flags, the first five payment types are set. The other eleven are not changed.

09.x Set Allowed Payments Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M09_SET_ALLOWED_PAYMENT Message Identifier – ASCII – "09."
4	16	Decimal	iConnectEFT Constant = P09_REQ_ENABLE_PAYMENT_ARRAY Enable Payment Array Array of 16 characters, one for each payment type. • 0 = Disabled • 1 = Enabled Example: If the payment type field is set to '1011000000000000', payment types 1, 3 and 4 are enabled. All others are disabled.
20	1	Constant	ASCII control character – ETX
21	1	Binary	LRC check character

6.2.13 10.x Hard Reset Message

The 10.x Hard Reset message can be sent by the POS to reset the terminal or by the terminal to notify the POS that the customer has forced a reset by pressing the CANCEL key on the terminal.

The 10.x message, when sent by the POS to the terminal, forces the terminal to the beginning of the transaction to clear the previous transaction or any activity at the terminal since the previous transaction. The POS sends a 10.x message at the end of every transaction.

The 10.x message, when sent by the terminal to the POS, notifies the POS that the customer has pressed the CANCEL key, causing a reset. The terminal clears any previous actions and returns to the ready state. The customer can choose another form of payment or restart the payment . The terminal sends the 10.x message to the POS only if it has received an Amount message from the POS and not yet responded with the Authorization Request message. If those two conditions are not met, the 10.x message is not sent to the POS. Upon receiving a 10.x message, the POS sends a current Amount message to the terminal to allow the new transaction to be completed.

RBA determines when to clear the scrolling receipt based on information from two sources:

- Source #1 - the 10.x message parameter value
- Source #2 - the RBA local configuration parameter, listed in `mainFlow.dat` file, index 0007_0007, Clear line item display on reset (0 = dont clear, 1 = clear).

RBA clears the scrolling receipt from the terminal screen in the following conditions:

- If the 10.x message received from the POS does not specify whether the line display should be cleared, the display is cleared based on the RBA local configuration selection.
- If the 10.x message received from the POS includes the parameter, then the parameter included in the message is used to select the clearing method, which is:
  - 0 – Do not clear the receipt
  - 1 – Clear the receipt

If RBA receives a message other than the 10.x, which also resets the transaction, such as the 01.x Online Message, or 30.x Advertising Request Message, the receipt is cleared based on the configuration selection.

10.x Hard Reset Message Format (from POS)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M10_HARD_RESET Message Identifier – ASCII – 10.
4	0 or 1	Decimal	iConnectEFT Constant = P10_REQ_CLEAR_LINE_DISPLAY Clear Line Display (optional) • 0 = Do not clear line display • 1 = Clear line display If invalid value or if left out of message, the setting in `mainFlow.dat` (0007_0007) will be used. This field is never sent to the POS.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

If the 10.x message is sent by RBA, the cardholder has either cancelled the transaction or declined the total. In either case, a new amount message must be sent to the terminal.

If the cardholder presses the [Cancel] key on the terminal, the following 10.x message is sent:

10.x Hard Reset Message Format (from Terminal)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M10_HARD_RESET Message Identifier – ASCII – 10.
4	1	Decimal	iConnectEFT Constant = P10_RES_DESTINATION Destination (optional - only sent if 0013_0009 is set to 1). • 0 = Cancel transaction • 1 = Decline total
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character

6.2.14 11.x Status Message

6.2.14.1 Overview of the 11.x Status Message

The 11.x Status message has been architected to flow in both directions as a request/response pair. It is defined to allow the POS system to keep track of the state of the terminal. The POS sends the 11.x request to the terminal to request status. The customer input returns a 11.x response, indicating its current status.

The 11.x request from the POS contains no Message Data. The 11.x response contains a two-character state code followed by a variable field that represents the terminal's current display message.

The POS can send the 11.x request message at any time. The 11.x response message is only sent in response to a 11.x request. The allowable message response to the 11.x request is a 11.x response or a 00.x Offline message. A 00.x message is returned if the terminal is in offline state; otherwise, an 11.x response message is returned. A new "Idle" status has been added to the 11.x Status response message. When the terminal is in idle mode, this message will be returned with a '99' status value and default "Idle" text in the Text field.

6.2.14.2 Appending the Form Name to the 11.x Status Response Message

An alternate form of the 11.x Status request message has been added so that the form name can be appended to the status response message, without the ".K3Z" extension (e.g., CSWIPE, CLSWIPE). The new '11.01' Status Request is used to retrieve this additional information. Refer to the following example:

1. POS sends a '11.01' message to the terminal.
2. Terminal returns a '11.01Slide Card[FS]CLSWIPE' message, indicating that it is displaying the text "Slide Card" on the "CLSWIPE" card swipe form. Because the current text display length is variable, a [FS] field separator character is inserted.

The following tables describe the 11.x Status Request message and 11.x Status Response message formats.

11.x Status Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M11_STATUS Message Identifier – ASCII – “11.”
4	2	Decimal	iConnectEFT Constant = P11_REQ_GET_CURRENT_FORM_NAME • 01 = Request Form Name. The Request Form Name field is optional.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

11.x Status Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M11_STATUS Message Identifier – ASCII – “11.”

Offset	Length	Type	Description
4	2	Decimal	iConnectEFT Constant = P11_RES_STATUS_INDICATOR State Indicator: • 00 = Offline. • 01 = Slide/Insert/Tap Card. • 02 = Transaction Type. • 03 = Enter PIN. • 04 = Amount. • 05 = Processing. • 06 = Approved/Declined. • 07 = Barcode Scan. • 10 = Please Sign. • 11 = Signature Accepted. • 12 = Card / Input Capture. • 13 = Card / Input Data Available. • 14 = Select Language. • 15 = Advertising, Offline Advertising. • 16 = Menu. • 17 = Textbox. • 18 = EMV. • 19 = WIC. • 20 = Smart card. • 21 = Barcode Bulk Scan • 22 = Signature Cancelled • 99 = Idle. State indicator 07 (Barcode Scan) applies to iSMPc and iSMP350 terminal models with barcode scanner. State indicator 14 (Select Language) will be returned only if the mainflow.dat parameter '0007_0004' is set to '0' (zero). See also, section Main Flow (mainFlow.dat).
6	Variable	Alphanum	iConnectEFT Constant = P11_RES_CURRENT_DISPLAY_TEXT Text indicating current display on terminal 0 to 32 ASCII characters on first message line.

Offset	Length	Type	Description
M	1	Constant	ASCII control character – FS (0x1C) This field separator is optional. It is only present when responding to a '11.01' Status Request, which specifies that the form name is to be appended to the status response message.
M + 1	Variable	Alphanum	iConnectEFT Constant = P11_RES_CURRENT_FORM_NAME Current form name. This field is optional. It is only present when responding to a '11.01' Status Request, which specifies that the form name is to be appended to the status response message.
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

6.2.15 12.x Account Message

The POS sends the 12.x Account Message to the terminal to enable the terminal to complete an EFT tender when it cannot read the required information from the card. This process is referred to throughout this document as a manual entry.

Data in the 12.x message is of variable length. The account number is separated from the expiration date by the equals sign (=). The account data may be keyed in by the cashier or read in the POS MSR reader.

The POS can send the 12.x message to the terminal when both these conditions are true:

- The POS operator has keyed in an account number and expiration date.
- The POS is expecting EFT tender.

The POS operator or customer must recognize that the terminal is not accepting the card data (display continues to show 'Slide Card'). The operator then keys in the account number and expiration date from the card at the POS keyboard, which then sends the 12.x message to the terminal. The terminal accepts the Account Message data as though it had been read from the card, and proceeds with the account selection and PIN entry. The data in the 12.x message format is included in the 50.x Authorization Request message constructed by the terminal.

When sending discretionary data using the 12.x message, only numeric data should be sent. Non-numeric data will not be accepted. An invalid example and valid example are given below:

Example 1: Invalid Format

4012345678909= 0905000399818VISA/INGENICO

In this example, the discretionary data will be interpreted as invalid.

Example 2: Valid Format

4012345678909= 0905000399818

This format will be accepted by the terminal and the transaction will continue in the normal flow. Refer to the following table which describes the 12.x message format.

12.x Account Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant = M12_ACCOUNT. Message Identifier – ASCII – '12.'
4	Variable	Alphanum	iConnectEFT Constant = P12_REQ_ACCOUNT_NUMBER. Account Number (13 to 24 characters).
M	1	Constant	iConnectEFT Constant = P12_REQ_CVV_SEPARATOR ASCII Character – '= '
M + 1	4	Decimal	iConnectEFT Constant = P12_REQ_EXPDATE. Expiration date in YYMM format.
M + 5	Variable	Numeric	iConnectEFT Constant = P12_REQ_DISCRETIONARY_DATA. Discretionary data (optional).
N	1	Constant	Field Separator – [FS].
N + 1	Variable	Alphanum	iConnectEFT Constant = P12_REQ_CVV_NUMBER. CVV number.
O	1	Constant	ASCII control character – ETX.
O + 1	1	Binary	LRC check character.

> The 12.x Message follows the same format as a card's Track 2 data. As such, it should only contain numeric data and field separators.

A 12.x response message will be sent to POS to indicate if the 12.x request message was successful. See response message format below.

The response can be enabled or disabled by setting `compat.dat` configuration parameter '0013_0021'.

12.x Account Message Response

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant = M12_ACCOUNT Message Identifier – ASCII – '12.'
4	1	Decimal	iConnectEFT Constant = P12_RES_STATUS Response status: • '0' = Successful. • '1' = Failed, due to one of the following: ◦ incorrectly formatted data ◦ value of '0091_0030' = '1' when encryption is enabled ◦ a card has already been swiped
5	1	Constant	ASCII control character – ETX.
6	1	Binary	LRC check character.

6.2.16 13.x Amount Message

6.2.16.1 Overview

The 13.x Amount message flows from the POS to the terminal to provide the current Balance Due amount and to request an Authorization Request message from the terminal.

The Amount message has a variable length. It may have a minimum of 1 and a maximum of 16 amount fields, separated by the FS (Field Separator) character. Each individual amount field length is from 3 to 9 digits, ASCII string, representing the amount in cents, without the decimal point (e.g., $10.85 is represented as 1085). For following example, the following message includes a Current Transaction Balance Due of $12.34.

```
<STX>13.1234<FS>5678<FS>7699<ETX><LRC>
```

A 13.x message with a single amount field is acceptable for compatibility with the existing POS systems. When received, the Amount Index value from the Cards section of `config.dfs` is ignored, and the amount is unconditionally accepted.

When a 13.x message with multiple fields is received, only one field is used. The Amount Index value from the Cards section of `config.dfs` (see Card Configuration Table) selects the correct amount for the authorization message.

- If the Amount Index value is 0, or if it points a field that does not exist, the terminal goes to offline mode.
- If the Amount Index points to a valid value, the amount that the index points to is used in the 50.x Authorization Request message.

- Index 1 points to the first amount field in the 13.x message (e.g., 1234 in the example above), index 2 to second field (5678 in the example above), and so on. The Amount Index applies to 13.x message with multiple fields only.

When the purchase amount value is received in a message other than a 13.x message, such as a 04.x Set Payment Type Request or 28.x Set Variable Request, it is handled the same as the 13.x message with a single amount field.

When the purchase amount is received in a 13.x message with multiple fields, the host variable Amount Due value is not available until the payment is selected.

The purchase amount may be received in the following messages:

- 13.x amount
- 04.x forced payment with amount (single amount field)
- 28.x set amount (single amount field)

6.2.16.2 Cashback Amount

To send the cashback amount to the terminal, the POS must send a 13.x Amount message with the cashback amount appended to the message following a group separator. For example: a transaction amount of $133.00 with a cashback amount of $40 is sent as follows:

[STX]13.13300[GS]4000[ETX][LRC]

The cashback amount is always the last data field appended to the 13.x message.

13.x Amount Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M13_AMOUNT Message Identifier – ASCII – 13.
4	Variable	Decimal	iConnectEFT Constant = P13_REQ_AMOUNT Current Transaction Balance Due (3 to 9 characters). ASCII string representing the BAL DUE in cents (e.g., $123.89 BAL DUE is represented by 12389).
M	1	Constant	Field separator (optional for alternate amount).
M + 1	Variable	Decimal	iConnectEFT Constant = P13_REQ_AMOUNT Next Balance Due (three to nine characters). ASCII string representing the BAL DUE in cents (e.g., $123.89 BAL DUE is represented by 12389) Optional for alternate amount
N	1	Constant	Group separator. Optional. Cashback amount is not required

Offset	Length	Type	Description
N+1	Variable	Decimal	iConnectEFT Constant = P13_REQ_CASHBACK Cashback amount. Optional. Can e present if preceded by a Group separator only. A previous cashback amount is not overwritten if this field is excluded.
O	1	Constant	ASCII control character – ETX
O+1	1	Binary	LRC check character

13.x messages can be sent with multiple empty amount fields or any amount fields excluded. If **any** amounts are unspecified, those transaction amounts are set to a placeholder, *amount not set* value. If the existing transaction amounts are intended to persist, the existing amounts must be present in their respective fields when the message is sent.
For example, if 13.400[FS]300 is sent, followed by 13.400, the excluded second amount (300) is set to *amount not set*.
Cashback amount is only updated if a numerical value is included after the group separator.

6.2.17 14.x Set Transaction Type

The 14.x Set Transaction Type message flows from the POS to the terminal. It is only required if the transaction is not a sale. The application flow changes based on the transaction type. For example, cash back is disabled for non-sale transactions.

14.x Set Transaction Type Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M14_SET_TXN_TYPE Message Identifier – ASCII – “14.”
4	2	Decimal	iConnectEFT Constant = P14_REQ_TXN_TYPE Set the transaction type: • 01 = Sale. • 02 = Void. • 03 = Return. • 04 = Void Return.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

6.2.17.1 EMV and non-EMV Transaction Type Values in the 14.x Message

The following table summarizes RBA 14.x message transaction type values with EMV transaction type tag values:

Transaction Types and Tags

RBA transaction type set via 14.x message	EMV transaction type returned via D1005	Contact EMV transaction type returned via T9C	Contactless EMV transaction type returned via T9C
01 = TT_purchase	00 = EMV_TRANS_TYPE_PURCHASE	00 = Purchase.	00 = Purchase.
02 = TT_void	02 = EMV_TRANS_TYPE_VOID_PURCHASE	Custom void sale/ purchase type value = • '0019_0016' for US common debit AIDs • '0019_0018' for all other AIDs	Custom void sale/purchase value = • '0008_0018' for US common debit AIDs • '0008_0020' for all other AIDs
03 = TT_return	01 = EMV_TRANS_TYPE_REFUND	20 = Refund.	20 = Refund.
04 = TT_voidReturn	03 = EMV_TRANS_TYPE_VOID_REFUND	Custom void return/ refund type value = • '0019_0017' for US common debit AIDs • '0019_0019' for all other AIDs	Custom void return/refund type value = • '0008_0019' for US common debit AIDs • '0008_0021' for all other AIDs
05 = TT_accountVerify	00 = EMV_TRANS_TYPE_PURCHASE	00 (Same as purchase.)	00 (Same as purchase; though not applicable for contactless.)

Void sale/purchase and void return/refund transaction types values are returned according to configurable values:

- '0019_0016' through '0019_0019' for contact EMV.
- '0008_0018' through '0008_0021' for contactless EMV.

6.2.18 15.x Soft Reset Message

The 15.x Soft Reset Message flows from the POS to the terminal. It is used to cancel or clear a specific function of the transaction.

15.x Soft Reset Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M15_SOFT_RESET Message Identifier – ASCII – “15.”
4	1	Decimal	iConnectEFT Constant = P15_REQ_RESET_TYPE Reset type: • 0 = Hard (same as 10.x message). • 1 = Cancel Signature (iSC250/iSC350/iSC480 only). • 2 = Cancel PIN. • 3 = Reset Amount. • 4 = Reset Signature (iSC250/iSC350/iSC480 only). • 5 = Continue Action. • 6 = Stop Action, cancel process started by on-demand message. Cancels EMV transaction if no process (such as signature, cashback, or PIN entry) active. • 7 = Reset PIN. • 8 = Clear Line Item Display. • 9 = Clear data but do not change state. Update line data (or not) per '0007_0007'. Other data is cleared like a 10.x or '15.0' message. Display is not updated. '15.9' should only be used for on-demand flow and while terminal is between commands, and never in the middle of the payment or EMV flow.
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character.

6.2.19 16.x Contactless Mode Request

The 16.x Contactless Mode Request is used by the terminal to send an unsolicited notification to the POS that one or more cards have been detected in the contactless field. As part of this notification, the UID, card index number and card type are sent. A response from the POS is not required. This message is used to support Contactless Key Card features. A message subtype included in the 16.x request message determines the message usage (for example, Contactless Key Card).

6.2.19.1 16.x Contactless Mode Request Message Format

The following tables describes the 16.x Contactless Mode Request message content.

16.5: Contactless Mode Request Format for Contactless Key Card

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – "16."
4	1	Alphanum	Message subtype • 5 = Contactless mode All other values are reserved.
5	1	Alphanum	Message status • 0 = Success • 1 = Failure • 2 = Unsupported message type • 5 = Data overflow. The data to be returned exceeds the communication buffer depth
6	1	Alphanum	Flag indicating whether or not additional data is available • 0 = No additional data is available • 1 = Additional data is available For contactless key card, this flag is always set to 0.
7	1	Alphanum	Card index number. • The first card detected always has a value of 0.
8	1	Alphanum	Card type. • 0 = Unsupported or unknown card type • 1 = MIFARE Classic 1K • 2 = MIFARE Ultralight, UL-C • 3 = MIFARE Mini • 4 = Mifare Classic 4K • 5 = MIFARE DESFire
9	Variable	Hex-ASCII	UID of the card. • MIFARE Classic 1K (format is 4-byte NUID, 8 hex-ASCII digits). • MIFARE Ultralight, UL-C, and DESFire (format is 7-byte NUID, 14 hex-ASCII digits). • MIFARE Mini (format is TBD).
M		Constant	ASCII control character – ETX

Offset	Length	Type	Description
M + 1		Binary	LRC check character

> If multiple cards are detected, an ASCII file separator character (0x1C) will follow the UID of the previous card, which is then followed by the card index number, card type and UID of the next card. Up to 9 cards can be detected in a single tap.

6.2.19.2 Usage Examples

The following examples illustrate 16.x message usage for sending the UID to the POS.

Example 1: MIFARE Classic 1K (with 4-byte NUID), NUID value = {0x4B, 0xE9, 0x1B, 0x4C}

Sequence	Message Content	POS	Terminal
1	16.500014BE91B4C	←	

Example 2: MIFARE Ultralight (with 7-byte NUID), NUID value = {0x04, 0x21, 0xDD, 0x09, 0x36, 0x02, 0x80}

Sequence	Message Direction	Message Content	POS	Terminal
1	Terminal to POS	16.500020421DD09360280	←	

> Telium RBA messaging is ASCII-based. Accordingly, any binary data must be sent as hex-ASCII.

> To read and/or change the contactless mode, use variable 412 (Contactless Mode) with the 29.x Get Variable Request and 28.x Set Variable Request messages.

6.2.20 17.x Merchant Data Write

The 17.x Merchant Data write message is used by the POS to send Contactless Key Card commands to the terminal and return command execution status. Refer to 17.x Merchant Data Write Message Usage Examples for usage examples.

6.2.20.1 ***Contactless Key Card***

The POS sends Contactless Key Card messages to the terminal using the 17.x Merchant Data Write Request message. This message includes the command type (for example, Authenticate sector with Key A (MIFARE Classic 1K only)). Also included in this message is the index number of the card where the command is to be executed, the command ID, and a flag which determines if the terminal should send a 36.x Notification of Command Execution message once the command has been executed. The 17.x Merchant Data Response message returns the command status (for example, successful, failed), confirms the command type and availability of data, and provides the requested data (for example, Authenticate Sector data).

6.2.20.2 17.x Merchant Data Write Request Message Format

Refer to the following table which describes the 17.x Merchant Data Write Request message format.

17.5 Merchant Data Write Request Message Format for Contactless Key Card

Offset	Length	Type	Description
0	1	Constant	ASCII control character - STX
1	3	Constant	Message Identifier – ASCII – 17.
4	1	Alphanum	Message subtype. • 5 = Contactless Key Card mode. All other values are reserved.

Offset	Length	Type	Description
5	1	Alpha	Command type. • A = Authenticate sector with Key A (MIFARE Classic 1K only). • B = Authenticate sector with Key B (MIFARE Classic 1K only). • R = Read ◦ For MIFARE Classic 1K, read 16-byte block from authenticated sector. ◦ For MIFARE Ultralight/UL-C, read four consecutive four-byte pages. ◦ For MIFARE DESFire, read from file with offset and number of bytes to be read (NBR). • W = Write. ◦ For MIFARE Classic 1K, write 16-byte block to authenticate sector. ◦ For MIFARE Ultralight, write four-byte pages. • I = Increment. ◦ For MIFARE Classic 1K, increment a source 16-byte block from an authenticated sector and store the new value in a destination 16-byte value block in the same authenticated sector. ◦ For MIFARE Ultralight, increment a source four-byte page and store the new value in a destination four-byte page. • D = Decrement. ◦ For MIFARE Classic 1K, decrement a source 16-byte block from an authenticated sector and store the new value in a destination 16-byte value block in the same authenticated sector. ◦ For MIFARE Ultralight, decrement a source four-byte page and store the new value in a destination four-byte page. • M = Move. ◦ For MIFARE Classic 1K, move the value in a source 16-byte value block from an authenticated sector into a destination 16-byte value block in the same authenticated sector. ◦ For MIFARE Ultralight, move the value in a source four-byte page into a destination four-byte page. • S = Select AID (MIFARE DESFire only). • L = Get List of AIDs (MIFARE DESFire only). • F = Get File IDs (MIFARE DESFire only). • U = Auto Mode (MIFARE DESFire only).

Offset	Length	Type	Description
			- C = Complete Tap Not all sectors and blocks are available for user data when using Classic MIFARE cards. Care must be given to not write to reserved blocks that could cause the card to be unusable. Block 0 of sector 0 is reserved by the card manufacturer. For every sector, block 3 is reserved for the access restrictions for each of the key types (A and B). User data can generally be written to sector 0, blocks 1 and 2, and for Sectors 1 - X, blocks 0, 1, and 2 totaling 752 bytes for the 1K Classic card and 3,440 bytes for the 4K Classic card.
6	1	Alphanum	Flag which indicates whether a 36.x notification message should be sent to the POS once this command has been executed. - 0 = No 36.x notification message is to be sent. - >0 = Send 36.x notification message.
7	1	Alphanum	Index number of the card where the command is to be executed. This index number must match the index number sent with the card UID in the 16.x card detection message.
8	1	Hex-ASCII	Command ID that is used in conjunction with the 36.x notification message. - If you are not requesting a 36.x notification message, then the value of the ID is ignored and can assume any one-byte (two hex-ASCII digits) value. - If you are requesting a 36.x notification message, the value of the ID should be selected so the individual command can be correctly identified by the POS when sent in the 36.x message.

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td>9</td><td>Variable</td><td>Hex-ASCII</td><td>Command data.
<ul>
<li>Authenticate sector (A or B). For MIFARE Classic 1K only.
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Sector index</td></tr>
<tr><td>2</td><td>6 bytes (12 hex-ASCII digits)</td><td>Key value</td></tr>
</table></li>
<li>Read (R).
<ul>
<li>For MIFARE Classic 1K:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Sector index</td></tr>
<tr><td>2</td><td>one byte (2 hex-ASCII digits)</td><td>Block index</td></tr>
</table></li>
<li>For MIFARE Ultralight:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Page address of the first of four consecutive four-byte pages to be read</td></tr>
</table></li>
<li>For DESFire:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Sector index</td></tr>
<tr><td>2</td><td>one byte (2 hex-ASCII digits)</td><td>Block index</td></tr>
<tr><td>3</td><td>16 bytes (32 hex-ASCII digits)</td><td>Block of data to be written</td></tr>
</table></li>
</ul></li>
<li>Write.
<ul>
<li>For MIFARE Classic 1K:</li>
</ul></li>
</ul></td></tr>
</table>

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Sector index</td></tr>
<tr><td>2</td><td>one byte (2 hex-ASCII digits)</td><td>Block index</td></tr>
<tr><td>3</td><td>16 bytes (32 hex-ASCII digits)</td><td>Block of data to be written</td></tr>
</table>
<ul><li>For MIFARE Ultralight:</li></ul>
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Page address</td></tr>
<tr><td>2</td><td>4 bytes (8 hex-ASCII digits)</td><td>Block of data to be written</td></tr>
</table>
<ul><li>Select AID (S). For DESFire only.</li></ul>
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>3 bytes (6 hex-ASCII digits)</td><td>AID to select</td></tr>
</table>
<ul>
<li>Complete Tap (omits this field)</li>
<li>Get List of AIDs (L) – DESFire only (omits this field)</li>
<li>Get File IDs (F) – DESFire only (omits this field)</li>
<li>Auto Mode (U) – DESFire only (omits this field)</li>
</ul>
</td></tr>
<tr><td>M</td><td>1</td><td>Constant</td><td>ASCII control character – ETX</td></tr>
<tr><td>M + 1</td><td>1</td><td>Binary</td><td>LRC check character</td></tr>
</table>

If multiple commands are sent in a batch, an ASCII file separator character (0x1C) will follow the command data of the previous command which is then followed by the command type, notification flag, card index number, command ID and command data of the next card.

6.2.20.3 17.x Merchant Data Write Response Message Format

Refer to the following table which describes the 17.x Merchant Data Write Request message format.

17.5 Merchant Data Write Response Message Format for Contactless Key Card

Offset	Length	Type	Description
0	1	Constant	ASCII control character - STX
1	3	Constant	Message Identifier – ASCII – 17.
4	1	Alphanum	Message subtype. • 5 = Contactless Key Card mode All other values are reserved.
5	1	Alphanum	Command status. • 0 = Command was successful • 1 = Command failed • 2 = Command not executed, unsupported command type • 4 = Command not executed, invalid command format All other values are reserved.
6	1	Alphanum	Indicates whether or not additional data is available. • 0 = No additional data is available • 1 = Additional data is available For Contactless Key Card, this flag will always be 0.
7	1	Alpha	Command type, returns the value from the request message.
8	1	Alphanum	Indicates whether a 36.x notification message should be sent to the POS when this command has been executed. • 0 = Do not send 36.x notification message • >0 = Send 36.x notification message
9	1	Alphanum	Index number of the card where the command is to be executed. This index number must match the index number sent with the card UID in the 16.x Contactless Mode Request.
10	2	Hex ASCII	Command ID, returns the value from the request message.

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td>12</td><td>Variable</td><td>Hex-ASCII</td><td>Command data.
<ul>
<li>Authenticate sector (A or B). For MIFARE Classic 1K only.
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Sector index</td></tr>
<tr><td>2</td><td>6 bytes (12 hex-ASCII digits)</td><td>Key value</td></tr>
</table>
</li>
<li>Read (R).
<ul>
<li>For MIFARE Classic 1K:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Sector index</td></tr>
<tr><td>2</td><td>one byte (2 hex-ASCII digits)</td><td>Block index</td></tr>
<tr><td>3</td><td>16 bytes (32 hex-ASCII digits)</td><td>Data read from the block</td></tr>
</table>
</li>
<li>For MIFARE Ultralight:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>Page address of the first of four consecutive four-byte pages to be read</td></tr>
<tr><td>2</td><td>16 bytes (32 hex-ASCII digits)</td><td>Data read from the pages</td></tr>
</table>
</li>
<li>For MIFARE DESFire:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>one byte (2 hex-ASCII digits)</td><td>File ID</td></tr>
</table>
</li>
</ul>
</li>
</ul>
</td></tr>
</table>

Offset	Length	Type	Description

Sequence	Format	Description
2	3 bytes (6 hex-ASCII digits)	Offset where the read operation should start (echoes the value in the request message)
3	3 bytes (6 hex-ASCII digits)	Number of bytes to read (echoes the value in the request message)
4	Variable (hex-ASCII digits)	Data bytes read from the file

- Write (W).
  - For MIFARE Classic 1K:

Sequence	Format	Description
1	one byte (2 hex-ASCII digits)	Sector index
2	one byte (2 hex-ASCII digits)	Block index
3	16 bytes (32 hex-ASCII digits)	Block of data to be written

  - For MIFARE Ultralight:

Sequence	Format	Description
1	One byte (two hex-ASCII digits)	Page address
2	Four bytes (Eight hex-ASCII digits)	Block of data to be written

- Select AID (S). For DESFire only.

Sequence	Format	Description
1	Three bytes (Six hex-ASCII digits)	AID that was selected

- Get List of AIDs (L). For DESFire only.

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Sequence</th><th>Format</th><th>Description</th></tr><tr><td>1-n</td><td>Variable (hex-ASCII digits)</td><td>List of AIDs existing on the card</td></tr></table><ul><li>Complete Tap (C) (omits this field).</li></ul></td></tr>
<tr><td>M</td><td>1</td><td>Constant</td><td>ASCII control character – ETX</td></tr>
<tr><td>M + 1</td><td></td><td>Binary</td><td>LRC check character</td></tr>
</table>

6.2.20.4 Executing Commands as a Batch

- All batched commands are executed in the order in which they appear in the 17.x message.
- Only one 17.x response message is sent for the entire batch message. A successful response will indicate that all commands were correctly executed. A negative response will indicate that there was a problem with one of the commands in the batch message.
- For each successful command executed within the batch, the response data for that command will be contained in the 17.x response message. If all commands are executed successfully, then all response data will be included in the 17.x response message.

> All previously written merchant data can be cleared by sending the 17.x message without including any new merchant data in the message.

> Telium RBA messaging is ASCII-based. Accordingly, any binary data must be sent as hex-ASCII.

6.2.20.5 17.x Merchant Data Write Message Usage Examples

6.2.20.5.1 Usage Examples

Example 1: Authenticating Card Sector via the 17.x Message

MIFARE Classic 1K, Sector 2 with Key A and Key Value of {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}

Sequence	Message Content	POS	Terminal
1	17.5A000002FFFFFFFFFFFF	→	
2	17.500A000002FFFFFFFFFFFF	←	

Example 2: Reading from a Card via the 17.x Message

MIFARE Classic 1K

Sector 2, block 1 containing data {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F}

Sequence	Message Content	POS	Terminal
1	17.5R00000203	→	
2	17.500R00000203000102030405060708090A0B0C0D0E0F	←	

MIFARE Ultralight

Pages 4,5,6,7 containing data {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F}

Sequence	Message Content	POS	Terminal
1	17.5R000004	→	
2	17.5000004000102030405060708090A0B0C0D0E0F	←	

Example 3: Writing to a Card via the 17.x Message

MIFARE Classic 1K

Sector 12, block 1 containing data {0x0F, 0X0E, 0X0D, 0X0C, 0X0B, 0X0A, 0X09, 0X08, 0X07, 0X06, 0X05, 0X04, 0X03, 0X02, 0X01, 0X00}

Sequence	Message Content	POS	Terminal
1	17.5W00000B010F0E0D0C0B0A09080706050403020100	→	
2	17.500W00000B010F0E0D0C0B0A09080706050403020100	←	

MIFARE Ultralight

Pages 4 containing data {0x03, 0x02, 0x01, 0x00}

Sequence	Message Content	POS	Terminal
1	17.5W00000403020100	→	
2	17.500W00000403020100	←	

Example 4: Completing Card Tap via the 17.x Message

Sequence	Message Content	POS	Terminal
1	17.5C0000	→	
2	17.500C0000	←	

6.2.20.5.2 Example Batch Messages

MIFARE Classic 1K

Commands in Batch message:

1. Authenticate card sector 2 with key A and key value of {0xFF, 0XFF, 0XFF, 0XFF, 0XFF, 0XFF}.
2. Read sector 2, block 1 containing data {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F}.
3. Write sector 2, block 1 with data {0x0F, 0X0E, 0X0D, 0X0C, 0X0B, 0X0A, 0X09, 0X08, 0X07, 0X06, 0X05, 0X04, 0X03, 0X02, 0X01, 0X00}.
4. Read sector 2, block 1 now containing data {0x0F, 0X0E, 0X0D, 0X0C, 0X0B, 0X0A, 0X09, 0X08, 0X07, 0X06, 0X05, 0X04, 0X03, 0X02, 0X01, 0X00}.
5. Complete the card tap.

The following table shows the messages exchanged between the POS and the terminal for the above described batch message.

MIFARE Classic 1K Example Batch Message Exchange

Sequence	Message Content	POS	Terminal
1	17.5A000002FFFFFFFFFFFF[FS]R00000201[FS]W000002010F0E0D0C0B0A0908 0706050403020100[FS]R00000201[FS]C0000 "[FS]" is the non-printable ASCII file separator character with a value of 0x1C. This field separator is commonly used in RBA messaging to separate variable length fields.	→	
2	17.500A000002FFFFFFFFFFFF[FS]R0000020100010203040506070809 0A0B0C0D 0E0F[FS]W000002010F0E0D0C0B0A090807060504030201 00[FS]R000002010F0 E0D0C0B0A0908070605040302 0100[FS]C0000	←	

MIFARE Ultralight

Commands in Batch message:

1. Read pages 4,5,6,7 containing data {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F}.
2. Write page 4 with data {0x03, 0x02, 0x01, 0x00}.
3. Read pages 4,5,6,7 now containing data {0x03, 0x02, 0x01, 0x00, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F}.
4. Complete the card tap.

The following table shows the messages exchanged between the POS and the terminal for the above described batch message.

MIFARE Ultralight Example Batch Message Exchange

Sequence	Message Content	POS	Terminal
1	17.R000004[FS]W00000403020100[FS]R000004[FS]C0000	→	
2	17.500R000004000102030405060708090A0B0C0D0E0F[FS]W00000403020100[FS] R00000403020100040506070808090A0B0C0D0E0F[FS]C0000	←	

6.2.21 18.x Non-Payment Card Message

The 18.x Non-Payment Card message flows from the terminal to the POS. When the cardholder swipes a card and selects a payment method, that method is checked against the local configuration in the `cards.dat` file. If the "Card Type" option for the selected payment is '1' (indicating a non-payment type card), the terminal sends out the 18.x Non-Payment card message.

18.x Non-Payment Card Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M18_INFO_CARD Message Identifier – ASCII – "18."
4	Variable	Decimal	iConnectEFT Constant = P18_RES_TRACK1 Track 1 data.
M	1	Constant	ASCII control character – FS
M + 1	Variable	Decimal	iConnectEFT Constant = P18_RES_TRACK2 Track 2 data.

Offset	Length	Type	Description
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

6.2.22 19.x BIN Lookup Message

6.2.22.1 Overview

The 19.x BIN Lookup message is sent from the terminal to the POS to allow for external BIN range lookup as a method of preselecting the payment type for the customer. It is automatically disabled if the unit is set to Select Payment Type first. When a cardholder swipes a card, this message is sent to the POS. If the POS responds within the set timeout range, the application either:

- Automatically selects the payment type for the cardholder
- If UNKNOWN was returned, allows the cardholder to proceed and select the payment type

For automatic payment selection to occur, the following conditions must be met:

- The response message must be received within the preset timeout.
- The response counter must match the request counter.
- The account number of the response message must match the account number in the request.

If there is any mismatch, the response is ignored, and the terminal continues waiting within the original timeout value.

When the payment type is automatically chosen for the customer, the payment selection screen is bypassed and the payment is processed as if the customer had pressed the payment key. To any of the screens that the transaction may advance (either via 19.x or the **Select Tender Type** prompt), a <CANCEL> button returns the transaction to the Payment Selection screen without clearing the card data from the application. This allows the cardholder to change the payment type to any other payment type, at which point the application proceeds as a normal Slide Card first transaction without looking up the BIN.

On the **Select Payment** screen, the customer has the option of canceling the previous card swipe, which clears the old MSR data and returns to the **Slide Card** screen to restart the application for the process to be repeated. If the POS fails to respond within the timeout range, the application allows the cardholder to manually select the payment type and ignore any further payment type messages. An account data origin option of *A* enables the application to distinguish between manual entry and an account message.

6.2.22.2 19.x BIN Lookup Message Format

The following table describes the 19.x message format.

19.x BIN Lookup Request Message Format (Sent by terminal)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M19_BIN_LOOKUP Message Identifier – ASCII – 19.

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P19_REQ_ACCOUNT_DATA_ORIGIN Indicates from where account data was derived: • P = Phone Number (used with PayPal). • H = Electronic Track 1. • D = Electronic Track 2 (default track to use). • B = Electronic Tracks (both 1 & 2). • h = Contactless Track 1. • d = Contactless Track 2. • b = Contactless Tracks (both 1 & 2). • X = Manual Track 1. • T = Manual Track 2. • A = 12.x Account Message. The Contactless indicators can be configured in the `config.dfs` file where 0008_0006 handles h and 0008_0007 handles d.
5	1	Alphanum	iConnectEFT Constant = P19_REQ_TRACK1_DATA_STATUS Track 1 good read indicator. If account number is from 12.x account message, set to 0. • 0 = Bad read. • 1 = Good read.
6	1	Alphanum	iConnectEFT Constant = P19_REQ_TRACK2_DATA_STATUS Track 2 good read indicator. If account number is from 12.x account message, set to 1. • 0 = Bad read. • 1 = Good read. If account number is from 12.x account message, set to 1.
7	1	Alphanum	iConnectEFT Constant = P19_REQ_LAYOUT_ID. ▪ 0 - PIN Encouragement request layout

Offset	Length	Type	Description
8	4	Alphanum	iConnectEFT Constant = P19_REQ_COUNTER Request Counter.
12	Variable	Alphanum	iConnectEFT Constant = P19_REQ_ACCOUNT_NUMBER Account number from source, Offset 4.
M	1	Constant	ASCII control character – FS (optional)
M + 1	Variable	Alphanum	iConnectEFT Constant = P19_REQ_TRACK1_DATA Track 1 data (optional).
N	1	Constant	ASCII control character – FS (optional)
N + 1	Variable	Alphanum	iConnectEFT Constant = P19_REQ_TRACK2_DATA Track 2 data (optional).
O	1	Constant	ASCII control character – FS (optional)
O + 1	1	Alphanum	iConnectEFT Constant = P19_REQ_SERVICE_CODE Service code (optional). If parameter 0005_0010 (Append Service Code) is set to 1 then the RBA will append a field separator character and the service code to the 19.x request message.
O + 2	1	Constant	ASCII control character – ETX
O + 3	1	Binary	LRC check character.

Sample Requests

Card Source	Sample Request
STB - Debit	19.A0007341111597242000
Contactless	19.d1100011400557800000150
MSR / Card Swipe	19.D11000106011000990911111
Manual Entry	19.T000000699999999800009901

19.x BIN Lookup Response Message Format (Sent by POS)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M19_BIN_LOOKUP Message Identifier – ASCII – 19.
4	1	Alphanum	iConnectEFT Constant = P19_RES_PAYMENT_TYPE_SELECTED Payment type selected: • - = Prompt user to insert card. • 0 = Invalid card. • 9 = Unknown card. • A = Card type 1: default is debit. • B = Card type 2: default is credit. • C = Card type 3: default is EBT cash. • D = Card type 4: default is EBT food stamp. • E = Card type 5: default is store credit. • ... • O = Card type 15. • P = Card type 16. Undefined responses are treated as Unknown.
5	4	Alphanum	iConnectEFT Constant = P19_RES_COUNTER Response counter (copied from request message).
9	Variable	Alphanum	iConnectEFT Constant = P19_RES_ACCOUNT_NUMBER Account number (copied from request message).
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

6.2.23 19.x Manual-Flow Entry Message

If external BIN searching is enabled via the host (parameter 0005_0002 is set to 1) and the encryption method is selected as P2PE for NCR/Retalix (Parameter 0091_0001 is set to 14), then a special manual 19.x message is sent after the PAN is entered, which enables the POS to control the expiration date and security code entry portions of manual entry.

After the card number is entered, a 19.x message including the PAN length, first six digits and last four digits of the card are send to the POS as requested.

The POS determines whether the CVV or Expiration date are required and notifies the application accordingly by response. If required, the cardholder is prompted to enter these, and this data is encrypted and returned to the POS.

6.2.23.1 *19.x Manual-Entry Flow Request Message Format (Sent by terminal)*

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – 19.
4	1	Alphanum	iConnectEFT Constant = P19_REQ_ACCOUNT_DATA_ORIGIN Account data source = T
5	1	Alphanum	iConnectEFT Constant = P19_REQ_TRACK1_DATA_STATUS Track 1 read indicator = 0
6	1	Alphanum	iConnectEFT Constant = P19_REQ_TRACK2_DATA_STATUS Track 2 read indicator = 1
7	1	Alphanum	iConnectEFT Constant = P19_REQ_LAYOUT_ID Manual entry format = 1
8	4	Alphanum	iConnectEFT Constant = P19_REQ_COUNTER Request counter
12	Variable	Alphanum	iConnectEFT Constant = P19_REQ_PAN_LENGTH PAN length
M	1	Constant	ASCII control character – FS (optional)
M+1	Variable	Alphanum	iConnectEFT Constant = P19_REQ_LEADING_PAN_DIGS Leading digits of PAN (controlled by parameter 0005_0008)
N	1	Constant	ASCII control character – FS (optional)
N+1	Variable	Alphanum	iConnectEFT Constant = P19_REQ_TRAILING_PAN_DIGS Trailing clear digits of PAN (controlled by parameter 0005_0008)
O	1	Constant	ASCII control character – ETX
O+1	1	Binary	LRC check character

6.2.23.2 *19.x Manual-Flow Entry Response (Sent by POS)*

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M19_BIN_LOOKUP Message Identifier – ASCII – 19.
4	1	Alphanum	iConnectEFT Constant = P19_RES_M_VALIDATION_REQUIRED Card validation required: • w = Expiration date and CVV required • x = Expiration date required • y = CVV required • z = No expiration date or CVV required
5	4	Alphanum	iConnectEFT Constant = P19_RES_M_COUNTER Request counter
9	Variable	Alphanum	iConnectEFT Constant = P19_RES_M_PAN_LENGTH Pan length
M	1	Alphanum	ASCII control character – FS
M+1	Variable	Alphanum	iConnectEFT Constant = P19_RES_M_LEADING_PAN_DIGS Leading digits of PAN (controlled by parameter 0005_0008)
N	1	Constant	ASCII control character – FS
N+1	Variable	Alphanum	iConnectEFT Constant = P19_RES_M_TRAILING_PAN_DIGS Trailing clear digits of PAN (controlled by parameter 0005_0008)
O	1	Constant	ASCII control character – ETX
O+1	1	Binary	LRC check character

6.2.24 20.x Signature Message (On-Demand)

The POS initiates Signature capture using the 20.x Signature Request Message (On-Demand). The terminal displays the requested form and prompt, allowing the cardholder to sign.

6.2.24.1 Request

The 20.x Signature Response Message (On-Demand) notifies the POS when the customer presses a keypad key or screen button during the signature process. Signature data is stored to RBA variables 700-709, as explained in Retrieval Using Get Variable.

6.2.24.2 Response

The response message contains a status and can indicate:

- An invalid prompt.
- A state save error (when 0009_0006 = 1).
- A button such as Cancel was pressed, interrupting the signature process.

6.2.24.3 Use

The 20.x Signature Request Message can be used when:

- The payment type requires a signature but the terminal is not configured to automatically prompt for a signature.
- The cashier wants to:
    - Compare the signature with the signature on the card, and would like the customer to sign again.
    - Collect a signature for something other than the payment transaction.

Refresh the Signature form prior to signature capture. Once the signature capture has been captured, the terminal will display the approval status as follows:

- For on-demand, the terminal will display "Signature accepted".
- For normal credit flow, the terminal will display "Approved".

6.2.24.4 Configuration

Parameter 0009_0002 determines if the terminal sends a signature notification message. If 0009_0002 is set to...

- 1, the terminal sends the Signature Ready Response Message (On-Demand) to notify the POS that the signature is completed and ready for download.
- 2, the POS must use the 11.x Status Message to check if signature data is available (also see Retrieval Using Get Variable).

6.2.24.5 Limitations

The terminal will not execute the following messages received during on-demand signature capture. The terminal sends a response including rejection status and the signature process continues.

- 21.x Numeric Input Request Message (On-Demand)
- 23.x Card Read Request (On-Demand)
- 24.x Form Entry Request (On-Demand)
- 25.x Terms and Conditions Request (On-Demand)

The 20.x Signature Request message should only be used after the 50.x Authorization Request message is received or before the transaction has begun. If this message is called during a transaction, the transaction will be aborted.

6.2.24.6 20.x Signature Message Format

The following tables describe the 20.x Signature Request and Response formats. Refer to Signature Ready Response Message (On-Demand) for a description of the 20.x Signature Ready Response. As of RBA version 19.x, the '20.0' response now includes the number of data blocks in the signature.

20.x Signature Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M20_SIGNATURE Message Identifier – ASCII – "20."
4	Variable	Alphanum	iConnectEFT Constant = P20_REQ_PROMPT_INDEX Prompt index number (if 1st character is numeric) or prompt (if 1st character is not numeric). Up to 230 characters. Using a prompt greater than 230 bytes or prompt index of 0 in this field will return a '20.6' (invalid prompt) response.
M	1	Constant	ASCII control character - FS
M + 1	Variable	Alphanum	iConnectEFT Constant = P20_REQ_FORM_NAME Form name. Using a form name greater than 230 bytes in this field will return a '20.6' (invalid prompt) response.
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

20.x Signature Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M20_SIGNATURE Message Identifier – ASCII – “20.”
4	1	Decimal	iConnectEFT Constant = P20_RES_STATUS Status: • 0 = <ENTER> key pressed. • 1 = Signature interrupted by a button pressed. • 6 – Invalid prompt. • 9 – State save error (only returned during errors when 0009_0006 = 1).
5	1	ASCII	iConnectEFT Constant = P20_RES_DATA Data. The use of this field depends on the content of the Status field in the above cell: • If Status = 0, this field contains the number of blocks in the signature • If Status has any other value, this field is excluded.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

6.2.24.7 Usage Examples

The following sections show an example of each type of 20.x request and a scenario for its use.

20.x Signature Request

- The POS sends a 20.x Signature Request message to the terminal requesting the SIGN.K3Z signature form with prompt index 165 ("Please sign and tap OK with pen"). The format for this message will be

  ```
  20.165[FS]SIGN.K3Z
  ```

  The terminal displays the signature form with the specified prompt, and the cardholder can then proceed with signing.

- The POS sends a 20.x Signature Request message to the terminal requesting the SIGN.K3Z signature form with prompt index 165 ("Please sign and tap OK with pen"). The format for this message will be

  ```
  20.165[FS]SIGN.K3Z
  ```

20.x Signature Response

- After signing, the cardholder presses the ENTER key on the terminal. The terminal then responds to the POS with a 20.x Signature Response message indicating which key or screen button was pressed. In this example, the format of the response message will be

  ```
  20.04
  ```

  This indicates the <ENTER> key was pressed and the signature contains four blocks.

- The cardholder begins signing, but presses CANCEL. In this example, the 20.x response is

```
20.1
```

This indicates the cardholder interrupted signature with a button press.

- An invalid prompt was included in the 20.x Signature Request message. The terminal returns the following 20.x Signature Response message:

```
20.6
```

20.x Signature Ready Response

- The 20.x Signature Ready Response Message (On-Demand) message has been enabled. The signature is completed and the terminal sends the following empty message to the POS:

```
20.
```

6.2.24.8 Signature Ready Response Message (On-Demand)

The 20.x Signature Ready Response message flows from the terminal to the POS. It signals the POS that a signature is ready to be downloaded. This message is also optionally sent to the POS when a signature is completed. This message is only sent to the POS if the sig.dat configuration parameter '0009_0002' is set to '1' specifying that the terminal must send a notification message once the signature capture is complete.

The following table describes the 20.x Signature Ready message format.

20.x Signature Ready Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M20_SIGNATURE Message Identifier – ASCII – “20.”
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

6.2.25 21.x Numeric Input Request Message (On-Demand)

The 21.x Numeric Input Request message flows from the POS to the terminal. When it is received, the terminal pauses the payment transaction and prompts the customer for numeric entry (for example, drivers license). When the customer has entered the data or cancelled entry, the Input Response Message is sent to the POS. The payment transaction then resumes where it was paused.

If the customer is in the middle of entering a PIN when the input request is received, the PIN entry will stop, and the customer will be prompted for the input. When the input is complete, PIN entry restarts. Any portion of the PIN that was entered before the interruption is lost. A 15.6 Stop Action soft reset cancels the input request and continues the payment transaction.

The Input Request message is always ACKed. Rules for the message are:

- If the prompt index is not valid, then an input response message with Exit Type = 9 is sent to the POS, and the request is ignored.
- If the prompt length is 0 (zero), that message is not executed, and the 21.9 reject response is sent.
- When the 21.x message is received during the execution of another on-demand function (20.x, 21.x, 23.x, 24.x, 27.x, or 31.x), the new 21.x on-demand message does not execute, a reject response status is returned, and the current on-demand Enter Generic Number process continues to operate.
- When the <CANCEL> button is tapped, a response 21.x message with Cancel state is sent, the terminal displays the Input Cancelled prompt for three seconds, the current process terminates, and RBA returns to the process before initial 21.x was received.
- The on-demand messages are not nested.

Execution of 21.x is terminated in one of the following ways:

- By a message - 00.x, 01.x, 10.x, 15.0, 15.6, 20.x, 30.x.
- By tapping the CANCEL button. The Input Cancelled prompt is displayed, and a 21.1 response (1 indicating the cancel state) is sent. The function terminates and returns to the interrupted state.

6.2.25.1 Use of the 21.x Message to Send Encrypted Clear Entry Data

Sensitive cardholder information, such as their social security number, is encrypted using the same encryption key used for the 94.x and 95.x Barcode Configuration Messages and sent to the POS via the 21.x Numeric Input Request Message. Use configuration parameters 0091_0019 through 0091_0022 to support both sensitive and barcode message encryption. Refer to Security Parameters (security.dat) for more detail. The encryption key used in both cases is base64,

The following configurations enable encryption for specific messages:

- 0091_0026 - Enable encryption of clear entry data via 21.x and 27.x messages.
- 0091_0027 - Enable encryption of barcode data via the 95.x barcode data message.

Exit types 8 and E have been added to the 21.x Numeric Input Response message as described in the below table titled **Numeric Input Response (On-Demand) Message Format.**

21.x Numeric Input Request (On-Demand) Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M21_NUMERIC_INPUT_ON_DEMAND Message Identifier – ASCII – 21.
4	1	Alphanum	iConnectEFT Constant = P21_REQ_DISPLAY_CHARACTER Display character: • 0 = Display digits entered Any other character is displayed in place of the entered digit.

Offset	Length	Type	Description
5	2	Alphanum	iConnectEFT Constant = P21_REQ_MIN_INPUT_LENGTH • Minimum input length: 0 – Maximum input length.
7	2	Alphanum	iConnectEFT Constant = P21_REQ_MAX_INPUT_LENGTH • Maximum input length: 1 – 40.
9	Variable	Alphanum	Prompt index number.
M	1		ASCII control character – FS (This field is optional.)
M + 1	Variable	Alphanum	iConnectEFT Constant = P21_REQ_FORMAT_SPECIFIER Format specifier ID number. See Format Specifiers for more information. (This field is optional.)
N	1	Constant	ASCII control character – FS (This field is optional.)
N + 1	Variable	Alphanum	iConnectEFT Constant = P21_REQ_FORM_SPECIFIC_INDEX_NUMBER Form specific index number from 1-30 or text that is the forms name.
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character.

21.x Numeric Input Response (On-Demand) Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M21_NUMERIC_INPUT_ON_DEMAND Message Identifier – ASCII – 21.

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P21_RES_EXIT_TYPE Exit Type: • 0 = Enter pressed. • 1 = Cancelled. • 2 = Button pressed. • 4 = Invalid form. • 5 = Invalid format. • 6 = Invalid prompt. • 8 = Returned input data encoded using base64 barcode key. • 9 = Declined/Rejected. • E = Error occurred during preparation of response message; no return data included.
5	Variable	Decimal	iConnectEFT Constant = P21_RES_INPUT_DATA Data input.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Info

Examples of Invalid formats (5 = Invalid format, in Response table, above) are non-numeric or negative min/max values.

6.2.26 22.x Application ID Request

The 22.x Application Message flows from the POS to the terminal. It queries the terminal for the ID of the application installed in the terminal.

22.x Application ID Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M22_APPLICATION_ID_REQUEST Message Identifier – ASCII – “22.”
M	1	Constant	ASCII control character – ETX

Offset	Length	Type	Description
M + 1	1	Binary	LRC check character.

22.x Application ID Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M22_APPLICATION_ID_REQUEST Message Identifier – ASCII – “22.”
4	Variable	Alphanum	iConnectEFT Constant = P22_RES_APPLICATION_NAME Application Name.
M	1	Constant	ASCII control character – FS
M + 1	Variable	Alphanum	iConnectEFT Constant = P22_RES_APPLICATION_VERSION Application Version.
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

6.2.27 23.x Card Read Request (On-Demand)

The 23.x Card Read Request message flows from the POS to the terminal. The 23.x request instructs the terminal to:

- Stop executing the current process
- Display the Card Swipe Process screen
- Wait for a manual entry *or* card swipe, insertion, and/or tap as specified in the request

When the MSR track data is available, the terminal:

- Sends a 23.x response with entered data
- Returns to the process the 23.x request interrupted

Note
A card can be inserted before the 23.x request in on-demand flow.

6.2.27.1 Rules

The 23.x message is handled as follows:

- The 23.x request is always ACKed.

- When the 23.x message is received during the execution of 20.x, 21.x, 23.x or 31.x, the 23.x message is not executed and the 23.9 reject response message is returned. The current function is continued.
- To activate a card reader with this message, the reader must be enabled in `config.dfs`. For example, if contactless is disabled (0008_0001 = 0), the 23.x message cannot enable it, so it returns 23.R, a reader-disabled error response.

The on-demand messages are not nested. Refer to 34.x Save and Restore State Messages regarding the ability to send multiple on-demand messages.

The execution of 23.x message during a 20.x Signature Message (on-demand) depends on the value for DFS Data Index 0009_0006 (Save state when signature request received), as follows:

- If 0009_0006 = 0, RBA processes the 23.x message and comes back to process the signature request
- If 0009_0006 = 1, RBA returns 23.9 (declined)

6.2.27.2 23.x MSR Usage

After a card swipe is executed, the following actions can happen:

- When the card read is correct, the Card Accepted prompt displays for a duration specified by 0007_0001 (default three seconds). The terminal sends a 23.x response with status of 0 and attached track data, and RBA returns to the interrupted process.
- When the card read is incorrect, the Card Read Cancel prompt displays for three seconds, the terminal sends a 23.x response with the error status, and the application returns to the interrupted process.

6.2.27.3 23.x EMV Usage

After a contactless card touch/tap is executed, the following actions can happen:

- When the card read is correct, the Card Accepted prompt displays for a duration specified by 0007_0001 (default three seconds). The terminal sends a 23.x response with status of 0 and attached track data, and RBA returns to the interrupted process.
- When the card read is not correct, the Card Read Cancel prompt displays for three seconds, the terminal sends a 23.x response, and the application returns to the interrupted process.

6.2.27.4 23.x Message flow when using iConnect EFT

The 23.x message has different flows depending on the setting of 0013_0014. This parameter controls whether to include or exclude the source field in the 23.x Card Read Response. If set to exclude, the source field from the 23.x response message, the iConnect EFT sends a 61.x Configuration Read message after each 23.x response if 0013_0014 is disabled.

6.2.27.5 Coupon Support Flag

A flag indicates that couponing data (if enabled) is returned in the 23.x response message as opposed to the 16.x Contactless Mode Request message. An optional one-byte Options field supports this feature. All optional fields before the Options field must be present to use it. When set to a value of 1, the application sends a 16.x response message with a status of 6 if payment data is available and a pending a 23.x response message.

Example 1:

In this example, the Options field is set to 1, indicating that the application will send a 16.x response with status 6 if payment data is available and pending 23.x response.

[STX]23.PROMPT[FS]FORM[FS]1[FS]MCS[ETX][LRC]

Example 2:

In this example, the following message uses the prompt Swipe, Tap or Insert Card with form FORM1.K3Z having MSR, Cless, and SCR readers enabled, and having a 16.x response with payment notification.

[STX]23.Swipe, Tap or Insert Card[FS]FORM1.K3Z[FS]1[FS]MCS[ETX][LRC]

Example 3:

This example uses the prompt Swipe, Tap or Insert Card with the default form selection, having all readers enabled by exclusion and forcing the 16.x response with payment notification.

[STX]23.Swipe, Tap or Insert Card[FS][FS]1[ETX][LRC]

Example 4:

This example uses the prompt Swipe or Tap Card with the default form selection, having MSR and Cless readers enabled, without the 16.x response notification.

[STX]23.Swipe or Tap Card[FS][FS][FS]MC[ETX][LRC]

6.2.27.6 MCS (Enable Readers) Field

The MCS field indicates which readers to enable (MSR, contactless and SCR). Any specified reader will not be enabled if the appropriate mode is not enabled in `config.dfs`, regardless of the MCS field content.

If the contactless reader is activated, card swipe on demand with contactless is set as the default form. Refer to the following examples of how MCS field use in the 23.x message affects the default form displayed.

- Use prompt Swipe, Tap or Insert Card with FORM1.K3Z, enabling MSR, contactless and SCR:
  [STX]23.Swipe, Tap or Insert Card[FS]FORM1.K3Z[FS][FS]**MCS**[ETX][LRC]

- Use prompt Swipe or Tap Card with default form, enabling MSR and contactless:
  [STX]23.Swipe or Tap Card[FS][FS]**MC**[ETX][LRC]

6.2.27.7 Bad-Read Errors and Manual-Entry Cancellations

A 23. error response message is returned for the following conditions:

- Invalid card swipe on a swipe reader: 23.1M[FS]
- Invalid contactless tap: 23.1C[FS]
- Invalid card insert on a smartcard reader 23.1S[FS]
- Invalid card insert on a combination swipe/smartcard reader (such as iUN/iUR): 23.1Z[FS]
- Cancellation of manual card number entry by the cardholder: 23.2?[FS]

6.2.27.8 Terminating the 23.x message

The in-progress 23.x message is terminated in one of the following ways:

- By the following messages:
  - 00.x Offline Message
  - 01.x Online Message
  - 10.x Hard Reset Message
  - 15.0 Soft Reset Message (same as 10.x Hard Reset Message)

- 15.6 Soft Reset Message
- 20.x Signature Message (on-demand)
- 30.x Advertising Request Message (on-demand)

- By tapping the Cancel button. The Card Read Cancelled prompt is displayed, and a 23.2 cancel response message is sent. The function is terminated and returned to the interrupted process.

> Encryption and BIN range checks on a card account number are performed based on the settings from the configuration file.

6.2.27.9 *23.x Card Read Request (On-Demand) Format*

The following tables describe the format for the 23.x Card Read Request and 23.x Card Read Response messages. Note that the 87.x On-Guard and KME Card Read Data message uses the same format.

23.x Card Read Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnect EFT Constant = M23_CARD_READ_REQUEST_ON_DEMAND Message Identifier – ASCII – 23.
4	Variable	Alphanum	iConnect EFT Constant = P23_REQ_PROMPT_INDEX Prompt index number. Can be literal text up to 230 characters (such as *Please swipe*), an index number from Prompt.xml, or nothing when used with an optional form_file_name. Value cannot be 0.
M	1	Constant	ASCII control character – FS (This field is optional.) Only used with M + 1.
M + 1	Variable	Alphanum	iConnect EFT Constant = P23_REQ_FORM_NAME Form File Name or Number (this field is optional). Only used with M. Maximum length is 15 characters.
N	1	Constant	ASCII control character – FS (Optional.)
N + 1	1	Alphanum	iConnect EFT Constant = P23_REQ_OPTIONS • 1 = RBA sends a 16.x response with status 6 if payment data is available and pending a 23.x response.

Offset	Length	Type	Description
N + 2	1	Constant	ASCII control character – FS (Optional.)
N + 3	Variable	Alphanum	iConnect EFT Constant = P23_REQ_ENABLE_DEVICES Enable Devices field (this field is optional). Letters in the field indicate which readers to enable: • M = Enable MSR • C = Enable contactless. Contactless is not enabled if 0008_0001 = 0 • S = Enable SCR. Smart card reader is not enabled for EMV if 0019_0001 = 0 or for WIC if 0020_0001 = 0 • H = Ignored if sent with the others. If sent alone, force manual entry for this transaction (0007_0029 = 0 is treated as 0007_0029 = 1) and ignore specified prompt and form. Regardless of readers enabled, 0007_0029 controls whether <ENTER CARD> is displayed on the standard swipe forms. The order of the letters does not matter, only whether or not they are included in the string. The reader(s) specified must be enabled in configuration for this message to activate them. If any specified readers are not enabled by configuration, 23.6 is returned. If no readers are specified, all readers enabled by configuration are activated. To avoid ambiguity, the Form field must be present (even with empty value) for the MCS field to be sent.
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character

23.x Card Read Response Format (if 0013_0014 = 1)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnect EFT Constant = M23_CARD_READ_REQUEST_ON_DEMAND Message Identifier – ASCII – 23.

Offset	Length	Type	Description
4	1	Alphanum	iConnect EFT Constant = P23_RES_EXIT_TYPE Exit type: • 0 = Good Read • 1 = Bad Read • 2 = Cancelled • 3 = Button Pressed • 4 = Cless Card Floor Limit Exceeded • 5 = Max Cless Floor Limit Exceeded • 6 = Invalid Prompt • 7 = Encryption Failed • 8 = Bad Key Card • 9 = Bad format of 23. message and/or on-demand 23. message not allowed because an on-demand request is in progress • A = Amount was not set and the contactless reader was not enabled • R = At least one specified reader is disabled
5	1	Alpha	iConnect EFT Constant = P23_RES_CARD_SOURCE Source of card read: • C = Contactless Reader • E = EMV Contactless • H = Manual entry • M = MSR • Q = Fast Quick Chip • S = One of the following: ◦ SLE5542 memory card ◦ EMV card ◦ WIC card • A = Account message entry • c = Coupon or key card • m = Mobile • Z = combination reader (such as iUC/iUR) - used only when there is a card read error • ? = Unknown/invalid card type Only included in Response if 0013_0014 parameter = 1.

Offset	Length	Type	Description
6	Variable	Decimal	iConnect EFT Constant = P23_RES_TRACK1 Track 1 data. (If the P23_RES_EXIT_TYPE is 3 = Button Pressed, then it is the single byte ID of the key.)
M	1	Constant	ASCII control character – FS
M + 1	Variable	Decimal	iConnect EFT Constant = P23_RES_TRACK2 Track 2 data
N	1	Constant	ASCII control character – FS (only if Track 3 data is present)
N + 1	Variable	Alphanum	iConnect EFT Constant = P23_RES_TRACK3 Track 3 data
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character

6.2.27.10 Sample Responses

The following table provides sample responses for contactless, manual entry and MSR/card swipe card read sources.

Card read examples

Card source	**Sample response**
0013_0014 not set to 1	23.0B4005578000000150^CARDHOLDER/ VISA^101210155555012340000000000556100571740000[FS] 4005578000000150= 10121015555557100741
Contactless	23.0CB4005578000000150^CARDHOLDER/ VISA^101210155555012340000000000556100460770000[FS] 4005578000000150= 10121015555546000771
Manual Entry	23.0H5444009999222205^MANUAL ENTRY/ ^1412000000123000000[FS]5444009999222205= 1412000000123000

Card source	Sample response
MSR / Card Swipe	23.0MB5444009999222205^TESTVOID/ TEST^1412101000000000000000071700[FS] 5444009999222205= 14121010000071700

6.2.28 24.x Form Entry Request (On-Demand)

The 24.x Form Entry Request message flows from the POS to the terminal. When the message is received and accepted, the RBA starts the Form Entry process of displaying the form requested by the command. When the process is complete, or there is an error, the response message is sent to the POS.

In the Form Entry Request format, the four offsets M through O work together to allow you to override a text field on a form. If the form has multiple text fields or buttons, you can repeat the set of fields for each text element or button that needs to change, as long as it has a different B or T number (T1, T2, T3, etc.), and as long as the total message is less than 247 bytes. The B and T blocks can be sent in any order.

6.2.28.1 Setting Prompt Text

The syntax for setting prompt text is as follows:

Tid,text

where

- id = the label ID from the K3Z form.
- text = the new prompt text.

Example: Set the prompt for label id "PROMPTLINE1" to "A new label":

TPROMPTLINE1,A new label

6.2.28.2 Setting Button Text

The syntax for setting button text is as follows:

Tvar,text

where

- var = the button's buttontext variable from the K3Z form.
- text = the new button text.

Example: Set the button with buttontext variable "btn1" to "text":

Tbtn1,text

> Button text in 24.x messages may be specified as an index in the `prompt.xml` file. For example, specifying "106" will call button index 106, [DECLINE]. See 24.x Form Entry Request Message Format table below, index M + 2.

6.2.28.3 Changing Button Visibility

The syntax for changing a button visibility is as follows:

Bid,visibility

where

- id = the button ID from the K3Z form.
- visibility = "S" for show or "H" for hide.

Example: Hide button with id "btn1":

Bbtn1,H

> If "S" or "H" are needed for the button text, then the following syntax should be used in order to avoid confusion:
>
> Bid, S
>
> Bid, H
>
> Note the extra space following "Bid".

6.2.28.4 Setting the Form Timeout

The syntax for setting the form timeout is as follows:

t,timeout

where "timeout" is the form timeout in seconds.

6.2.28.5 Displaying the Offline Form on Bluetooth-Capable Terminals

It is important to note that on terminals supporting Bluetooth pairing, the prior prompt (e.g., "Approved") may be displayed following the
request to display the offline form. In order to correctly display the offline form, the 24.x message must specify T1 formatting and
include the offline.K3Z message. The required message to display the "This Lane Closed" prompt is as follows:
[STX]24.offline.K3Z[FS]T1,This Lane Closed[ETX][LRC]

24.x Form Entry Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M24_FORM_ENTRY_REQUEST_ON_DEMAND Message Identifier – ASCII – “24.”
4	Variable	Alphanum	iConnectEFT Constant = P24_REQ_FORM_NUMBER Form Number or Form Name.
M	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
M + 1	1	Constant	iConnectEFT Constant = P24_REQ_TYPE_OF_ELEMENT ASCII character – 'T'
M + 2	Variable	Alphanum	iConnectEFT Constant = P24_REQ_TEXT_ELEMENTID Text Element ID.
N	1	Constant	ASCII character - ','
N + 1	Variable	Alphanum	iConnectEFT Constant = P24_REQ_PROMPT_IDX Prompt index number.
O	1	Constant	ASCII control character – FS
O + 1	1	Decimal	ASCII character – B.
O + 2	1	Decimal	iConnectEFT Constant = P24_REQ_BUTTONID Button ID.
O + 3	1	Decimal	iConnectEFT Constant = P24_REQ_BUTTON_STATE Button State: • 0 = Up. • 1 = Down.
O + 4	1	Constant	ASCII control character – ETX
O + 5	1	Binary	LRC check character.

24.x Form Entry Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant =M24_FORM_ENTRY_REQUEST_ON_DEMAND Message Identifier – ASCII – "24."

Offset	Length	Type	Description
4	1	Decimal	iConnectEFT Constant = P24_RES_EXIT_TYPE Exit Type: • 0 = Successful. • 1 = Invalid form. • 6 = Invalid prompt. • 8 = Timeout occurred. • 9 = Declined.
5	1	Decimal	iConnectEFT Constant = P24_RES_KEYID ID of key pressed to exit the form.
6	1	Decimal	iConnectEFT Constant = P24_RES_BUTTONID Button ID. This is always paired with the Button State. This is repeated for every check box and radio button.
7	1	Decimal	Button State: • 0 = Up. • 1 = Down. Check Boxes: • 0 = Not Checked. • 1 = Checked. Radio Buttons: • x, where x is a value of 1 to y, where y is the number of buttons in the group. iConnectEFT Constant = P24_RES_BUTTON_STATE Always in a pair with the Button ID. Repeated for every check box and radio button. See also, section "Using multiple buttons with the 24.x message."
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

The 24.x Form Entry Request message is an on-demand message. Accordingly, this messages interrupts the normal operation of the RBA in order to execute a new function requested by the host. The rules for the on-demand functions are:

- On-demand messages cannot be nested.

A received 24.x request is validated prior to execution.

- When a 24.x request has been successfully executed, a 24.x response message is returned with a status of '0' indicating successful execution.
- A 24.x request is not executed when either of the following situations occur:
    - The form file index is invalid, or the form file is not present in the terminal's DFS memory. A 24.x response message is returned with a status of '1' indicating that the form was invalid.

- Another on-demand request is already running. In this event, a 24.x response message is returned with a status of '9' indicating that the request has been declined.

The execution of a 24.x request can be terminated using the following types of reset messages:

- Reset whole transaction and return to card swipe process. This can be implemented using any of the following messages:
  - 10.x Hard Reset Message
  - '15.0' Soft Reset Message
  - 01.x Online Message
  - 00.x Offline Message
  - 30.x Advertising Request Message (On-Demand)
  - 20.x Signature Message (On-Demand)
- A '15.6' Soft Reset Message terminates the current process.

6.2.28.6 Using Multiple Buttons with the 24.x Message

When using multiple buttons on a form (such as a group of radio buttons), the 24.x message returns a single digit group number followed by a single digit button ID for the selected radio button at the time the form was closed. There can be up to four groups (1 – 4) on a single form.

Example:

A form exists with four groups of four radio buttons (group 1 with four radio buttons, group 2 with four radio buttons, etc.). If radio button 4 is selected from group 1, and radio button 2 is selected from group 3, the 24 response may look like this:

24.0[CR]14213241

So, digit-by-digit, the above response reads like this: message 24 returns for group 1/button 4, for group 2/button 1, for group 3/button 2, and for group 4/button 1. The default selection for each group is ‘1’ (button 1).

> Important to note, here, is that the button ID is paired with its value.

The letters [CR] represent non printable characters in the ASCII set.

> Note that the RBA application does not support the simultaneous use of both radio buttons and check boxes on a single form, nor does it support the simultaneous use of radio buttons or check boxes with a signature box on a single form.

6.2.28.7 Displaying Text Data

The maximum radio button and check box length has been increased from 30 characters to 100 characters. If the text exceeds 100 characters, then only the first 100 characters will be displayed.

6.2.28.8 Displaying numerals or symbol characters using the 24.x message

There may be a need to display a numeral or a symbol, such as a dollar sign ($), as the first character on a terminal line. The text in a command string that follows a comma (following a 'T' number) displays this using the following rules:

Parser Action by Character Type

If the first character is… following the comma that follows a 'T' number	**Then the parser does this…**
… a numeral	The parser assumes the numeral to be a prompt index number into the prompt file, so the corresponding prompt will be displayed.
… not a numeral	The parser checks to see if the first character is the Ctrl/Q character (aka DC1, aka ox11 ASCII value). If this non-printable control character is found, then the parser will display whatever text follows. This feature is in place to allow for the printing of text that starts with a number, but it will print any text that follows, not just numbers.
… not a numeral NOR a Ctrl/Q (e.g., 'DC1') character	The parser simply displays all of the text after the comma.

Example:

A form exists with variables (in dollar amounts) for Sales Total, Tax and Total. The three dollar amounts on the form are 5.50, 8.50 and 9.50. This set of data is returned as the following string to the terminal, instructing the terminal what to display:

24.total.k3z[FS]T8,[DC1]5.50[FS]T9,[DC1]8.50[FS]T10,[DC1]9.50

The letters [FS] represent a non-printable character known as Field Separator.

The letters [DC1] represent the Ctrl/Q character.

> Note about the 'T' number: This number should be matched to a 'PROMPTLINEx' number in the K3Z file where 'x' is the 'T' number. The object's 'id' and 'text' fields should use 'PROMPTLINEx' in the same manner as one of the standard RBA K3Z forms using prompts.

The following code from a K3Z file illustrates this point:

```
<Label id='PROMPTLINE8' textsource='custom' text='&lt;?ivPROMPTLINE8?
&gt;' x='320' y='135' width='320' height='30' border='false' bordercolor='000000' textcolor='000
000' fontsize='20px' fontweight='normal' fontfamily='sans-serif'
align='left' background='false' backgroundcolor='FFFFFF' />
```

6.2.29 25.x Terms and Conditions Request (On-Demand)

The 25.x Terms and Conditions Request flows from the POS to the terminal. When the message is received and accepted, the RBA initiates the terms and conditions confirmation process.

The process is opened with a screen defined by the `tc.K3Z` form file. The text displayed is taken from the `tcX.xml` file, where 'X' is the number specified in the command. This file must be ASCII text only with no returns or line feeds. The application will fit the text to the form properly. When the form is displayed (as illustrated in the Terms and Conditions section), the specified terms and conditions are written to the text fields in the form.

Buttons are available to scroll up and down a page, or to scroll one line at a time. Buttons to accept or decline the terms and conditions are specified in the file.

If declined, a response message is returned informing the POS that the terms and conditions were declined. If accepted, the application checks to see if a signature is required. If required, a signature capture form is displayed as illustrated in the Terms and Conditions Signature section. The signature is then retrieved as described in the Signature Retrieval section. Once signature capture is completed, or if the signature is not required, a response message is sent to the POS indicating that the terms and conditions have been accepted.

By default, the Cancel and Enter keys on the keypad are disabled and not acknowledged when the generic Terms and Conditions form is displayed. However, the Terms and Conditions templates would allow the Cancel, Clear, or Enter keys to be acknowledged, and an event would be returned to the application if the .K3Z form did not disable these keys.

25.x Terms and Conditions Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M25_TERMS_AND_CONDITIONS_REQUEST Message Identifier – ASCII – “25.”
4	1	Decimal	iConnectEFT Constant = P25_REQ_SIGNATURE_ON_ACCEPT Signature required: • 0 = Do not prompt for a signature. • 1 = Prompt for a signature.
5	Variable	Decimal	iConnectEFT Constant = P25_REQ_TEXT_NUMBER Text file ID Example: If this field is set to “12”, the text from TC12.XML will display in the text box.
M	1	Constant	ASCII control character – FS (This field is optional)

Offset	Length	Type	Description
M + 1	Variable	Decimal	iConnectEFT Constant = P25_REQ_SIGNATURE_PROMPT_IDX Signature prompt ID.
N	1	Constant	ASCII control character – FS • Only required if form name is included in message.
N + 1	Variable	Decimal	iConnectEFT Constant = P25_REQ_FORM_INFO Form Name (Optional).
O	1	Constant	ASCII control character - ETX
O + 1	1	Binary	LRC check character.

25.x Terms and Conditions Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M25_TERMS_AND_CONDITIONS_REQUEST Message Identifier – ASCII – “25.”
4	1	Alphanum	iConnectEFT Constant = P25_RES_SIGNATURE_ON_ACCEPT Signature on Accept: • 0 = OK - Accept or Decline key was pressed. • 1 = Error - Form not found. • 2 = Error - Text not found. • 9 = Invalid format.
5	1	Alphanum	iConnectEFT Constant = P25_RES_KEY_PRESSED ID of key pressed to exit • "@" = Accept. • "B" = Decline.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

6.2.30 26.x Run Script Request (On-Demand)

The 26.x Run Script Request message is used to request the execution of a specified script file. The request message contains the 26.x message identifier and script filename. A 26.x return message indicates if the script was successfully executed or if there were any errors (e.g., Form not found, Script not found). Telium script files contain arrays of statements which define tags with associated tag parameters. The following table lists and describes the tag parameters used in Telium scripts.

Telium Script Tag Parameters

Parameter	Description
Button	Controls the action of buttons on the form associated with the tag when active.
Form	Associates a form with a tag. The associated form will be displayed when the tag is active.
Scroll	Specifies which text parameter is associated with a scrolling text frame on the form associated with the tag.
Text	Specifies the text to be displayed by a text frame on the form associated with the tag.

Each statement in the script must be contained within one line of the script file. Comments may be included to describe the function of the script. These comments are generally ignored by the script parser. No white spaces, however, are permitted within the actual tag or parameter. Tag descriptions in the script describe the screen which is to be displayed when that tag is active. They also describe transitions to screens associated with other tags. The first tag describes the initial screen and is the first to become active. The order of the other tags in the script is irrelevant. Selection of any buttons which are not associated with a tag parameter will result in termination of the script with their default return value.

The 26.x Run Script Request message provides the file name of the script to be executed. When completed, or if an error occurred, a 26.x Run Script Response message is returned indicating execution status. If an error occurred, the error type will be included in this return message. If a key was pressed to cancel the script, the message will identify which key was pressed.

Refer to the following tables which describe the 26.x Run Script Request message format and 26.x Run Script Response message format.

26.x Run Script Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M26_RUN_SCRIPT_REQUEST Message Identifier – ASCII – “26.”

Offset	Length	Type	Description
4	Variable	Alphanum	iConnectEFT Constant = P26_REQ_FILE_NAME Filename of the script file to execute (must follow Telium filename rules).
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

26.x Run Script Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M26_RUN_SCRIPT_REQUEST Message Identifier – ASCII – “26.”
4	1	Alphanum	iConnectEFT Constant = P26_RES_SCRIPT_RESULT Result: • 0 = Script Complete. • 1 = Form not found. • 2 = Text file not found. • 3 = Script not found. • 4 = Error parsing script file. • 5 = Tag not found.
5	1	Alphanum	iConnectEFT Constant = P26_RES_KEY_PRESSED_TO_QUIT Key pressed to terminate the script. "0" if result field is not "0".
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Info

For additional explanation of the content and format of the script file (.txt file) used, see Appendix B. RBA Script Language.

6.2.31 27.x Alpha Input Message (On-Demand)

The 27.x Alpha Input Message flows from the POS to the terminal. When it is received, the terminal pauses the transaction and prompts for text entry (for example, the customers name). When the customer enters the data or cancels entry, the Alpha Input Message is sent to the POS. The transaction resumes at the step where it was paused.

If the cardholder is in the middle of entering a PIN when the input request is received, the PIN entry will stop, and the customer will be prompted for the input. When the input is complete, PIN entry will restart. Any portion of the PIN that was entered before the interruption will be lost. A 15.6 Stop Action soft reset message cancels the input request and continue the financial transaction.

The Alpha Input Message is always ACKed. Rules for the message are:

- When the 27.x message is received and its prompt length is 0, that message is not executed and the 27.9 reject response is sent .
- When the Enter Generic Text 27.x (also referred to as the Alpha Input message) is received during the execution of another on-demand function (20.x, 21.x, 23.x, 24.x, 27.x or 31.x), the new 27.x message is not executed. A reject response status is returned, and the current on-demand Enter Generic Number process continues .
- If during the execution of the Enter Generic Text 27.x message, the CANCEL button is tapped, a response 27.x message with Cancel state is sent. The terminal displays the Input Cancelled prompt for three seconds, the current process terminates, and RBA returns to the process before the initial 27.x message was received.
- The on-demand messages are not nested.
- Do not use a custom palette on this screen.

Execution of 27.x is terminated in one of the following ways:

- By a message:
  - 00.x Offline Message
  - 01.x Online Message
  - 10.x Hard Reset Message
  - 15.x Soft Reset Message (including 15.6 message)
  - 20.x Signature Message (On-Demand)
  - 30.x Advertising Request Message (On-Demand)
- By tapping the CANCEL button. The Input Cancelled prompt is displayed, and a 27.1 response message is sent. The function is terminated and returned to the interrupted state.

When sending a 27.x message with a form file name, the format specifier field is not required. Both [FS] characters, however, are necessary. The first [FS] separates the prompt from the format, while the second [FS] separates the format from the form name. Refer to the following example message which includes the form name ALPHA.K3Z in the form name field.

27.00020205[FS][FS]ALPHA.K3Z

6.2.31.1 Using the 27.x Message to Send Encrypted Clear Entry Data

Sensitive cardholder data and barcodes can be encrypted with the same encryption key used for the 95.x: Barcode Data Messages and sent to the POS via the 27.x Alpha Input Message. Configuration parameters 0091_0019 through 0091_0022 support generic message encryption. Refer to Security Parameters (security.dat) for more detail. The encryption key used is the base64 barcode key. The following configurations enable encryption for specific messages:

- 0091_0026 - Enables encryption of clear-entry data via 21.x and 27.x messages.
- 0091_0027 - Enable encryption of barcode data via the 95.x barcode data message.

27.x Alpha Input (On-Demand) Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M27_ALPHA_INPUT Message Identifier – ASCII – 27.
4	1	Alphanum	iConnectEFT Constant = P27_REQ_DISPLAY_CHAR Display character: • 0 = Display characters entered. Any other character is displayed in place of the entered digit
5	2	Alphanum	iConnectEFT Constant = P27_REQ_MIN_INPUT_LENGTH • Minimum input length: 0 – Maximum input length.
7	2	Alphanum	iConnectEFT Constant = P27_REQ_MAX_INPUT_LENGTH • Maximum input length: 1 – 40.
9	Variable	Alphanum	iConnectEFT Constant = P27_REQ_PROMPT_INDEX Prompt index number.
M	1	Constant	ASCII control character – FS (This field is optional.)
M + 1	Variable	Alphanum	iConnectEFT Constant = P27_REQ_FORMAT_SPECIFIER Reserved - Not supported. The Alpha Input feature does not require a format specifier (see Format Specifiers). (This field is optional and will be ignored if present.)
N	1	Constant	ASCII control character – FS (This field is optional.)
N + 1	Variable	Alphanum	iConnectEFT Constant = P27_REQ_FORM_SPECIFICID Form specific index number from 1-30 or text that is the form name.
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character.

27.x Alpha Input (On-Demand) Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M27_ALPHA_INPUT Message Identifier – ASCII – 27.
4	1	Alphanum	iConnectEFT Constant = P27_RES_EXIT_TYPE Exit Type: • 0 = Enter pressed. • 1 = Cancelled. • 2 = Button pressed. • 4 = Invalid form. • 6 = Invalid prompt. • 8 = Returned input data encoded using base64 barcode key. • 9 = Declined/Rejected. • E = Error occurred during preparation of response message; no return data included.
5	Variable	Alphanum	iConnectEFT Constant = P27_RES_DATA_INPUT Data input.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

6.2.32 28.x Set Variable Request

The Set Variable message flows in both directions as a request/response pair. It allows the POS to set the value of internal variables, either temporarily or permanently. If stored permanently, the value is set into the file system for retrieval after loss of power. The Set Variable Request is sent from the POS to the terminal to set the value of variables. The terminal returns a Set Variable Response message with the corresponding response code, such as *successful* or *error*.

The POS can send a Set Variable Request at any time. The most common use of the Set Variable Request message is to display items as they are purchased. This feature is referred to as the line display or digital receipt. The standard line display used on generic RBA forms consists of seven lines with 40 characters per line, displayed using the standard Ingenico monospace typeface.

- Variable `104` is used to write the new data to the next available line from the top of the display.
  - If the screen is already full, all existing text is scrolled up one line to accommodate the new data.
- Variables `111` to `119` are used to write to a specific display line.

    - Variable `111` is used to write to the top line (highest on the screen), variable `112` writes to the line beneath the top line, and so on.
    - The number of lines displayed depends on the size of the window defined on the form. While the default scrolling receipt area displays five lines at a time, some can display up to nine. Assign variables to the lines that can be viewed in the terminal scrolling receipt area only. Variables assigned to lines that are not viewable are ignored.
      For example, if the scrolling receipt area displays five lines, use variables `111` to `116` only. Variables assigned to `117` to `119` are not viewable.
- Variable `404` (payment type) is returned with a value from A to P where:
    - A = Debit
    - B = Credit
    - C = EBT Cash
    - D = EBT Food Stamps
    - E = Store Charge
    - F = Loyalty
    - G = PayPal

  Payment types H through P are reserved and customer definable. Refer to Card Configuration (cards.dat) for more information.

The variables may be combined to create a four-line item display with a total line on the bottom by writing the item information to variable `115`, then writing the total line to variable `104`. By default, the terminal clears the line display on reset. Disable this feature and clear the line display by sending a 15.8 Soft Reset Message.

6.2.32.1 Changing Contactless Mode

Contactless mode is changed through the 28.x Set Variable request message. When contactless is enabled using variable `412` with the 28.x message, contactless mode will only remain enabled until the terminal is rebooted. It will be disabled following a reboot of the terminal. To permanently enable contactless mode (such that contactless remains enabled following a reboot), use the 0008_0001 configuration parameter and the 60.x Configuration Write message. The 0008_0001 configuration parameter defines whether the contactless card reader is enabled, and which supported mode is selected (e.g., Isis SmartTap, Google Wallet, EMV).

6.2.32.2 Configuring GMT Variables for PayPal Authorization

Configuration of both Local Time and Greenwich Mean Time (GMT) is required in order to support PayPal payment authorization. Setting the GMT (or GM Time) and Date is especially important in a mixed (U32 and Telium) environment. To facilitate this, variables 201 and 202 are updated with the time and date, respectively. Both variables can also be set via the Telium Manager menu.

The 28.x Set Variable Request message can be used to set the GMT offset variable. The GMT Offset is the time difference in seconds between Local Time and Greenwich Mean Time. For example:

- Variable `205` = GMT Offset (in seconds).
    - Offsets <u>east</u> of the Prime Meridian use a <u>positive</u> number of seconds
    - Offsets <u>west</u> of the Prime Meridian use a <u>negative</u> number of seconds

Info

When calculating the GMT Offset, consideration must be given to any adjustments in time (For example: Daylight Saving Time, British Summer Time).

Info

The 28.x message used to set the GMT Offset must be sent to the terminal at least once after each reboot before any PayPal entry. Periodically update the local time to prevent clock skew; one time per day is adequate.

Info

When rebooting Telium terminals, the GMT Offset variable `(205)` is not saved. The values for variables `201` and `202` are saved upon reboot.

The Local Date and Time variables`(201` and `202`, respectively`)` may be set or changed in any order.

6.2.32.3 Variables

The following table describes the available variables by number. Variables are always available unless noted otherwise.

An in the Get column indicates that the variable can be read with the Get Variable message. An X in the Set column indicates that the variable can be written with the Set Variable message.

Variable Numbers

Variable Number	Description	Notes	Get	Set
1	User variable 1.		X	X
2	User variable 2.		X	X
3	User variable 3.		X	X
4	User variable 4.		X	X
5	User variable 5.		X	X
6	User variable 6.		X	X
7	User variable 7.		X	X

Variable Number	Description	Notes	Get	Set
8	User variable 8.		X	X
9	User variable 9.		X	X
10	User variable 10.		X	X
11	User variable 11.		X	X
12	User variable 12.		X	X
13	User variable 13.		X	X
14	User variable 14.		X	X
15	User variable 15.		X	X
16	User variable 16.		X	X
17	User variable 17.		X	X
18	User variable 18.		X	X
19	User variable 19.		X	X
20	User variable 20.		X	X
21	User variable 21.		X	X
22	User variable 22.		X	X
23	User variable 23.		X	X
24	User variable 24.		X	X
25	User variable 25.		X	X
104	Scrolling line display.			X
111	1st line of line display.	See Note 1.		X
112	2nd line of line display.	See Note 1.		X
113	3rd line of line display.	See Note 1.		X
114	4th line of line display.	See Note 1.		X
115	5th line of line display.	See Note 1.		X

Variable Number	Description	Notes	Get	Set
116	6th line of line display.	See Note 1.		X
117	7th line of line display.	See Note 1.		X
118	8th line of line display.	See Note 1.		X
119	9th line of line display.	See Note 1.		X
120	Bottom line of line display.			X
180	Advertising image.			X
200	Cash Register Number.		X	X
201	Time (HHMMSS).		X	X
202	Date (MMDDYY).		X	X
203	Set GMT Time (HHMMSS).		X	X
204	Set GMT Date (MMDDYY).		X	X
205	Set Different GMT Hour (HH).	Not saved after reboot.	X	X
206	Number of ticks since the terminal boot.		X	
207	Terminal Verification Results (TVR) for a contactless D-PAS transaction where the outcome is *Use another interface*.	Verifies the results of specific certification tests, not standard transactions.	X	
251	Customized Terminal Name (e.g., Retail Base).		X	
252	Customized Terminal Version Number (e.g., 3.4.0.1).		X	
253	Original App Name (e.g., Retail Base).		X	
254	Original Version Number (e.g., 1.1.0.8).		X	
255	Terminal Reference ID (e.g., iPP350, iSC250, iSC350).		X	

Variable Number	Description	Notes	Get	Set
256	Stored value for the RKIVERSION. Enables the POS to compare the current RKIVERSION with its own value to determine if a new `.RKI` file needs to be downloaded. • RKIVERSION can be up to four characters in length. • There are no character restrictions. • 0000 is returned if the RKIVERSION was not set.		X	X
257	Hardware serial number or injected serial number if present.	For iUN, contains iUP serial only.	X	
259	Package version from `package.txt` on the terminals HOST directory	Contains six alphanumeric characters maximum	X	

Variable Number	Description	Notes	Get	Set

260	RGB background color of the iUN display.	Format of the 28 message to use this new variable is...
		28.10000260BBB,GGG,RRR,III
		...where:
		• BBB is the blue value between 0 and 255
		• GGG is the green value between 0 and 255
		• RRR is the red value between 0 and 255
		• III is the intensity value between 0 and 100
		Examples:
		• For a white backlight with full intensity, use the following message:
		`28.10000260255,255,255,100`
		• For a red backlight with medium intensity, use the following message:
		`28.100002600,0,255,50`
			X	X

303	Amount due.	Returns the current transactions purchase amount as provided in the 13.x Amount Message .	X	X
304	Cash Back Limit.		X	X
305	Cash Back Amount.		X	
306	Current Transactions Maximum Cash Back Amount.		X	
307	Transaction Total Amount		X	
310	Cash Back Process		X	

Variable Number	Description	Notes	Get	Set
394	Three-digit service code for last 23.x Card Read Request (On-Demand) message card swipe.		X	
395	On-Demand Card Data Encrypted flag. Indicates whether card data read via an on-demand message were encrypted. • 0 if the card data were not encrypted (either because no P2P encryption was configured, or because the card was found on the whitelist, or because it was a nonstandard card that should not be encrypted) • Encryption type number, if the card data were encrypted	Encryption type numbers are the ones listed in Supported Encryption Methods; for example, 5 for Voltage TEP2.	X	
396	Mod-10 check digit in card read transaction flow. Set to F if the Mod-10 check fails.		X	
397	Mod-10 check digit in 23.x message. Set to F if the Mod-10 check fails.	23.x Card Read Request (On-Demand) message or 41.x Card Read Message	X	
398	Card read On-Demand account number.	23.x Card Read Request (On-Demand) message or 41.x Card Read Message	X	
399	Card read On-Demand account name.	23.x Card Read Request (On-Demand) message or 41.x Card Read Message	X	
400	Card read On-Demand expiration date.	23.x Card Read Request (On-Demand) message or 41.x Card Read Message	X	
401	Payment card account number.		X	
402	Payment card account name.		X	

Variable Number	Description	Notes	Get	Set
403	Payment card account expiration date.		X	
404	Payment type. Refer to Card Configuration (cards.dat) for valid values.		X	
405	Payment track (track which will be used in the 50.x Authorization Request message).		X	
406	MSR Track 1 data.		X	
407	MSR Track 2 data.		X	
408	PIN block.		X	
409	Default language. This is the language which is used on the screen following any reset.		X	X
410	Current language. Screen prompts will appear in this language, but the default language remains unchanged.		X	X
411	MSR Track 3 data.		X	
412	Contactless Mode.	Once enabled, this variable remains enabled until the terminal is rebooted.	X	X
413	Service code for card which is used to determine whether or not the swiped card is an EMV card. The POS should check this service code when the card is swiped. If the swiped card is an EMV card then the cardholder will be prompted to insert the card in the payment terminal chip card reader.	This variable is always available for card type verification whether EPS encryption is enabled or not.	X	

Variable Number	Description	Notes	Get	Set
414	Standard Flow Card Data Encrypted flag. Indicates whether card data read during the standard payment process flow were encrypted. • 0 if the card data were not encrypted (either because no P2P encryption was configured, or because the card was found on the whitelist, or because it was a nonstandard card that should not be encrypted) • Encryption type number, if the card data were encrypted	Encryption type numbers are the ones listed in Supported Encryption Methods; for example, 5 for Voltage TEP2.	X	
415	Temporarily overrides the <CANCEL> key behavior configured for input entry using parameter 0013_0022. The configured value is restored when any application reset message is sent (e.g., 00.x, 01.x, 10.x). This variable applies to non-iUN terminals only.			
416	Account type for MSR transactions	Interac only. • 1 = Checking • 2 = Savings	X	
420	Determines whether to allow EMV fallback. • 0 = Disabled. Proceed with normal process. • 1 = Enabled. Implement fallback and continue transaction. Refer to EMV Flags (emv.dat) for fallback condition flags.	Reset to 0 at the start of each transaction. It must be sent after reset and before an EMV transaction begins. This variable is applicable to unattended terminals with combo readers only.	X	X

Variable Number	Description	Notes	Get	Set
421	Samsung Loop Pay detected.	The terminal detected a Samsung Loop Pay swipe rather than a regular card swipe. • 1 = Samsung Loop Pay detected. • 0 = Not detected.	X	
430	Response of required fleet card prompts.	The content of this variable is described in Fleet Prompting.	X	
440	Voltage rollover date.		X	
450	M/S key present - Index 0.		X	
451	M/S key present - Index 1.		X	
452	M/S key present - Index 2.		X	
453	M/S key present - Index 3.		X	
454	M/S key present - Index 4.		X	
455	M/S key present - Index 5.		X	
456	M/S key present - Index 6 - Non KP4 only.		X	
457	M/S key present - Index 7 - Non KP4 only.		X	
458	M/S key present - Index 8 - Non KP4 only.		X	
459	M/S key present - Index 9 - Non KP4 only.		X	
470	DUKPT key present - Index 0.		X	
471	DUKPT key present - Index 1.		X	
472	DUKPT key present - Index 2.		X	
473	DUKPT key present - Index 3.		X	
474	DUKPT key present - Index 4.		X	
475	DUKPT key present - Index 5.		X	
476	DUKPT key present - Index 6 - Non KP4 only.		X	

Variable Number	Description	Notes	Get	Set
477	DUKPT key present - Index 7 - Non KP4 only.		X	
478	DUKPT key present - Index 8 - Non KP4 only.		X	
479	DUKPT key present - Index 9 - Non KP4 only.		X	
500	24-hour reset counter.	This variable can be used to query the status of the automatic 24-hour reboot feature for PCI v4 compatible terminals. Return values are as follows: • Positive number = number of seconds until the next automatic reboot. • 0 = 24-hour reboot feature is not enabled. • -1 = An error has occurred.	X	
510	Number of seconds until the next scheduled Estate Manager download	Format is hours:minutes:seconds. For example, 4:31:07 (four hours, 31 minutes, seven seconds).	X	

Variable Number	Description	Notes	Get	Set
511	Number of seconds until the next scheduled Estate Manager download	Similar to variable 510, with these differences: Variable 511 can be **set** as follows to force a Estate Manager download: • Positive value N = Requests a future download N seconds from now • Negative value = Requests an immediate Estate Manager download • 0 = Stops the Estate Manager scheduling process When **getting** the value, the format is simply the number of seconds (not hh:mm:ss): • A positive value reflects the number of seconds until the next Estate Manager download; this is true whether the download was scheduled via TMS.XML or via setting variable 511 • A negative value indicates one of the following situations: ◦ TMS.XML is not available ◦ TMS.XML is not scheduled for a future download ◦ Variable 511 has been set to 0 to stop the Estate Manager scheduler	X	X
600	EMV contact configuration.		X	
601	EMV contactless configuration.		X	
602	Set the suspend list for contactless EMV transactions.			X

Variable Number	Description	Notes	Get	Set
603	Set the update list for contactless EMV transactions.			X
604	Set the suspend timer for contactless EMV transactions.			X
605	Contactless Magstripe Application ID (AID).	The value for this variable is cleared when any of the following events occur: • Terminal goes online. • Terminal goes offline. • Start of transaction. • Transaction reset.	X	
606	Contactless Magstripe Device Type	Stores the value of tag T9F6E to make Device Type data available for use during MSD transactions.		X
700	Signature block 1.		X	
701	Signature block 2.	See Note 2.	X	
702	Signature block 3.	See Note 2.	X	
703	Signature block 4.	See Note 2.	X	
704	Signature block 5.	See Note 2.	X	
705	Signature block 6.	See Note 2.	X	
706	Signature block 7.	See Note 2.	X	
707	Signature block 8.	See Note 2.	X	
708	Signature block 9.	See Note 2.	X	
709	Signature block 10.	See Note 2.	X	
711	Maximum signature size in bytes.	See Note 2.	X	X
712	Signature size in blocks.		X	
713	Signature size in bytes.		X	
800	Host IP address.		X	X

Variable Number	Description	Notes	Get	Set
801	Host IP Port Number.		X	X
802	Communications inactivity timeout.		X	X
803	Reject new connections (IP only).		X	X
804	Add time stamp to message.		X	X
805	Clear-Text Key Press Input	• 0 = Disabled. • 1 = Enabled, asterisk (*) and pound (#) keys supported. • 2 = Enabled, asterisk (*) and pound (#) keys not supported. • (blank) = Default, functions as though disabled.	X	X
806	Get or set the IP address of the terminal in both static and dynamic modes	• Setting the new IP address takes effect only if the terminal is in static address mode. • If the terminal is switched from dynamic mode to static mode, the new IP address is used. • The terminal reboots after sending a 28.x message to set this variable.	X	X
807	Temporarily overrides 0006_0014 setting	When RBA is reset, this variable reverts to the value of 0006_0014 in `pin.dat`.	X	X
810	List of injected keys (KSN, KCV keys)	Lists the encryption keys injected in the terminal by Key Sequence Number (KSN) or Key Check Value (KCV). There is an entry for each injected key, and each entry is terminated with a Line Feed character (0x0A). **Example:** The following code represents the KSN for the key injected in slot 2: `KSN_2=FFFF3D01000000E00001`	X	

Variable Number	Description	Notes	Get	Set
820	Cradle association data (iSMP only)	Used for iSMP cradle association only. It holds a string of the 14 hex-ASCII character address of the cradle. When a 28.x message is received, the previous cradle association is replaced with the new cradle address. The first two characters of this variable are 00 because the Telium Manager does not save the first two bytes.	X	X
830	Battery Power % (iSMP4 only)	The percentage of charge remaining in the terminal battery. When the battery level is less than five percent, and the battery is not on the charging base, the terminal powers off. An alarm sounds every five seconds for the last minute before powering off, prompting the user to place the terminal on a charger. The power down time can be set using Automatic Power Off (0007_0035) in the `mainflow.dat` file. **Note:** If the terminal is in sleep mode, applications are idle, and the terminal cannot power off automatically, but when the charge reaches zero percent, the terminal powers off.	X	
831	Plugged-in status (iSMP4 only)	• 0 = The terminal is not receiving power • 1 = The terminal is connected to a charger or dock	X	

Variable Number	Description	Notes	Get	Set
832	Current charge setting (iSMP4 only)	This variable reverts to 2 when the terminal boots or is removed from an external power source: • 0 = No charging • 1 = Low (500 mA) • 2 = High (2000 mA)	X	X
833	Current charge setting (iPad used in serial mode with iSMP4 only)	Sets the current charge of the iPad connected with an iSMP4 in a Wi-Case operating in serial mode: • 0 = Turn off charging • 1 = 500 ma • 2 = 1000 ma • 3 = 2100 ma • 4 = 2400 ma **Note:** Invalid values default to 1000ma.	X	X
840	Bit rate (iUC285 only) • Upon starting the kiosk, the telemeter must connect to the terminal via serial connection using a 9600 bit rate. • When a connection is established, the terminal must switch to a bit rate of 115200. Every time variable 840 is set, the terminal sends an ACK response to the request using the prior bit rate. If the response is: • Successful, the terminal sends: ◦ A 28.1 response at the prior bit rate ◦ A 28.2 response at the new bit rate • Unsuccessful, the terminal sends a 28.3 response at the prior bit rate.	The bit rate an iUC285 communicates at: • 300 • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 115200	X	X

Variable Number	Description	Notes	Get	Set
900	Last calculated Token value.		X	

Note 1

Variables exceeding the window height are ignored. The line display height is defined by the current form.

Note 2

Blocks beyond the number indicated by the `712` variable do not contain data.

6.2.32.4 28.x Set Variable Request and Response Message Formats

If the request message has the response type set to 1, a Set Variable Response message is sent to the POS. The response is not sent if the response type is set to 9. The following tables describe the Set Variable Request and Set Variable Response message formats.

28.x Set Variable Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M28_SET_VARIABLE_REQUEST Message Identifier – ASCII – 28.
4	1	Decimal	iConnectEFT Constant = P28_REQ_RESPONSE_TYPE Set the response type: • 1 = Send response message. • 9 = Do not send response message.
5	1	Decimal	Reserved = ASCII - 0 (should be set to 0)
6	6	Decimal	iConnectEFT Constant = P28_REQ_VARIABLE_ID Variable ID.

Offset	Length	Type	Description
12	Variable	Alphanum	iConnectEFT Constant = P28_REQ_VARIABLE_DATA Variable Data.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

28.x Set Variable Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M28_SET_VARIABLE_REQUEST Message Identifier – ASCII – 28.
4	1	Decimal	iConnectEFT Constant = P28_RES_STATUS Status: • 2 = Successful. • 3 = Error. • 4 = Insufficient Memory. • 5 = Invalid ID. • 6 = No Data.
5	1	Decimal	Pad – ASCII – 0
6	6	Decimal	iConnectEFT Constant = P28_RES_VARIABLE_ID Variable ID.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

6.2.33 29.x Get Variable Request

The Get Variable message is architected to flow in both directions as a request/response pair. A predefined set of internal variables are stored in the terminal's RAM. The Get Variable Request message is sent from the POS to request the value of one of these variables. The terminal returns a Get Variable Response message with the data for the requested variable.

The Get Variable Request message sent from the POS contains no variable data. The Get Variable Response message will contain the following:

- Success status of the variable retrieval ,
- The variable number echoed,
- The data contained in the variable.

The 29.x Get Variable Request message works in conjunction with the 28.x Set Variable Request message. Variables are set using the 28.x message and then retrieved using the 29.x message. The POS can send a Get Variable Request at any time. The data contained in that variable at the time of the request is returned. The Get Variable message can also be used to retrieve a signature as described in the Signature Retrieval section, once the customer has finished signing. Refer to the Variable Numbers table in the 28.x Set Variable Request section for a list of variables used in the message, including their respective ID numbers.

6.2.33.1 29.x Get Variable Request and Response Message Formats

The following tables describe the Get Variable Request and Get Variable Response message formats.

29.x Get Variable Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant = M29_GET_VARIABLE_REQUEST Message Identifier – ASCII – “29.”
4	2	Constant	ASCII Character – “00.”
6	6	Decimal	iConnectEFT Constant = P29_REQ_VARIABLE_ID Variable ID.
M	1	Constant	ASCII control character – ETX.
M+1	1	Binary	LRC check character.

29.x Get Variable Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M29_GET_VARIABLE_REQUEST Message Identifier – ASCII – “29.”

Offset	Length	Type	Description
4	1	Decimal	iConnectEFT Constant = P29_RES_STATUS Status: • 2 = Successful. • 3 = Error. • 4 = Insufficient Memory. • 5 = Invalid ID. • 6 = No Data.
5	1	Decimal	Pad – ASCII – “0.”
6	6	Decimal	iConnectEFT Constant = P29_RES_VARIABLE_ID Variable ID.
12	Variable	Alphanum	iConnectEFT Constant = P29_RES_VARIABLE_DATA Variable Data.
M	1	Constant	ASCII control character – ETX.
M + 1	1	Binary	LRC check character.

6.2.34 30.x Advertising Request Message (On-Demand)

The 30.x Advertising Request message flows from the POS to the terminal. When this message is received, the current payment transaction is cancelled and the terminal begins displaying the advertising screens if advertising is enabled. This message executes the same functions as the 10.x Hard Reset Message with one exception; the RBA proceeds to advertising provided that parameter 0010_0003 (End of Transaction Mode) is not set to 0. When the 10.x Hard Reset message is received, however, the RBA proceeds to transaction start.

When a 30.x message is received during the execution of any on-demand message, that process is aborted and RBA proceeds with advertisements. As an example, if a card is swiped using a payment method which is not supported by the terminal, the 30.x message can be used to abort the transaction and display the advertising screens while the cashier processes the transaction without the terminal.

When the 30.x message is received in an acceptable state (other than offline), the RBA implements the following:

- ACKs the received message but does not send a response.
- Clears the financial transaction and all customer data including; signature, purchase amount, receipt items, and account values.
- Terminates the current process and proceeds to advertising without delay.

During the execution of 30.x message, the following may occur:

- 11.x request is received and the response contains numerical value 15 and text “Advertising.”
- The advertising bitmaps are displayed one at a time. The mode of displaying advertising images is selected in the Advertising section in config.dfs, index 2. Based on the selected option, the RBA will:

  - Show one bitmap and wait for a transaction reset message from the POS
  - Show one bitmap, reset the transaction, and automatically go to the transaction start
  - Display all bitmaps, one at a time, until the reset transaction message is received
  - Display all ads one time and wait for the reset

The 30.x message is ignored under the following conditions:

- When the terminal is in the Offline mode and Offline advertising is disabled.
- When this message is received in the advertising mode entered from the Offline mode.

In both cases the terminal will respond with a 00.x Offline Message.

In order to enable advertising when in the Offline mode, the Offline Advertising Mode parameter (0010_0001) must be set to a value other than 0. If the Offline Advertising Mode is disabled (configuration parameter 0010_0001 = 0) and the RBA is not in the offline state, then the 30.x message overrules that option and proceeds to the advertisements.

Execution of 30.x is terminated by the following messages:

- 00.x Offline Message
- 01.x Online Message
- 10.x Hard Reset Message
- 15.0: Soft Reset Message
- 15.6: Soft Reset Message
- 20.x Signature Message (On-Demand)
- 21.x Numeric Input Request Message (On-Demand)
- 23.x Card Read Request (On-Demand)

The following table describes the format for the 30.x Advertising Request message.

30.x Advertising Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M30_ADVERTISING_REQUEST_ON_DEMAND Message Identifier – ASCII – “30.”
4	Variable	Decimal	iConnectEFT Constant = P30_REQ_NUMBER_OF_ADDS Number of ads to display (optional). If an invalid value is provided or if omitted from the message, the setting for configuration parameter 0010_0008 (Form to display when 0010_0007 is set to 1) in ads.dat is used.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

6.2.35 31.x PIN Entry Messages (On-Demand)

6.2.35.1 Overview of the 31.x PIN Entry Message

The POS sends the 31.x PIN Entry Request to display a form on the terminal prompting the customer to enter the PIN number.

RBA now supports forms for capturing PayPal passcodes. This is implemented by selecting the Key Type as "P" for PayPal, setting the Prompt Index Number to "31", and including the form name in the message. Also when selecting the Key Type as "P" for PayPal, the Customer Account Number parameter is not required. Consider the following example 31.x request message:

31.P031[FS][FS]PPALPCAN.HTM

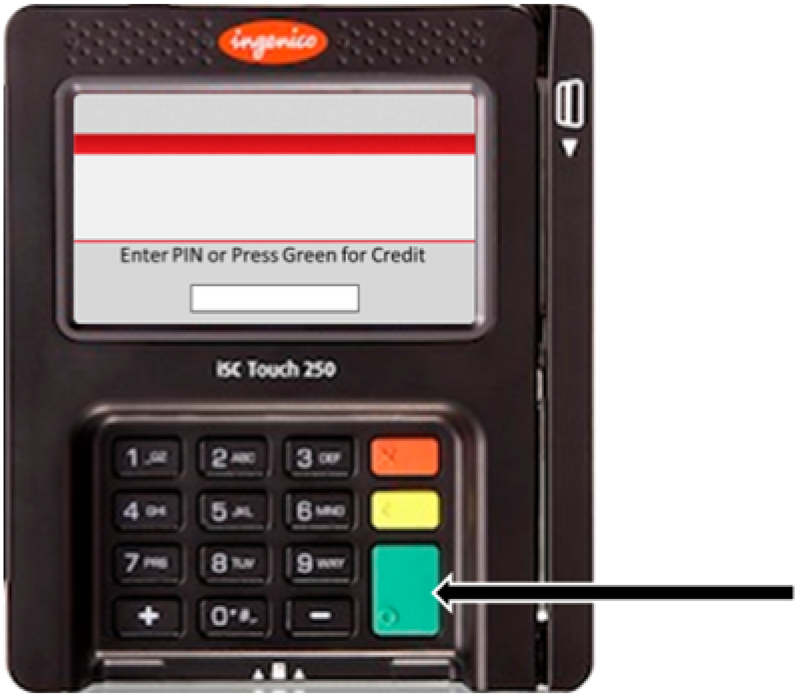
In the above example, the 31.x message is sent with the following parameter settings:

- "P" (PayPal) selected as the Key type.
- "0" encryption configuration.
- Prompt Index Number of "31" which selects the PayPal PIN entry prompt.
- Form name "PPALPCAN.HTM".

Note that the Customer Account Number field has been omitted from the message.

6.2.35.2 PIN Entry with Credit Selection Option

A new PIN entry display message has been added which provides the cardholder with the option of entering their PIN or selecting Credit by pressing the green <Enter> key. Prompt number 17 has been added to the `SECURPROMPT.XML` to support this new feature. When selected, the `INPUT.K3Z` form will display the "Enter PIN or Press Green for Credit" message prompting the cardholder to enter their PIN or select Credit. If no PIN is entered and the green <Enter> key is pressed, the transaction will be processed as a credit transaction. Forms supporting this feature are illustrated for various Ingenico terminals in the Enter PIN or Press Green for Credit section of this document. The following figure illustrates an Ingenico iSC Touch 250 with the new PIN entry form.



Press Green for Credit

Ingenico iSC Touch 250 with PIN Entry and Credit Selection Option

6.2.35.3 Account Number Verification

The RBA compares the account number provided in the 31.x PIN Entry Request to the account number read from the card swipe via the 23.x Card Read Request (On-Demand) message. If there is any discrepancy then a '31.7' PIN Entry Response message will be returned indicating a mismatch.

Refer to the following tables for a more detailed description of the 31.x PIN Entry Request message format and 31.x PIN Entry Response message format.

31.x PIN Entry Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M31_PIN_ENTRY Message Identifier – ASCII – “31.”
4	1	Alphanum	iConnectEFT Constant = P31_REQ_SET_KEY_TYPE Key Type selection: • M = Master/Session. • D = DUKPT. • P = PayPal. • * = Default.

Offset	Length	Type	Description
5	1	Alphanum	iConnectEFT Constant = P31_REQ_SET_ENCRYPTION_CONFIGURATION Overwrites the encryption configuration: • 0-9 = Overwrites the configuration. • * = Uses the default configuration.
6	Variable	Alphanum	iConnectEFT Constant = P31_REQ_PROMPT_INDEX_NUMBER Prompt index number. Examples include: • 17 = Enter PIN or Press Green for Credit prompt. • 31 = PayPal PIN Entry prompt.
M	1	Constant	ASCII control character – FS
M+1	Variable	Alphanum	iConnectEFT Constant = P31_REQ_CUSTOMER_ACC_NUM Customer’s Account Number. If Key Type (P31_REQ_SET_KEY_TYPE) is selected as PayPal, then the Customer's Account Number is not mandatory.
N	1	Constant	ASCII control character – FS
N + 1	Variable	Alphanum	iConnectEFT Constant = P31_REQ_FORM_NAME Form Name.
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character.

31.x PIN Entry Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant = M31_PIN_ENTRY Message Identifier – ASCII – “31.”

Offset	Length	Type	Description
4	1	Decimal	iConnectEFT Constant = P31_RES_STATUS Status: • 0 = PIN entered. • 1 = Cancelled, or invalid form. • 2 = Invalid Length. • 3 = Invalid Index. • 6 = Invalid prompt. • 7 = Account number in the 31.x PIN Entry message does not match the account number returned from the card swipe via the 23.x Card Read Request message. • 9 = Request declined. • A = Physical button pressed (key value is return in the PIN data). • B = Onscreen button pressed. • T = PIN entry timed out. • | = PIN entry bypassed without a key press, such as by timeout. • ; = PIN entry restarted without a key press, such as by timeout.

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td>5</td><td>Variable</td><td>Alphanum</td><td>
<p>iConnectEFT Constant = P31_RES_PIN_DATA</p>
<p>This field's contents depend on value of the preceding Status field:</p>
<ul>
<li>If Status is 0 (PIN entered), this field contains PIN data.</li>
<li>If Status is 4 (Onscreen button pressed), this field is one character long, and its content contains the onscreen button's <code>buttonid</code> form element, depending on the value of the onscreen button's <code>key</code> form element.
<ul>
<li>If <code>key</code> is mapped to a physical button, this field will return the physical button's hex value. For example, [CR] (carriage return character) is returned for a virtual Enter key. A virtual button mapped to the Cancel key results in a 31.1 response.</li>
<li>If <code>key = 'CUSTOM'</code>, refer to the following table.
<p><b>Handling of Custom Keys for MSR Debit</b></p>
<table>
<tr><th>buttonID</th><th>Terminal Action</th></tr>
<tr><td>A decimal digit 1-9</td><td>The terminal returns to appropriate PAY<b>X</b>.K3Z select payment type form where X = the decimal digit if available.</td></tr>
<tr><td>An alpha char A-P</td><td>The terminal selects the appropriate/ enabled <b>payment type</b> if enabled.</td></tr>
<tr><td>Any other value</td><td>The terminal instead sends 24.0<b>?</b> where <b>?</b> is the <code>buttonID</code>.
For EMV PIN entry, 24.0? is sent for all buttonid values.</td></tr>
</table>
</li>
</ul>
</li>
<li>If Status is A (PIN key pressed), this field is one character long, and contains one of the following values:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>*</td><td>PIN numeric digit pressed</td></tr>
<tr><td>=</td><td>'Clear' key pressed (sent regardless of whether a PIN digit was or was not removed)</td></tr>
</table>
</li>
</ul>
</td></tr>
</table>

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>/</td><td>Either:<ul><li>'Enter' key pressed when <b>too few</b> PIN digits entered, or</li><li>another PIN digit pressed when <b>max</b> number of PIN digits are already entered</li></ul></td></tr>
<tr><td>|</td><td>'Enter' key pressed when <b>no</b> PIN digits entered <b>and</b> PIN bypass enabled (e.g. MSR debit PIN entry and '0006_0013' = '1')</td></tr>
<tr><td>;</td><td>'Cancel' key pressed when at least one PIN digit entered <b>and</b> PIN restart enabled ('0013_0022' or RBA variable #415 = '1')</td></tr>
<tr><td>?</td><td>Invalid PIN key pressed (e.g. '+'/'-', 'F', 'F' keys, etc.)</td></tr>
</table>
<ul><li>If Status = B, this field either contains:<ul><li>| if a virtual enter button is pressed ONLY for MSR or EMV PIN entry, or</li><li>The single-character buttonID of the virtual button pressed.
31.B messages do not contain the buttonid for CANCEL, but the terminal instead returns a 31.1 response.</li></ul></li><li>For all other values, this field is empty.</li></ul>
</td></tr>
<tr><td>M</td><td>20</td><td>Alphanum</td><td>Optional Key Serial Number (KSN) used for DUKPT encryption.</td></tr>
<tr><td>M + 20</td><td>1</td><td>Constant</td><td>ASCII control character - ETX</td></tr>
<tr><td>M + 21</td><td>1</td><td>Binary</td><td>LRC check character.</td></tr>
</table>

PIN Bypass
By default, at least 4 digits must be entered, or a number of digits equal to the value set by '0006_0011'. If '0006_0004' is set to '1' then a valid PIN or and empty PIN is accepted when submitted. This does not change the behavior of PIN entry during normal transaction flow. Refer to PIN Entry (pin.dat) parameter '0006_0004'.
31.x PIN bypass returns '31.40x0D', where '0x0D' is the hex value for 'Enter' key. 31.x could possibly return virtual key presses from PIN entry form but will not return any other physical keys.

6.2.36 34.x Save and Restore State Messages

The 34.x Save and Restore State Messages message enables you to process consecutive on-demand messages in your transaction flow, especially when used in conjunction with the '0007_0001' (Duration to display results) parameter.

Best Practice: Set '0007_0001' (Duration to display results) to a value of ‘0’ to stop the displaying of the results screens the cardholder sees on the terminal (e.g., stops the flashing of multiple results screens).

34.x Save State and Restore Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “34.”
4	1	Alphanum	Sets the state: • S = Save the current state. • R = Restore the last saved state. • C = Clear the saved state.
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character.

There is an implied 'clear saved state' on a 10.x Hard Reset message.

34.x Save State and Restore Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “34.”

Offset	Length	Type	Description
4	1	Decimal	Status: • 0 = Successful. • 1 = Tried to restore without saved state. • 9 = Error.
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character.

6.2.37 35.x Menu Message (On-Demand)

The POS sends the 35.x Menu Message to prompt the terminal to display a custom menu on the terminal screen. The request specifies:

- A form name
- Prompt index or custom prompt text
- Index of the selected entry
- A series of up to 64 menu items (limited by the maximum 247 byte message length)

6.2.37.1 Limitations

- Unattended devices do not support the 35.x message.
- The 35.x message only populates form menus that create the list from `MENU_TEXT(n)` variables. It does not work with forms such as `EmvMenu.K3Z`:
  - The application selection form `EmvMenu.K3Z` uses the variable `EMVAIDLIST` to populate the menu selection list instead of searching `MENU_TEXT(n).`
  - A 35.x message calling `EmvMenu.K3Z` does not display menu items specified in the message.
- Special characters must be preceded with a backslash for the terminal to display them correctly in a menu list. The following characters are not supported:
  - &
  - +
  - ~

The following tables describe the 35.x request and response formats.

35.x Menu Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	Message Identifier – ASCII – “35.”
4	Variable	Alphanum	Form Name. Uses `menuCmd.k3z` if left blank

Offset	Length	Type	Description
M	1	Constant	ASCII Control Character – FS
M+1	Variable	Decimal	Prompt. Can be either text or index into prompt file
N	1	Constant	ASCII Control Character – FS
N+1	Variable	Decimal	Selected Entry Index. Index of entry to select at start
O	1	Constant	ASCII Control Character – FS
O+1	Variable	Alphanum	Menu Item 1. Text to display for the first item in the menu
P	1	Constant	ASCII Control Character – FS
P+1	Variable	Alphanum	Menu Item 2. Text to display for the second item in the menu
Repeat FS and menu item fields for each additional item in the menu list			
Z	1	Constant	ASCII Control Character – ETX
Z+1	1	Binary	LRC check character

35.x Menu Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	Message Identifier – ASCII – “35.”
4	1	Decimal	Result • 0 = Item selected • 1 = Key pressed • 2 = Cancel • 3 = Invalid form • 4 = Invalid item list • 5 = Invalid selected index • 6 = Invalid prompt • 9 = Declined

Offset	Length	Type	Description
5	Variable	Decimal	Data • If Result = 0, this field contains the index of the selected item • If Result = 1, this field contains the ID of the button pressed • If Result > 1, this field is excluded
M	1	Constant	ASCII Control Character – ETX
M+1	1	Binary	LRC check character

6.2.38 36.x Notification of Command Execution

6.2.38.1 Overview of the 36.x Notification of Command Message

The 36.x Notification of Command Execution message is sent from the terminal to the POS to indicate successful execution of a command. When the notification flag in the 17.x Merchant Data write message is set for a command sent , this feature is enabled. This message can also be used by the POS to monitor the tap process. The response 36.x message will include the two-digit command ID (in hex-ASCII format) which matches up with the command ID of the 17.x message. As an example, a '36.AA' message sent from the terminal to the POS indicates a command with an ID of "AA" has been successfully executed.

Refer to the 17.x Merchant Data Write section for 17.x message usage examples.

6.2.38.2 36.x Notification of Command Message Format

The following table describes the 36.x Notification of Command Execution message format.

36.x Notification of Command Execution Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M36_CONTACTLESS_NOTIFY Message Identifier – ASCII – “36.”
4	2	Alphanum	iConnectEFT Constant = P36_REQ_CLESS_CARD_CMD_ID_FOR_NOTIFY Command ID.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

6.2.38.3 Usage Example

Notification of Command Execution via the 36.x Message

Sequence	Message Content	POS	Terminal
1	17.5R10AA04	→	
2	36.AA	←	
3	17.500R000004000102030405060708090A0B0C0D0E0F	←	

6.2.39 37.x Rating Message

The POS sends the 37.x Rating Request to prompt the terminal to display a rating form. The terminal replies with a 37.x Rating Response containing either the cardholder's input or an error.

6.2.39.1 Form Limitations

The 37.x message must specify a form that:

- Uses the `TEMPLLD.HTM` template.
- Contains only text, graphics and buttons (other controls do not work properly).
- Uses the question label `SURVEY_QUES1`.

> Default forms include `SURQUES.K3Z` for use with the 37.x message.

37.x Rating Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – 37.
4	Variable	Alphanum	iConnect EFT Constant = P37_REQ_FORM_NAME Form name including file extension
M	1	Constant	ASCII control character – FS
M+1	Variable	Decimal	iConnect EFT Constant = P37_REQ_FORM_TIMEOUT Form display timeout • 0 = No timeout • >0 = Number of tenth-seconds to display form
N	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
N+1	Variable	Alphanum	iConnect EFT Constant = P37_REQ_QUESTION Survey question: • Literal text up to 230 characters (such as "Please swipe") • An index number from Prompt.xml Value cannot be 0.
O	1	Constant	ASCII control character – ETX
O+1	1	Constant	LRC check character

37.x Rating Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – 37.
4	Variable	Alphanum	iConnect EFT Constant = P37_RES_RESULT Result: • 0 = Key pressed • 1 = Invalid form requested • 2 = Security violation: Consecutive 37.x requests or time between requests has not expired. • 6 = Invalid prompt requested • 8 = Timeout expired • 9 = Terminal does not support this request (such as iUN terminals)
5	Variable	Alphanum	iConnect EFT Constant = P37_RES_DATA Data. If Result field is: • 0, this field contains the key pressed to exit rating. • Nonzero, this field contains 0.
M	1	Constant	ASCII control character – ETX
M+1	1	Constant	LRC check character

6.2.40 40.x Survey Messages

Survey messages can be posted on swipe card screens for the customer to answer. This section describes the messages available to control this feature.

6.2.40.1 40.0 Survey Request

This message sent from the POS to the terminal is used to prompt the customer with the button choices for the survey question displayed on the terminal. This message is preceded by one or more 40.x Survey Question messages, and is only valid if the terminal is currently displaying the swipe card screen.

The Survey Request message only supports requesting a display of the survey question configured for the currently active language (e.g., '1' = English, '2' = Spanish).

- Survey Request messages which attempt to override the currently active language (e.g., '40.1', '40.2') are considered invalid.
- Any other invalid message format found (e.g., '40.?', where "?" is any character other than '0') will return an error message.

Survey Request Format

Offset	Length	Type	Description
0	1	Constant	STX – 0x02
1	3	Constant	iConnectEFT Constant = M40_SURVEY Message identifier – “40.”
4	1	Decimal	iConnectEFT Constant= P40_REQ_TYPE • Type – “0”
5	1	Constant	ETX – 0x03
6	1	Binary	LRC check character

6.2.40.2 40.0 Survey Response

The terminal sends this message to the POS when the cardholder responds to survey question(s) or an error prevents the survey from ocurring.

Survey Response Format

Offset	Length	Type	Description
0	1	Constant	STX – 0x02
1	3	Constant	iConnectEFT Constant = M40_SURVEY Message identifier – “40.”

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P40_RES_STATUS Status: • 1 = Customer pressed first button. • 2 = Customer pressed second button. • 3 = Customer pressed third button. • C = Message canceled via 'Cancel' key. • D = Message not supported on current terminal (only iSC250/iSC350 are supported). • E = Message encounters any other generic error. • L = Survey Request message not valid for specified language. • Q = Question not configured for current language. • S = Message cancelled due to card swipe/tap/insert/etc. • T = Message not supported on current form.
5	1	Constant	ETX – 0x03
6	1	Binary	LRC check character

6.2.40.3 40.0 Survey Request Message Display Behavior

After receiving a '40.0' Survey Request message, the terminal will continue to display the survey until:

- answered by the customer
- canceled by the customer either by:
  - the 'Cancel' key
    Or:
  - swiping (tapping/inserting) their card
- interrupted by an on-demand message from the POS
  Or:
- canceled by the POS from an offline, online, or reset message.

The survey will be re-displayed after returning to the card swipe screen if interrupted by an on-demand message. However, it will not re-display following receipt of an offline, online, or reset message (which includes when the current transaction ends and before a subsequent transaction begins).

6.2.40.4 40.x Survey Question Request

This message sent from the POS to the terminal sets the survey question and button text that is displayed for a specific language. This message must be sent at least once for each available language, and must be sent at least once before a '40.0' Survey Request is made.

Survey Question Request Format

Offset	Length	Type	Description
0	1	Constant	STX – 0x02
1	3	Constant	iConnectEFT Constant = M40_SURVEY Message identifier – “40.”
4	1	Decimal	iConnectEFT Constant = P40_REQ_LANGUAGE The Survey Question message supports up to four languages: • 1 = Set text for language 1. • 2 = Set text for language 2. • 3 = Set text for language 3. • 4 = Set text for language 4.
5	Variable	Alphanum	iConnectEFT Constant = P40_REQ_QUESTION • Question (up to 150 characters)
M	1	Constant	ASCII control character – FS (0x1C)
M + 1	Variable	Alphanum	iConnectEFT Constant = P40_REQ_BUTTON1 • Button 1 text (up to 20 characters)
N	1	Constant	ASCII control character – FS (0x1C)
N + 1	Variable	Alphanum	iConnectEFT Constant = P40_REQ_BUTTON2 • Button 2 text (up to 20 characters)
O	1	Constant	ASCII control character – FS (0x1C)
O + 1	Variable	Alphanum	iConnectEFT Constant = P40_REQ_BUTTON3 • Button 3 text (up to 20 characters)
P	1	Constant	ETX – 0x03
P + 1	1	Binary	LRC check character

6.2.40.5 40.x Survey Question Response

This message sent from the terminal to the POS shows the result of a Survey Question message.

Survey Question Response Format

Offset	Length	Type	Description
0	1	Constant	STC – 0x02
1	3	Constant	iConnectEFT Constant = M40_SURVEY Message identifier – “40.”
4	1	Decimal	iConnectEFT Constant = 40_RES_LANGUAGE Language: • 1 = Set text for language 1. • 2 = Set text for language 2. • 3 = Set text for language 3. • 4 = Set text for language 4.
5	1	Alphanum	iConnectEFT Constant = P40_RES_STATUS Status: • 0 = OK; Survey Question message parsed without error. • 1 = Button 1 text is too long (> 20 characters), or contains invalid characters. • 2 = Button 2 text is too long (> 20 characters), or contains invalid characters. • 3 = Button 3 text is too long (> 20 characters), or contains invalid characters. • B = All button text fields have length of 0 (zero) characters. • E = Survey Question message has any other generic error. • L = Survey Request message not valid for specified language. • Q = Survey Question contains invalid characters, is 0 (zero) characters in length, or is too long (more than 150 characters). • ? = Survey Question message has any other invalid characters. A survey button is hidden when button text length = 0 characters.
6	1	Constant	ETX – 0x03
7	1	Binary	LRC check character

Additional Survey Question Message Characteristics

Offset	Description
Survey Question message fields	May ONLY include: - Printable characters, including 8-bit extended ASCII characters. - Single space character. May not include: - Invalid control characters (e.g., group separator 'GS' character '01xD'). - Non-printable characters. - Any whitespace characters EXCEPT for the space character (e.g., tab, form feed, line feed, carriage return).
Language Preservation	All 40.x Survey Questions for all languages are cleared at terminal boot. Once set, the 40.x Survey Question for the specific language is set: - Across all transactions and 40.0 Survey Requests. - Until a new and valid 40.x Survey Question for the specific language is set. Any invalid 40.x Survey Question messages will not clear the currently configured 40.x Survey Question for any language.

Sample Invalid 40.x Message Formats

Invalid message characteristic	Invalid message examples
End of the message before valid question and 3 button text fields are found	Input: `40.1Question` Response: `40.1?`
More than 3 button text fields are found	Input: `40.1What is your dog's name? [FS]Fluffy[FS]Spot[FS]Harvey[FS]Curly` Response: `40.1?`

Invalid message characteristic	Invalid message examples
All button text fields have a length of 0 (zero) characters	Input: `40.1What is your dog's name?[FS][FS][FS]` Response: `40.1B`
Any invalid characters are found in the Question field or any of the button fields	Input: `40.1[GS][FS]x[FS]y[FS]z` (In this example, ‘1’ is the language, ‘[GS]’ is the invalid character in the Question field.) Response: `40.1Q` Input: `40.1What is your dog’s name?[FS]x[FS][GS][FS]z` (In this example, ‘1’ is the language, ‘[GS]’ is the invalid character in the second button’s text.) Response: `40.12` (If buttons 2 and 3 both have invalid text, the message will display the number corresponding to the first button (from the left) that contains the error.)
Invalid or no language exists	Input: `40.Best Service Ever?[FS]Yes[FS]No[FS]Maybe` (In this example, no language is defined.) Response: `40.BL` (In this response, the ‘B’ represents the first letter of the question, not the ‘B’ status.) Input: `40.What is your favorite ice cream flavor?[FS]Vanilla[FS]Chocolate[FS]Strawberry` (In this example, no language is defined.) Response: `40.WL` Input: `40.5What is your favorite ice cream flavor?[FS]Vanilla[FS]Chocolate[FS]Strawberry` (In this example, a #5 language is defined, which is not a valid language.) Response: `40.5L`
No question included	Input: `40.1[FS]x[FS]y[FS]z` Response: `40.1Q`

6.2.41 41.x Card Read Message

6.2.41.1 Overview

The 41.x Card Read Request message enables or disables the following terminal card readers:

- MSR
- Contactless
- Smart card (EMV)

If the terminal supports LEDs for the card readers, they illuminate as appropriate.

- This is not an on-demand message.

6.2.41.2 Smart Card Reader Support

The EMV Transactions Supported flag 0019_0001 must be set to 1 enable the smart card reader. When a 41.1001 request to enable the smart card reader is received and the EMV Transactions Supported flag is set to support EMV transactions, then the smart card reader will be enabled as well. If a smart card is inserted, a 09.020201I message indicating smart card insertion is sent to the POS. The terminal responds with a 41.S message, indicating the smart card reader is enabled.

6.2.41.3 41.x Message Format

The following tables describe the 41.x Card Read Request Message format and 41.x Card Read Response Message format. When a card is read by any of the card readers, all card readers are disabled, and a message is returned to the POS.

41.x Card Read Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M41_CARD_READ_REQUEST Message Identifier– ASCII – 41.
4	1	Alphanum	iConnectEFT Constant = P41_REQ_PARSE_FLAG Indicates if parsed data fields should be returned. • 0 = Do not return parsed fields. • 1 = Return parsed fields.

Offset	Length	Type	Description
5	1	Alphanum	iConnectEFT Constant = REQ_MSR_FLAG MSR enable. • 0 = Disable MSR. • 1 = Enable MSR. • 2 = Do not change MSR enable status.
6	1	Alphanum	iConnectEFT Constant = REQ_CONTACTLESS_FLAG Contactless enable. • 0 = Disable contactless. • 1 = Enable contactless. • 2 = Do not change contactless enable status.
7	1	Alphanum	iConnectEFT Constant = REQ_SMC_FLAG SMC enable. • 0 = Disable SMC. • 1 = Enable SMC. • 2 = Do not change SMC enable status.
8	1	Constant	ASCII control character – ETX
9	1	Binary	LRC check character

41.x Card Read Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M41_CARD_READ_REQUEST Message Identifier– ASCII – 41.

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = RES_SOURCE Source: • C = Contactless • E = Contactless EMV • M = MSR • P = Problem • Q = Fast quick chip • S = Smart card (SMC) • U = Unknown All fields following the RES_SOURCE field can be empty, especially if the card source is Unknown.
5	2	Alphanum	iConnectEFT Constant = RES_ENCRYPTION Encryption type: • 00 = Encryption disabled • 01 = Magtek MagneSafe POS • 02 = Ingecrypt • 03 = EPS • 04 = Voltage TEP1 • 05 = Voltage TEP2 • 06 = Voltage TEP3 (not currently supported) • 07 = Monetra Cardshield • 08 = Mercury • 09 = RSA-OAEP • 10 = TransArmor • 11 = TDES DUKPT Generic • 12 = S1
7	1	Alphanum	iConnectEFT Constant = RES_TRACK_1_STATUS Track 1 status: • 0 = Good read • 1 = Error or empty track
8	1	Alphanum	iConnectEFT Constant = RES_TRACK_2_STATUS Track 2 status: • 0 = Good read. • 1 = Error or empty track.

Offset	Length	Type	Description
9	1	Alphanum	iConnectEFT Constant = RES_TRACK_3_STATUS Track 3 status. • 0 = Good read. • 1 = Error or empty track.
10	Variable	Alphanum	iConnectEFT Constant = RES_TRACK_1 Track 1 data.
M	1	Alphanum	ASCII control character – [FS]
M + 1	Variable	Alphanum	iConnectEFT Constant = RES_TRACK_2 Track 2 data.
N	1	Alphanum	ASCII control character – [FS]
N + 1	Variable	Alphanum	iConnectEFT Constant = RES_TRACK_3 Track 3 data.
O	1	Alphanum	ASCII control character – [FS] (only if parsed)
O + 1	Variable	Alphanum	iConnectEFT Constant = RES_PAN PAN (only if parsed)
P	1	Alphanum	ASCII control character – [FS] (only if parsed)
P + 1	Variable	Alphanum	iConnectEFT Constant = RES_MASKED_PAN Masked PAN (only if parsed)
Q	1	Alphanum	ASCII control character – [FS] (only if parsed)
Q + 1	Variable	Alphanum	iConnectEFT Constant = RES_EXPIRATION_DATE Expiration Date (only if parsed)
Q + 5	1	Alphanum	ASCII control character – [FS] (only if parsed)
Q + 6	Variable	Alphanum	iConnectEFT Constant – RES_ACCOUNT_NAME Account Name (only if parsed) This field is left empty when 0091_0038 is set to 1.
R	1	Constant	ASCII control character – ETX
R + 1	1	Binary	LRC check character

6.2.42 50.x Authorization Request

The 50.x Authorization Request message format is defined by the **VISA Second Generation** specification. This message can be sent by the terminal or by the POS. Each message has a different meaning and format. This section describes the messages sent from the terminal to the POS. The received message is described in 0x and 50.x Authorization Response Message Format.

When all required data is collected from both the customer and the POS, the terminal uses that information to create the 50.x Authorization Request message. This message is sent to the POS out unsolicited. When this message is acknowledged by the POS, the terminal proceeds to the next state and waits for the 50.x Authorization Response message from the POS for the transaction approval/decline decision. If the 50.x Authorization Request message is not acknowledged, then the message will be resent up to three times. If no response from the POS is received after the third attempt, then the terminal displays the message "Authorization Request Not Sent" for three seconds and terminates the transaction. An advertisement will be displayed on the screen (if enabled).

50.x Authorization Request Message Format (Sent from Terminal)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M50_AUTHORIZATION Message Identifier – ASCII – “50.”
4	6	Alphanum	iConnectEFT Constant = P50_REQ_ACQUIRING_BANK Acquiring Bank Number.
10	12	Alphanum	iConnectEFT Constant = P50_REQ_MERCHANT_ID Merchant ID Number.
22	4	Alphanum	iConnectEFT Constant = P50_REQ_STORE Store ID Number.
26	4	Alphanum	iConnectEFT Constant = P50_REQ_PIN_PAD Terminal ID Number.
30	4	Alphanum	iConnectEFT Constant = P50_REQ_STANDARD_INDUSTRY_CLASSIFICATION Standard Industry Classification.
34	3	Alphanum	iConnectEFT Constant = P50_REQ_COUNTRY_OR_CURRENCY_TYPE Country / Currency Type.

Offset	Length	Type	Description
37	5	Alphanum	iConnectEFT Constant = P50_REQ_ZIP_CODE Zip Code.
42	3	Alphanum	iConnectEFT Constant = P50_REQ_TIME_ZONE_DIFF_FROM_GMT Time Zone different from GMT.
45	2	Alphanum	iConnectEFT Constant = P50_REQ_TRANSACTION_CODE Transaction Code.
47	8	Alphanum	iConnectEFT Constant = P50_REQ_PIN_PAD_SERIAL_NUM Terminal Serial Number.
55	1	Constant	Index Code (always "0")
56	4	Alphanum	iConnectEFT Constant = P50_REQ_POS_TRANSACTION_NUM POS Transaction Number.
60	1	Constant	Message Status Code (always "@")

Offset	Length	Type	Description
61	1	Alphanum	iConnectEFT Constant = P50_REQ_ACC_DATA_SOURCE Account Data Source: • P = Phone Number (used with PayPal). • H = Electronic Track 1. • D = Electronic Track 2. • B = Electronic Tracks (both 1 & 2). • h = Contactless Track 1. • d = Contactless Track 2. • b = Contactless Tracks (both 1 & 2). • R = Manual entry Track facsimile (both Track 1 and Track 2). • X = Manual entry Track 1. • T = Manual entry Track 2. • A = account_data_source (when account information is sent via 12.x Account Message). The Contactless indicators can be configured in the Contactless Reader Configuration (cless.dat) file where parameter '0008_0006' handles 'h', parameter '0008_0007' handles 'd', and parameter '0008_0005' handles 'b'.
62	Variable	Alphanum	iConnectEFT Constant = P50_REQ_MAG_SWIPE_INFO Magnetic Stripe Information: Track 1, Track 2, or manual entry. If there is more than one track present, the track data will be separated by a "[FS]" field separator character.
M	1	Constant	ASCII control character – FS
M + 1	2, 23, or 43	Alphanum	PIN Information: See Table "PIN Block Format" further.
N	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
N + 1	Variable	Decimal	iConnectEFT Constant = P50_REQ_TRANSACTION_AMOUNT Transaction Amount in cents (Minimum of 3 digits). If an amount is less than 3 digits in length it shall be prepended by zeros. As an example, '33' cents will be sent up as '033'.
O	1	Constant	ASCII control character – FS
O + 1	Variable	Alphanum	ETB (Optional) Voltage ETB Base64 encoded. Included if '0091_0008' = '1' and Voltage encryption is enabled.
P	1	Constant	ASCII control character – FS (included only with Token value)
P + 1	Variable	Alphanum	Token value generated by the terminal (Optional)
Q	1	Constant	ASCII control character – ETX
Q + 1	1	Binary	LRC check character.

Sample Requests

Card Source	Sample Request
Contactless	50.1234567890123456789012345678901234567890020700055830001@d4005578000000150= 10121015555554400751[FS]1@[FS]1025[FS]
MSR / Card Swipe	50.1234567890123456789012345678901234567890020700055830002@D4005578000000150= 10121015555540600761[FS]1@[FS]1025[FS]
Account Message	50.12345678901234567890123456789012345678900208049229800001@T4445222299990007=1214[FS]1@[FS]1025[FS]

PIN Block Format

Offset	Length	Type	Description	No PIN	M/S PIN	DUKPT PIN
0	2	Constant	ASCII character – "1@"	X	X	X
2	1	Constant	ASCII control character – CR		X	X
3	2	Constant	iConnectEFT Constant = P50_REQ_PIN_LENGTH Entered PIN Length. • Always '12'.		X	X
5	2	Constant	PIN Block Format. • Always '01'.		X	X
7	16	Alphanum	iConnectEFT Constant = P50_REQ_PIN_ENCRYPTED_BLOCK Encrypted PIN block.		X	X
23	4	Constant	ASCII characters – For example: • FFFF • F876			X
27	6	Alphanum	iConnectEFT Constant = P50_REQ_PIN_KEY_SET_IDENTIFIER Key set identifier (KSI).			X
33	5	Alphanum	iConnectEFT Constant = P50_REQ_PIN_DEVICE_ID Terminal ID (-1 bit).			X
38	5	Alphanum	iConnectEFT Constant = P50_REQ_PIN_ENCRYPTION_COUNTER Encryption counter (+ 1 bit).			X

6.2.43 51.x Beep On-Demand Message

The 51.x Beep On-Demand message is used to set the tone and tone duration for beep on-demand. The following table describes the message format.

51.x Beep On-Demand Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “51.”
4	1	Decimal	iConnectEFT Constant = P51_REQ_TONE Tone selection: • 0 = Low tone (4200Hz). • 1 = Middle tone (6000Hz). • 2 = High tone (9000Hz). • 3 = Error tone (3800Hz).
5	1	Decimal	iConnectEFT Constant = P51_REQ_TIME Tone duration: • 0 = Click length (150 ms). • 1 = Short length (500 ms). • 2 = Long length (1000 ms). • 3 = Error length (500 ms).
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

51.x Beep On-Demand Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “51.”
4	1	Decimal	iConnectEFT Constant = P51_RES_STATUS Success status: • 0 = Success. • 1 = Message error. • 2 = Device error (currently deprecated).
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character.

6.2.44 52.x PayPal Preauthorization Message

RBA sends the 52.x PayPal Preauthorization message to the POS after the terminal has collected the information needed to identify the cardholder so the POS can retrieve any coupons or discounts from the cardholder's PayPal account. The POS can apply these discounts to the purchase before sending the total to the application for verification.

52.x PayPal Preauthorization Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M52_PAYPAL_PREAUTHORIZE Message Identifier - ASCII – "52."
4	1	Alphanum	iConnectEFT Constant = P52_ACCOUNT_DATA Account Data Source: • P = Phone Number (used with PayPal). • H = Electronic Track 1. • D = Electronic Track 2. • B = Electronic Tracks (both 1 & 2). • h = Contactless Track 1. • d = Contactless Track 2. • b = Contactless Tracks (both 1 & 2). • X = Manual Track 1. • T = Manual Track 2. The Contactless indicators can be configured in the `config.dfs` file where '0008_0006' handles 'h', '0008_0007' handles 'd', and '0008_0005' handles 'b'.
5	Variable	Alphanum	iConnectEFT Constant = P52_MAGNETIC_STRIPE_1 Magnetic Stripe Track 1.
M	1	Constant	ACSII control character – FS
M + 1	Variable	Alphanum	iConnectEFT Constant = P52_MAGNETIC_STRIPE_2 Magnetic Stripe Track 2.
N	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
N + 1	Variable	Alphanum	iConnectEFT Constant = P52_PAYPAL_PIN_BLOCK PayPal PIN block.
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character.

6.2.45 58.x Terminal Discovery Message

6.2.45.1 Overview

The application opens a UDP port when the following conditions are met:

- Ethernet is selected as the communication method
- SSL is enabled on the terminal, and the correct server `.pgz` file is loaded
- The terminal connection is open and not in use

When the POS:

- Connects to a terminal over the TCP, the UDP connection is closed.
- Closes the TCP connection with the terminal, the UDP connection for the 58.x Terminal Discovery message re-opens.

The UDP port uses the same port number that is used for the TCP connection with the POS.

The 58.0 Terminal Discovery request message is senr from the POS to the terminal. The POS uses this message to request the following terminal information:

- Serial number
- MAC ID number
- IP address

When the terminal receives the 58.0 request message, it sends a 58.x response message with this information.

> The Serial Number is the injected serial. If unavailable, the hardware serial number is inserted.

6.2.45.2 Usage Examples

If the POS is connected to a network with multiple terminals. When the POS sends the 58.0 request message, all terminals return a 58.x response message containing their serial number and IP address. The POS can extract the IP addresses from these messages and initiate communications. The format of the 58.0 request message is as follows:

[STX]58.0[ETX][LRC]

The format of the response message is:

[STX]58.SerialNumber[FS]MacAaddress[FS]IpAddress[ETX][LRC]

where SerialNumber is the injected serial number or the terminal serial number.

As an example, a terminal returns a 58.x message with the following information:

[0x02]58.80377780[FS]54:7f:54:aa:6a:03[FS]192.168.17.145[0x03][0x37]

where

- Serial number = 80377780
- MAC address = 54:7f:54:aa:6a:03
- IP address = 192.168.17.145

In the following example, two terminals are connected to the POS, which sends the 58.0 request message and receives the following responses:

- Terminal 1 responds with [0x02]58.71081574[FS]54:7f:54:1a:7c:bb[FS]192.168.0.109[0x03][0x55]
- Terminal 2 responds with [0x02]58.80377752[FS]54:7f:54:aa:69:e7[FS]192.168.0.106[0x03][0x31]

The serial numbers and IP addresses for these terminals are as follows:

- Terminal 1: Serial number = 71081574, IP address = 192.168.0.109
- Terminal 2: Serial number = 80377752, IP address = 192.168.0.106

6.2.45.3 Terminal Discovery Message Format

The following tables describe the formatting for the 58.0 request and 58.x response messages:

58.0 Terminal Discovery Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M58_DISCOVER_DEVICES Message Identifier - ASCII – 58
4	1	Alphanum	iConnectEFT Constant = P58_REQ_ACTION • Always 0
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character

58.x Terminal Discovery Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M58_DISCOVER_DEVICES Message Identifier - ASCII – 58

Offset	Length	Type	Description
4	Variable	Alphanum	iConnectEFT Constant = P58_RES_SERIAL_NUMBER Injected serial number if present; otherwise the terminal serial number
M	1	Constant	ASCII control character - FS
M + 1	Variable	Alphanum	iConnectEFT Constant = P58_RES_MAC_ADDRESS Terminals unique MAC ID number. Format is YY:YY:YY:YY:YY:YY
N	1	Constant	ASCII control character - FS
N + 1	Variable	Alphanum	iConnectEFT Constant = P58_RES_IP_ADDRESS Terminals IP address. Format is ZZZ.ZZZ.ZZZ.ZZZ
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character

6.2.46 60.x Configuration Write

6.2.46.1 Overview

- The 60.x Configuration Write message is used to permanently change configuration parameters and the display prompts in its data file system (DFS) memory. This single message accepts a variable number of IDN blocks.
- The 60.x message can accept many IDN blocks. The total message length may not exceed the maximum acceptable message length (240 bytes). The application returns 60.x response to each 60.x request.
- The 60.x message is accepted by the application only in the offline state. The values are stored in RAM until either a 01.x Online or a 00.x Offline is received. Configuration settings are then written to Flash memory.
- The IDN blocks are separated from each other by FS (field separator) value 0x1C. Data fields inside the block are separated with the group separator GS, value 0x1D.
- Response to 60.x messages are sent after writing to the DFS memory is finished, and RAM value is updated. Time for the response message might vary. Therefore it is recommended to keep a small number of configuration IDN blocks (grpNum + inxNum + data) from the same group in a single 60.x message. If timing from 60.x response is not a concern, the 60.x may be long.
- If an error is detected in one of the blocks, the rest of the message is not executed. When there are no errors in the IDN block, data from the block is saved in DFS memory. When an error is detected, data from that block is not written to the DFS and the rest of the 60.x message is ignored. Writing a data value to the file system overwrites the current value. 61.x reading a value before writing is the only way to have an original copy of the value after a write is done via 60.x message. 61.x messages can be used to verify the value of the configuration parameter.

- The 0007_0044 Country Terminal parameter should be set in the mainflow.dat file. If it is set with the 60.x Set Parameter message, a reboot is required for the country change to be processed correctly.

Refer to the following example where a parameter is permanently changed using the 60.x Configuration Write message. This means that the parameter will retain its new value following reboot.

1. POS sends 00.x Offline Message to terminal.
2. Terminal goes offline.
3. POS sends 60.10[GS]2[GS]4 message to terminal.
4. Terminal responds with 60.2 success message.
5. POS sends 61.10[GS]2[GS] message.
6. Terminal responds with 60.210[GS]2[GS]4 message confirming changed parameter.

Group 0 and index 0 are not valid selections.

60.x Configuration Write Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M60_CONFIGURATION_WRITE Message Identifier - ASCII – 60.
4	Variable	Alphanum	iConnectEFT Constant = P60_REQ_FILE_NUM_IDN_BLOCK File number of IDN block #1
M	1	Constant	ASCII control character – GS
M + 1	Variable	Alphanum	iConnectEFT Constant = P60_REQ_INDEX_NUM_IDN_BLOCK Index number of IDN block #1
N	1	Constant	ASCII control character – GS
N + 1	Variable	Alphanum	iConnectEFT Constant = P60_REQ_DATA_CONFIG_PARAM Data of config parameter #1
O	1	Constant	ASCII control character – FS
O + 1	Variable	Alphanum	iConnectEFT Constant = P60_REQ_GROUP_NUM File number of IDN block #2 (optional)
P	1	Constant	ASCII control character – GS
P + 1	Variable	Alphanum	iConnectEFT Constant = P60_REQ_INDEX_NUM Index number of block #2 (optional)

Offset	Length	Type	Description
Q	1	Constant	ASCII control character – GS
Q + 1	1	Constant	iConnectEFT Constant = P60_REQ_DATA_CONFIG_PARAM Data of config parameter #2 (optional)
R	1	Constant	ASCII control character – FS
R + 1			... (optional, more configuration settings)
S	1	Constant	ASCII control character – ETX
S + 1	1	Binary	LRC check character

60.x Configuration Write Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M60_CONFIGURATION_WRITE Message Identifier – ASCII – 60.
4	1	Decimal	iConnectEFT Constant = P60_RES_STATUS Response status: • 1 = Failed, unknown error. • 2 = Successful. • 3 = Error, one or more of parameters has invalid ID. • 4 = Error, one or more of parameters was not updated. • 5 = message rejected, wrong message format. • 9 = message rejected, cannot be executed.
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character

6.2.46.2 60.x Variations

val1 GS val2 GS val3 FS

x1D x1D

Val1 – Val3

Value	Description
Val1	File number: Example: msr.dat number = 0003, pin.dat = 0006, prompt1.SPA = 0032.
Val2	Index of parameter in a file specified by val2.
Val3	Data string.

Warning
After changing all global parameters be sure to do the following to preserve the changes:

1. Send online 01.00000000 message
2. Send offline 00.0000 message
3. Power cycle the terminal so that all of the applications installed in the terminal will be updated with the new values. by sending 97.x message

Examples:
Disable display of advertisements.
10 GS 1 GS 0
Set global cash back limit for all payment types to 10000 cents (100.00 dollars).
2 GS 1 GS 10000

6.2.47 61.x Configuration Read

The POS sends the 61.x Configuration Read request to the terminal to retrieve the setting of a configuration parameter specified by its DFS index. A 61.x Configuration Read Response message is returned with status (successful, error, and so on), DFS index, and the value of the specified parameter.

6.2.47.1 DFS Index

Every DFS index consists of two parts. In parameter 0007_0001 for example,

- 0007 is the IDN block, indicating the parameter is in mainflow.dat
- 0001 is the index, indicating it is the first parameter in mainflow.dat

Because some DFS indices are reserved or deprecated, some index values may be skipped. See Configuring the Application for each parameter's DFS index value.

6.2.47.2 Usage Examples

Consider the following examples:

1. The POS sends a '61.7[GS]1' request indicating that the parameter to be read is located at IDN block 7 and index number 1 (0007_0001).
2. The terminal responds with '61.27[GS]1[GS]30' indicating a successful read operation, confirming the DFS index number, and including the parameter setting '30'.

1. The POS sends a '61.5[GS]2' request indicating that the parameter to be read is located at IDN block 5 and index number 2 (0005_0002).
2. The terminal responds with '61.25[GS]2[GS]1' indicating a successful read operation, confirming the the DFS index number, and including the parameter setting '1'.

Reading multiple configuration setting (Example):

1. The POS sends a '61.7[GS]1[FS]19[GS]1' message to request 0007_0001 and 0019_0001 configuration parameter settings.
2. The terminal responds with a '61.27[GS]1[GS]30[FS]19[GS]1[GS]0' message indicating a successful read operation.
3. The POS then confirms the IDN block and index number, and the setting for 0007_0001 (30) and 0019_0001 (0).

The following tables describe the 61.x Configuration Read Request and 61.x Configuration Read Response message formats.

61.x Configuration Read Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M61_CONFIGURATION_READ Message Identifier – ASCII – “61.”
4	Variable	Alphanum	iConnectEFT Constant = P61_REQ_GROUP_NUM_IDN_BLOCK Group number of IDN block #1
M	1	Constant	ASCII control character – GS
M + 1	Variable	Alphanum	iConnectEFT Constant = P61_REQ_INDEX_NUM_IDN_BLOCK Index number of IDN block #1.
N	1	Constant	ASCII control character – FS (optional, included only with additional parameters)
N + 1	Variable	Alphanum	iConnectEFT Constant = P61_REQ_GROUP_NUM_IDN_BLOCK Group number of IDN block #2 (optional)
O	1	Constant	ASCII control character – GS (optional)
O + 1	Variable	Alphanum	iConnectEFT Constant = P61_REQ_INDEX_NUM_IDN_BLOCK Index number of IDN block #2 (optional)
M	1	Constant	ASCII control character – FS (optional)

Offset	Length	Type	Description
P			... (optional, in case of multiple parameters)
Q	1	Constant	ASCII control character – ETX
Q + 1	1	Binary	LRC check character.

61.x Configuration Read Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M61_CONFIGURATION_READ Message Identifier – ASCII – "61."
4	1	Decimal	iConnectEFT Constant = P61_RES_STATUS Response status: • 1 = Failed, unknown error. • 2 = Successful. • 3 = Error, one or more of parameters has invalid ID. • 4 = Error, one or more of parameters was not updated. • 5 = Message rejected, wrong message format. • 9 = Message rejected, cannot be executed.
5	Variable	Alphanum	iConnectEFT Constant = P61_RES_GROUP_NUM Group number of IDN block #m.
M	1	Constant	ASCII control character – GS
M + 1	Variable	Alphanum	iConnectEFT Constant = P61_RES_INDEX_NUM Index number of IDN block #m.
N	1	Constant	ASCII control character – GS
N + 1	Variable	Alphanum	iConnectEFT Constant = P61_RES_DATA_CONFIG_PARAMETER Data of config parameter #m.
O	1	Constant	ASCII control character – FS (optional, in case of multiple parameters requested)
O + 1			... (optional, in case of multiple parameters)

Offset	Length	Type	Description
P	1	Constant	ASCII control character – ETX
P + 1	1	Binary	LRC check character.

The message response length is limited to 2000 bits. If response goes over the bit limit, it will return error code “4”.

6.2.48 62.x File Write

The POS sends the 62.x File Write message to write a file to terminal storage. Image files must be in .bmp, .jpg, .png, or .gif format.

The application capitalizes all letters in file names included in the 63.x Find File request from the POS. All letters in file names must be capital. The 63.x request does not find files named using lowercase characters.

If the terminal does not automatically reboot after a file update, manually reboot the terminal. The following table indicates which file types require a manual reboot:

File Type Updated	Automatic Reboot: Terminal reboots automatically after update	Manual Reboot: Send 97.x Message or press Terminal key combination for manual reboot
.OGZ	X	
.PGZ		X
.TGZ		X
.K3Z		X

6.2.48.1 Handling Large Files

Large files take more time to download using the 62.x message, which is intended to update single files. Large files can be uploaded much more rapidly using IBMEFT download or TMS. An EFTL file created from an .OGZ downloads in considerably less time than the time required for the 62.x File Write with most communications.

6.2.48.1.1 Interruptions

Loading large files using 62.x might take several minutes for the terminal to process the download before rebooting and unpacking the downloaded data. If the user attempts to reboot the terminal before the download is completed, the file will not be updated.

6.2.48.2 Message components

Included within the 62.x request are:

- **Record Type**: If a file is:
  - 4079 or fewer bytes in length (3569 or fewer for seven-bit encoding) including the length of filename, it can be written using a single message of Record Type 0 (according to the following **File Write Request** table)
  - Longer, multiple messages must be used to send the entire file. The first record must be Record Type 1
    - If a file requires more than two records, some number of Record Type 2 records will follow. The file is finished by sending a Record Type 3

- **Encoding Format**: To avoid confusing a protocol and because some systems only use seven data bits, the data must be encoded. Two methods are supported:
  - The more efficient method requires an eight-bit data path (refer to the **Eight-Bit Encoding** table in this section)
  - If only seven bits are supported, use the seven-bit encoding (refer to the **Seven-Bit Encoding** table in this section)
    - Since the file name field can contain a path and a file name, the amount of data in a Record Type 0 or Record Type 1 may have to be reduced in order to keep the total message size 247 bytes or fewer

- **ASCII File Segment Number:** The first two bytes in the six-byte reserved data field contain an ASCII file segment number. These two bytes effectively functions as a two-digit decimal value ranging from 01 to 99. When the segment number reaches 99 it the next value is 01

6.2.48.3 Best Practices

Ingenico recommends using a block size of 4kB or 2kB if your environment does not allow for 4kB messages. A terminal uses 2kB memory blocks to allocate transferred data to ensure no memory is wasted during transfers.

6.2.48.4 Setting the RKIVERSION during Download of .rki Files

Record Type option 4 supports setting the RKIVERSION. When selected:

- The encoding format is ignored and the RKIVERSION string is provided in the File Data field
- The 62.1 message must precede the 62.4 message to start the download
- A 62.3 message must end the download

> This method of updating the RKIVERSION during a file download cannot be used if the .RKI file is sent in one packet with the 62.0 message. Instead, divide the file into two smaller packets with a 62.1 and 62.3 or send all of the data in the 62.1 message, and send an empty 62.3 message. The 62.4 message must be sent between the 62.1 and 62.3 messages as described for large .RKI files.

6.2.48.5 Aborting the Previous File Download

A New File (62.0 or 62.1) request or 62.9 Abort request causes the application to abort any previously active file download.

The following tables describe the 62.x request and response format.

62.x File Write Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M62_FILE_WRITE Message Identifier – ASCII – 62.
4	1	Alphanum	iConnectEFT Constant = P62_REQ_RECORD_TYPE Record Type: • 0 = New file: All data is in this record • 1 = New file: Initial record of multiple • 2 = Continuation (must follow a 62.1 or 62.2) • 3 = Last record (must follow a 62.1 or 62.2) • 4 = Set RKIVERSION (excludes offsets 5 - M) • 9 = Abort current File Write request (excludes offsets 5 - M+1)
5	1	Alphanum	iConnectEFT Constant = P62_REQ_ENCODING_FORMAT Encoding format. • 7 = Seven-bit encoding • 8 = Eight-bit encoding
6	2	Alphanum	iConnectEFT Constant = P62_REQ_SEGMENT_NBR ASCII file segment number. • Value ranges from 00 to 99 • Segment number starts at 01 and increments by 1 • Segment number wraps around to 01 after 99
8	4	Alphanum	iConnectEFT Constant = P62_REQ_RESERVED • Reserved, set to 0000
12	1	Binary	iConnectEFT Constant = P62_REQ_UNPACK_FLAG Determines whether to unpack the file or load to HOST as-is. • 0 = Download file, unpack, and verify signing before sending to HOST (for OGZ and Application files only) • 1 = Download file and send to HOST as-is (accepts any file type) .K3Z, .TGZ, and .PGZ files require a reboot following download.

Offset	Length	Type	Description
13	1	Binary	iConnectEFT Constant = P62_REQ_FAST_DOWNLOAD Enables fast download feature. Allowable values are: • 0 = Standard download speed • 1 = Fast download speed If the fast download feature value in the message is set to 1: • The application ACKs each message but does not send a 62.0 response. • After the download completes, the terminal sends one 62.x response for all 62.x requests indicates whether the download was successful.
14	Variable	Alphanum	iConnectEFT Constant = P62_REQ_FILE_NAME orP62_REQ_OS_FILE_NAME File name (only if Record Type 0 or Record Type 1)
M	1	Constant	ASCII control character – FS (only if Record Type 0 or Record Type 1)
M+1	Variable	Alphanum	iConnectEFT Constant = P62_REQ_FILE_DATA File data (see the following tables) The RKIVERSION string is provided in this field when option 4 (Set RKIVERSION) is selected in the Record Type field. The RKIVERSION string consists of 1 to 4 characters.
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

62.x File Write Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M62_FILE_WRITE Message Identifier – ASCII – “62.”

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P62_RES_STATUS Response status: • 0 = Successful • 1 = Request out of order • 2 = File input/output error. For example, filename exceeds 15 characters. Returned even during fast download. • 3 = Data error • 8 = Error unzipping file • 9 = Abort current file
5	Variable	Alphanum	iConnectEFT Constant = P62_RES_FILE_LENGTH File length – only included in response to last packet
M	1	Constant	ASCII control character – ETX
M + 1	Variable	Binary	LRC check character.

Eight-Bit Encoding

Original Byte	Encoded 1st Byte	Encoded 2nd Byte	Example: Original	Example: Encoded
0h – 1Fh	FFh	Original + 20h	10h	FFh 30h
20h – FEh	Original	None	30h	30h
FFh	FFh	FFh	FFh	FFh FFh

Seven-Bit Encoding

Original Byte	Encoded 1st Byte	Encoded 2nd Byte	Example: Original	Example: Encoded
0h – 1Fh	7Dh	Original + 20h	10h	7Dh 30h
20h – 7Ch	Original	None	30h	30h
7Dh – 9Fh	7Eh	Original – 20h	80h	7Eh 60h
A0h – FFh	7Fh	Original – 80h	E0h	7Fh 50h

6.2.48.6 Clean PIN Pad: CLEANPINPAD.PGN

6.2.48.6.1 Overview

The Clean PIN Pad feature removes unused application files during a software update to:

- Support devices that might be approaching their memory limits
- Remove unnecessary or unwanted components from the terminal

The Clean PIN Pad feature removes only application files, such as RBA, drivers, and libraries. It does NOT remove any data files, such as /HOST form files, prompt files, or .dat configuration files; however, the latest version of RBA overwrites the existing RBA default data files.

A terminal can be cleaned of old files using one of the following methods: Whitelist and Clean-All-Applications.

Whitelist Request

A clean whitelist request clears all application files NOT listed in a WHITELIST.TXT from the terminal following the download of the latest RBA release. The WHITELIST.TXT must be included as an extra data file in the RBA DATA7362.PGN file and lists ALL files in the release.

- SYSTEM;8440220142.PGN;0000
- SYSTEM;8205000016.DGN;5155
- SYSTEM;8296270011.LGN;CBDB
- SYSTEM;8205011948.DGN;20B2
- SYSTEM;8440210142.PGN;0000
- SYSTEM;8296280011.LGN;91A4
- SYSTEM;8295380194.AGN;0C06
- SYSTEM;DATA7362.PGN;0000

Add the following RBA parameter files to the WHITELIST.TXT file:

- DATA7362.PGN
- DATA7369.PGN
- EP2CPAR.PGN

> This method is the only method available for removing unwanted application files when upgrading from a version prior to RBA v12, but it can be used when upgrading from RBA v12 and higher.

Clean-All-Applications Request

A clean-all-applications request clears all application files before updating a terminal. Execute this request by including a signed CLEANPINPAD.PGN file in the .OGZ, EFT, or LLT download of the RBA release.

> This method may be used for removing unwanted application files when upgrading from RBA v12 or higher only.

Ingenico provides a customized RBA download package (.OGZ or EFT), which contains either:

- A WHITELIST.TXT file for the RBA version and terminal model
- A signed CLEANPINPAD.PGN file

Removing Application Files per Method

Method	62.x / Save File	EFT	Notes
OGZ whitelist	Must download customized RBA package to the terminal via 62.x message to /SWAP	Must download customized RBA package to the terminal via 62.x message to /SWAP	This method is the only method available for removing unwanted application files when upgrading from a version prior to RBA v13 but may also be used when upgrading from RBA v13. This is intended to clear all unwanted application files not listed in the whitelist from the terminal immediately following the download and update to a newer RBA release version. The `WHITELIST.TXT` is embedded as an extra data file in the RB DATA7362.PGN and lists all application files in the release.
EFT whitelist	Not used	Must download customized RBA package to the terminal via EFT download (via TDA)	
OGZCLEANPINPAD.PGN	Must download customized RBA package to the terminal via 62.x message to /SWAP	Must download customized RBA package to the terminal via 62.x message to /SWAP.	This method may be used when upgrading from RBA v. 13.x and higher only.
EFTCLEANPINPAD.PGN	Not used	Must download customized RBA package to the terminal via EFT download (via TDA)	

Additional reboots are required when application files are cleaned/deleted/removed from the terminal. The terminal might continue to display TELIUM MANAGER INVALID immediately after the first reboot for application files that have not yet been cleaned/removed; however, it should NOT display on subsequent reboots after all of the previous version's application files are cleaned/removed from the terminal.

6.2.49 63.x Find File

6.2.49.1 Overview

The 63.x Find File message checks for the existence of a file and returns a flag indicating whether the file was found. If it finds the file, it returns the file length in addition to sending the 'success' flag status.

As an added feature, the 63.x message can be used to retrieve the CRC32 value in addition to the status and file size. To implement this, an optional [FS] character and checksum flag have been added to the 63.x request message. When the [FS] character is inserted in the 63.x request message and the checksum flag is set to '1', the CRC32 value will be appended to the 63.x response message. Consider the following example:

The POS sends a '63.BOOT.HTM[FS]1' request message.

The terminal responds with a '63.02960[FS]6e970159' response message which indicates that the file was found (status = '0') with a file size of '2960' and CRC32 value of '6e970159'.

> RBA will capitalize all letters in file names included in the 63.x request message that it received from the POS.
> As such, all letters in file names should be capitalized, as the 63.x request message will not find any files named with lowercase characters.

6.2.49.2 63.x Find File Message Format

The following tables describe the message formats for the 63.x Find File request and response messages.

63.x Find File Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M63_FIND_FILE Message Identifier – ASCII – “63.”
4	Variable	Alphanum	iConnectEFT Constant = P63_REQ_FILE_NAME File name.
M	1	Constant	ASCII control character – FS (0x1c) – Optional.

Offset	Length	Type	Description
M + 1	1	Constant	iConnectEFT Constant = P63_REQ_REQUEST_CRC Checksum flag (optional). • '1' = Checksum will be CRC32. • All other values are reserved for future use. This is an optional character which is present when the preceding field separator character is present, used to request the checksum value.
M + 2	1	Constant	ASCII control character – ETX
M + 3	1	Binary	LRC check character.

63.x Find File Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M63_FIND_FILE Message Identifier – ASCII – “63.”
4	1	Alphanum	iConnectEFT Constant = P63_RES_RESULT Result • 0 = Found • 1 = Not Found
5	Variable	Alphanum	iConnectEFT Constant = P63_RES_FILE_LENGTH Length of file in bytes

Offset	Length	Type	Description
M	1		ASCII control character – FS (0x1c) – Optional. This field separator character will be present if the optional checksum value is present.
M + 1	Variable	Alphanum	iConnectEFT Constant = P63_RES_CRC Checksum value. This is an optional field. When the [FS] character is inserted and the checksum flag is set to '1'in the 63.x request message , the CRC32 value will be appended to the 63.x response message in this field. The CRC32 field will be returned as an ASCII hex string of variable length with a maximum length of 8 characters. If the P63_RES_RESULT value is '1' indicating that the file was not found, the checksum field will be 1 byte in length with a '0' value.
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

6.2.50 64.x Delete File

The 64.x Delete File message deletes the specified file. The file name can include a directory path to the file to delete. If no path is specified, the file is assumed to be in RBA's secure directory. Use extreme care to not delete required files. If required files are deleted, the RBA will not function until the required files are downloaded into the terminal.

64.x Delete File Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M64_DELETE_FILE Message Identifier – ASCII – “64.”
4	Variable	Alphanum	iConnectEFT Constant = P64_REQ_FILE_NAME File name.
M	1	Constant	ASCII control character – ETX
M + 1	Variable	Binary	LRC check character.

64.x Delete File Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M64_DELETE_FILE Message Identifier – ASCII – “64.”
4	1	Alphanum	iConnectEFT Constant = P64_RES_RESULT Result • 0 = Found. • 1 = Not Found. • 2 = Error Deleting File.
5	1	Constant	ASCII control character – ETX
6	Variable	Binary	LRC check character.

6.2.51 65.x Retrieve File Contents

The POS sends a 65.x message to retrieve the contents of a specific ASCII text file. For each 65.x request, the terminal returns only one 65.x response. To receive the contents of a file that must be separated into multiple blocks, repeat the request until the terminal returns the response containing the last block.

6.2.51.1 Retrieving a Manifest

The following tables describe the contents of the available manifest files.

To retrieve a list of...	Request file...
All files on the terminal	manifest.txt

To retrieve a list of...	Request file...
All files on the terminal with checksum	manifestcrc.txt
All .K3Z and .HTM form files on the terminal	manifestfrm.txt

65.x Retrieve File Contents Request Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – 65.
4	1	Numeric	Message Type: • 0 = Next data block • 1 = Start from file beginning
5	1	Numeric	Data format type: • 0 = Plain text. File data is sent unchanged • 1 = Convert file data into Base 64 format
6	Variable	ASCII	File name, including path and extension
M	1	Constant	ASCII control character – ETX

65.x Retrieve File Contents Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – 65.
4	1	Numeric	Result: • 0 = File found • 1 = File not found • 2 = Error while converting original data to Base 64 format • 3 = Format change error. The block requested is in a different data format than the previously requested block
5	Variable	Numeric	Total number of data blocks
M	1	Constant	ASCII control character – FS
M+1	Variable	Numeric	Block number

Offset	Length	Type	Description
N	1	Constant	ASCII control character – FS
N+1	Variable	Alphanum	CRC value
O	1	Constant	ASCII control character – FS
O+1	1	Numeric	Data format type: • 0 = Plain text. File data is sent unchanged • 1 = Convert file data into Base 64 format
O+2	1	Constant	ASCII control character – FS
O+3	Variable	Alphanum	Data (file content)
P	1	Constant	ASCII control character – ETX

6.2.51.2 65.x Example

The POS sends the following message to check the contents of the APPDAPP.K3Z form file:

```
65.01/HOST/APPDAPP.K3Z
```

The terminal responds with:

```
65.00001[FS]0001[FS]b5098ef9[FS]0[FS]<Form x='0' y='0' width='128' height='64'
template='TEMPLLD.HTM' backgroundcolor='FFFFFF' timeout='0' enterenabled='false'
entertone='0' clearenabled='false' cleartone='0' cancelenabled='false' canceltone='0'
tonetype='0' f1enabled='false' f2enabled='false' f3enabled='false' f4enabled='false' />

<LineDisplay id='linedisplay1' width='173' height='62' rows='4' cols='21' x='0' y='0'
textcolor='000000' fontsize='8px' fontweight='normal' fontfamily='userfont2'
backgroundcolor='FFFFFF' background='true' bordersize='0' bordercolor='000000'
overridewidth='true' overrideheight='true' scrolltype='auto'
scrollbarvisibility='hidden' />

<Label id='PROMPTLINE1' textsource='custom' text='&lt;?ivPROMPTLINE1?&gt;' x='0' y='38'
width='126' height='10' border='true' bordercolor='000000' textcolor='000000'
fontsize='9px' fontweight='normal' fontfamily='userfont1' align='center'
background='false' backgroundcolor='FFFFFF' />
```

This example response shows:

- 0 = The file is found
- 0001 = There is only 1 block
- 0001 = This is the first block
- b5098ef9 is the CRC value.
- 0 = The data block is in plain text
- The remaining text after the last field separator is the form data

6.2.52 70.x Update Form Element Message

The 70.x Update Form Element message flows from the POS to the terminal and is used to update text fields or prompts in the terminal forms displayed. The text field in the 70.x message has a limit of 500 characters. After expanding newline characters, the new text field limit is 625 characters.

When the terminal receives a 70.x message, it updates the form element with the specified ID in the currently displayed form with the new field data provided in the message. As a general rule, the ID used in the 70.x message to select the prompt update must be the variable which is referenced in the "text= " parameter of the label in the .K3Z form.

As an example, a form contains the following label:

<Label id= '**VAR3**' textsource= 'custom' text= '&lt;?ivVAR3?&gt;' x= '10' y= '5' width= '780' height= '280' border= 'false' bordercolor= '000000'

textcolor= '000000' fontsize= '20px' fontweight= 'medium' fontfamily= 'sans-serif' align= 'left' background= 'false' backgroundcolor= 'FFFFFF' />

For the above example label, the message to set the prompt in the form would be "70.T**VAR3**,A Prompt".

6.2.52.1 Setting Prompt Text

The syntax for setting prompt text is as follows:

Tid,text

where

- id = the label ID from the K3Z form.
- text = the new prompt text.

Example: Set the prompt for label id "PROMPTLINE1" to "A new label":

TPROMPTLINE1,A new label

6.2.52.2 Setting Button Text

The syntax for setting button text is as follows:

Tvar,text

where

- var = the button's buttontext variable from the K3Z form.
- text = the new button text.

Example: Set the button with buttontext variable "btn1" to "text":

Tbtn1,text

> Button text in 70.x messages may be specified as an index in the `prompt.xml` file. For example, specifying "106" will call button index 106, [DECLINE]. See 70.x Update Form Element Message Format table below, index 5.

6.2.52.3 Changing Button Visibility

The syntax for changing a button visibility is as follows:

Bid,visibility

where

- id = the button ID from the K3Z form.
- visibility = "S" for show or "H" for hide.

Example: Hide button with id "btn1":

Bbtn1,H

> If "S" or "H" are needed for the button text, then the following syntax should be used in order to avoid confusion:
>
> Bid, S
>
> Bid, H
>
> Note the extra space following "Bid".

The following table which describes the 70.x message format.

70.x Update Form Element Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M70_UPDATE_FROM_ELEMENT Message Identifier – ASCII – “70.”
4	1	Alphanum	iConnectEFT Constant = P70_REQ_ELEMENT_TYPE_TO_CHANGE Indicates element type to change: • T = text field or prompt. • B = Button.
5	Variable	Alphanum	iConnectEFT Constant = P70_REQ_ID_OF_FIELD_TO_CHANGE ID of field to update. If the element type to change is "B" (button), then a button with an ID of "P" <hex50> will be visible while a button with an ID of "A" <hex41> or "$" <hex24> will become hidden.
M	1	Constant	ASCII comma (0x2C)

Offset	Length	Type	Description
M + 1	Variable	Alphanum	iConnectEFT Constant = P70_REQ_NEW_FIELD_DATA New field data: • Alphanumeric = text field will be updated to the alphanumeric string. • Numeric = text field will be updated to the corresponding prompt index. To force a text field to display a number, include a space as the first character in this field. Text fields have a limit of 500 characters. After expanding newline characters, the text field limit is 625 characters. If element type = B: • SHOW = make button visible. • HIDE = hide the button.
N	1	Constant	ASCII control character – FS Optional. Only required if another element follows.
N + 1	Variable	Varies	Optional data for another element.
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character.

6.2.53 72.x Audio Play Request

The 72.x message flows from the POS to the terminal. It can be sent on any screen of the terminal (Offline, Online, and Transaction Flow screens are all valid). The message is used to prompt the terminal to play an .OGG audio file or files specified in the message.

Audio files must be loaded onto the terminal in order to be played. Volume is controlled by 0007_0052.

The following table outlines the 72.x message format.

72.x Audio Play Request message format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant =M72_AUDIO_PLAY_REQUEST Message Identifier – ASCII – “72.”

Offset	Length	Type	Description
4	1	Alphanum	Message subtype: • 0 = Stop playback. Playback will stop once the current audio file finishes playing. • 1 = Play list of audio files in the order listed.
5	Variable	Alphanum	Name of audio file (15 character limit including extension). Must be given in upper case. This field only included if the previous field is "1".
M	1	Constant	ASCII control character – FS (This field is optional.) Only used with M + 1.
M+1	Variable	Alphanum	Name of next audio file (15 character limit including extension). Must be given in upper case. (This field is optional.) Only used with M.
Any number of M and M+1 Pairs may be used, up to a total of 20 audio files per 72.x request.			
N	1	Constant	ASCII control character – ETX.
N+1	1	Binary	LRC check character.

Example play request: '72.1ONE.OGG[FS]HUNDRED.OGG'

Example cancel request: '72.0'

All audio files are custom, and should be placed in the HOST directory on the device. The RBA integration kit does not include any audio files. Only the OGG file format is supported.

72.x Audio Play Response message format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant =M72_AUDIO_PLAY_RESPOSE Message Identifier – ASCII – “72.”

Offset	Length	Type	Description
4	1	Alphanum	Status. • 0 = Audio play complete. • 1 = One or more files not present on the terminal. • 2 = Media file list is empty. • 3 = Media play is interrupted. • 4 = Media play is in progress. Sent if a 72.x request is sent while another 72.x request's audio is already playing. • 5 = Invalid request. For example, in response to a 72.0 when no playback is in progress.
5	1	Constant	ASCII control character – ETX.
6	1	Binary	LRC check character.

6.2.54 80.x MAC Calculation Message Format

80.x MAC Calculation Request Message Format for Single Length Key

Offset	Length	Format	Description
0	1	Constant	ASCII control character, STX
1	3	Constant	Message identifier: • 80
4	1	Alphanum	Indicates the MAC master key index. The value of this field is not supported. Use the key index as the parameter 0016_0001 Encryption Key Index setting.
5	1	Alphanum	Session key length flag. • 1 = Single key length
6	16	Constant	Encrypted MAC session key (encrypted with MAC master key)
22	Variable	Alphanum	Base64-encoded MAC data (data length less than 4072 bytes)
M	1	Constant	ASCII control character, ETX
M+1	1	Binary	LRC check character

80.x MAC Calculation Request Message Format for Double Length Key

Offset	Length	Format	Description
0	1	Constant	ASCII control character, STX
1	3	Constant	Message identifier: • 80
4	1	Alphanum	The MAC master key index The value of this field is not supported. Use the key index as the parameter 0016_0001 Encryption Key Index setting.
5	1	Alphanum	Session key length flag. • 2 = Double key length
6	32	Constant	Encrypted MAC session key (encrypted with MAC master key)
38	8	Alphanum	Key Check Value (KCV) to be verified after key is loaded
46	Variable	Alphanum	Base64-encoded MAC data (data length fewer than 4048 bytes)
M	1	Constant	ASCII control character, ETX
M+1	1	Binary	LRC check character

80.x MAC Calculation Response Message Format

Offset	Length	Format	Description
0	1	Constant	ASCII control character, STX
1	3	Constant	Message identifier: • 80
4	1	Alphanum	MAC calculation result. • 0 = Success • 1 = Failure • 9 = Security application error
5	8	Alphanum	Calculated MAC value
13	1	Constant	ASCII control character, ETX
14	1	Binary	LRC check character

6.2.55 81.x MAC Verification Message Format

81.x MAC Verification Request Message Format for Single Length Key

Offset	Length	Format	Description
0	1	Constant	ASCII control character, 'STX'.
1	3	Constant	Message identifier: • 80.
4	1	Alphanum	Indicates the MAC master key index. The value of this field is not currently used. The RBA uses the key index as the parameter '0016_0001' Encryption Key Index setting.
5	1	Alphanum	Session key length flag: • 1 = Single length key.
6	16	Constant	Encrypted MAC session key (encrypted with MAC master key).
22	Variable	Alphanum	Base64 encoded MAC data (data length less than 4072 bytes).
M	1	Constant	ASCII control character, 'ETX'.
M+1	1	Binary	LRC check character.

81.x MAC Verification Request Message Format for Double Length Key

Offset	Length	Format	Description
0	1	Constant	ASCII control character, 'STX'.
1	3	Constant	Message identifier. • 81.
4	1	Alphanum	Indicates the MAC master key index. The value of this field is not currently used. The RBA uses the key index as the parameter '0016_0001' Encryption Key Index setting.

Offset	Length	Format	Description
5	1	Alphanum	Session key length flag: • 2 = Double key length.
6	32	Constant	Encrypted MAC session key (encrypted with MAC master key).
38	8	Alphanum	Key check value (KCV), to be verified after key is loaded.
46	Variable	Alphanum	Base64 encoded MAC data (data length less than 4048 bytes).
M	1	Constant	ASCII control character, 'ETX'.
M+1	1	Binary	LRC check character.

81.x MAC Verification Response Message Format

Offset	Length	Format	Description
0	1	Constant	ASCII control character, 'STX'.
1	3	Constant	Message identifier: • 81.
4	1	Alphanum	MAC calculation result: • 0 = Success. • 1 = Failure. • 9 = Security application error.
5	8	Alphanum	Calculated MAC value.
13	1	Constant	ASCII control character, 'ETX'.
14	1	Binary	LRC check character.

6.2.56 82.x On-Guard and KME Session Key Injection Message

The 82.x On-Guard and KME Session Key message allows the application to inject the KME session key into the terminal. The KME master key must first be loaded into the terminal by the Key Injection Application (KIA). This master key must be handled by the terminal and POS. The target key slot used by the injection command is specified in the configuration file, which was loaded to enable E2EE encryption.

82.x On-Guard and KME Session Key Request Message Format - Single Length

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M82_KME_SESSION_KEY_INJECTION Message identifier – ASCII – 82
4	16	Alphanumeric	iConnectEFT Constant = P82_REQ_SESSION_KEY Session key
20	1	Constant	ASCII control character – ETX
21	1	Binary	LRC check character

82.x On-Guard and KME Session Key Request Message Format - Double Length

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M82_KME_SESSION_KEY_INJECTION Message identifier – ASCII – 82
4	32	Alphanumeric	iConnectEFT Constant = P82_REQ_SESSION_KEY Session key
36	8	Alphanumeric	iConnectEFT Constant = P82_REQ_KCV Key Check Value (KCV)
44	1	Constant	ASCII control character – ETX
45	1	Binary	LRC check character

82.x On-Guard and KME Session Key Injection Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M82_KME_SESSION_KEY_INJECTION Message identifier – ASCII – 82
4	1	Decimal	iConnectEFT Constant = P82_RES_STATUS Status • 0 = Success • 1 = Failed
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character

This message is used with On-Guard or KME encryption enabled only.

6.2.57 83.x On-Guard and KME Enable Message

The 83.x On-Guard and KME Enable Message command activates the E2EE feature and updates the `E2ECFG` and `security.dat` files if all of the following prerequisites are met:

- The selected cryptogram key is place
- For On-Guard:
    - 0091_0001 = 2
    - Settings enabled within `E2ECFG`

This message is used with On-Guard or KME encryption enabled only. It is rejected if another encryption type is enabled.

No reboot is required after sending the 83.x message to enable the specified encryption. The encryption type is enabled, even after a reboot.

This message enables the terminal to be preloaded with the required software and keys. If a valid message is received while the terminal is already enabled, the terminal parameters are switched if all enabling checks for the new encryption parameters to be active are met. Switching the encryption type does not disable E2EE. When the POS and network are ready, the POS can send a single command to enable E2EE. The only means of disabling E2EE encryption is to erase the terminal and reload all components.

83.x On-Guard and KME Enable Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M83_E2EE_ENABLE Message identifier – ASCII – 83
4	1	Alphanum	iConnectEFT Constant = P83_REQ_E2EE_MODE E2EE mode: • 1 = KME mode is enabled • 2 = On-Guard mode is enabled
5	1	Alphanum	iConnectEFT Constant = P83_REQ_OUTPUT_FORMAT Output format: • A = Type A Base 24 returns Track 2 only and supports Base24 framing • B = Type B IngeCrypt returns Track 2 only (with no framing) and the KSN for the cryptogram
6	1	Alphanum	iConnectEFT Constant = P83_REQ_KEY_TYPE Key type: • M = Master Session key • D = E2EE DUKP
7	1	Alphanum	iConnectEFT Constant = P83_REQ_KEY_NUMBER Key number: • 0 - 9 for Telium key pattern 1 • 0 - O for Telium key pattern 2 • 0 - 5 for Telium key pattern 4 Key number value must be 2 to specify KME key
8	1	Alphanum	iConnectEFT Constant = P83_REQ_LOCAL_STORAGE_KEY Optional local storage key: • 0 - 9 for key number of optional TDES local storage data encryption key The format of the LS data block is always that of the manual-entry definition.
9	1	Constant	ASCII control character – ETX

Offset	Length	Type	Description
10	1	Binary	LRC check character

83.x On-Guard and KME Enable Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M83_E2EE_ENABLE Message identifier – ASCII – 83
4	1	Decimal	iConnectEFT Constant = P83_RES_STATUS Status code: • 0 = Success; On-Guard or KME is enabled • 1 = Failure; invalid command • 2 = Failure; cannot switch to the requested mode
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character

6.2.58 85.x On-Guard and KME Non-Payment Card Message

The 85.x On-Guard and KME Non-Payment Card message is sent from the terminal to the POS. When On-Guard or KME encryption is enabled, this message is sent in place of the standard 18.x Non-Payment Card Message.

1. When the cardholder swipes a card and selects a payment method, the method is checked against the local configuration in the `cards.dat` file.
2. If the Card Type option for the selected payment is set to 1 (indicating a non-payment card type), the terminal sends the 85.x message.
3. The card information is encrypted before being incorporated into the message.

> This message is used with On-Guard or KME encryption enabled only.

85.x On-Guard and KME Non-Payment Card Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M85_E2EE_INFO Message identifier – ASCII – 85
4	Variable	Decimal	iConnectEFT Constant = P85_RES_ENCR_CARD_DATA Encrypted card data Refer to On-Guard Card Data Encryption Rules
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character

6.2.59 86.x On-Guard and KME BIN Lookup (PIN Encouragement) Message

The 86.x On-Guard and KME BIN Lookup message is sent to the POS to allow for external BIN range lookup, which is used to pre-select the payment type for a cardholder. This message is similar to the 19.x message, except card data is encrypted. The POS returns the payment type of the swiped card only if it is a non-chip card. For a chip card, the character, '-', is returned to indicate that a chip card is detected.

This message is used with On-Guard or KME encryption enabled only.

The 86.x message supports Canadian applications using On-Guard or KME encryption.

Note that the POS performs the BIN range checking and returns the card type in the response message.

86.x On-Guard and KME BIN Lookup Request Message Format (as received from RBA)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M86_E2EE_BIN_LOOKUP Message identifier – ASCII – 86

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P86_REQ_ACCOUNT_SOURCE Indicates where account data was derived: • H = Electronic Track 1 • D = Electronic Track 2 (default track to use) • h = Contactless Track 1 • d = Contactless Track 2 • T = Manual Track 2 The contactless indicators can be configured in the `config.dat` file.
5	1	Alphanum	iConnectEFT Constant = P86_REQ_TRACK_1_INDICATOR Track 1 good read indicator: • 0 = Bad read • 1 = Good read
6	1	Alphanum	iConnectEFT Constant = P86_REQ_TRACK_2_INDICATOR Track 2 good read indicator: • 0 = Bad read • 1 = Good read
7	1	Alphanum	Reserved (use 0)
8	4	Alphanum	iConnectEFT Constant = P86_REQ_REQUEST_COUNTER Request counter
12	6	Decimal	iConnectEFT Constant = P86_REQ_PAN_FIRST_6DIGIT PAN, first six digits
18	1	Constant	ASCII control character – FS
19	4	Decimal	iConnectEFT Constant = P86_REQ_EXP_DATE Expiry Date (YYMM)

Offset	Length	Type	Description
23	1	Constant	ASCII control character – FS
24	3	Decimal	iConnectEFT Constant = P86_REQ_SERVICE_CODE Service code
27	1	Constant	ASCII control character – FS
28	1	Decimal	iConnectEFT Constant = P86_REQ_PAN_MOD_10_FLAG PAN Mod-10 check flag
29	1	Constant	ASCII control character – ETX
30	1	Binary	LRC check character

86.x On-Guard and KME BIN Lookup Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M86_E2EE_BIN_LOOKUP Message identifier – ASCII – 86

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P86_RES_PAYMENT_TYPE Payment type selected: - - = Prompt user to insert card - 0 = Invalid card - 9 = Unknown card - A = Card type 1: default is debit - B = Card type 2: default is credit - C = Card type 3: default is EBT cash - D = Card type 4: default is EBT food stamp - E = Card type 5: default is store credit - ... - O = Card type 15 - P = Card type 16 Undefined responses are treated as Unknown
5	4	Alphanum	iConnectEFT Constant = P86_RES_RESPONSE_COUNTER Response counter (copied from request message)
9	1	Constant	ASCII control character – ETX
10	1	Binary	LRC check character

6.2.60 87.x On-Guard and KME Card Read Data

The 87.x On-Guard and KME Card Read Data message is used to send On-Guard and KME encrypted card data to the POS upon request. It functions exactly like the 23.x Card Read Request (On-Demand) message, and its request and response formats are the same as the 23.x message

> This message is only used with On-Guard or KME encryption enabled. Also note that if a card whitelisted by `E2EBIN` is used (**not** `secbin.dat`), the data will be formatted in the clear rather than encrypted. See Type B Formatting for details.

87.x On-Guard and KME Card Read Data Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M87_E2EE_CARD_READ Message identifier – ASCII – "87."
4	Variable	Alphanum	iConnectEFT Constant = P87_REQ_PROMPT_INDEX Prompt index number. Can be literal text up to 230 characters (such as 'Please swipe'), an index number from Prompt.xml, or nothing when used with an optional form_file_name. Value cannot be "0".
M	1	Constant	ASCII control character – FS. This field is optional, and is only used with the Form Name or Number. Only used with M + 1.
M + 1	Variable	Alphanum	iConnectEFT Constant = P87_REQ_FORM_NAME Form Name or Number. This field is optional. When used, it is preceded by the above 'FS' ASCII control character.
N	1	Constant	ASCII control character – FS. This field is optional, and is only used with the Form Name or Number. Only used with N + 1.
N + 1	1	Alphanum	iConnect EFT Constant = P87_REQ_OPTIONS • 1 = RBA will send 16.x response with status '6' if payment data is available and pending a 87.x response.
N + 2	1	Constant	ASCII control character – FS (This field is optional.)

Offset	Length	Type	Description
N + 3	Variable	Alphanum	iConnect EFT Constant = P87_REQ_ENABLE_DEVICES Enable Devices field (this field is optional). Letters in the field indicate which readers to enable: • M = Enable MSR • C = Enable contactless. Contactless will not be enabled if 0008_0001 = 0 • S = Enable SCR. Smart card reader will not be enabled for EMV if 0019_0001 = 0 or for WIC if 0020_0001 = 0 • H = Ignored if sent with the others. If sent alone, force manual entry for this transaction (0007_0029 = 0 is treated as 0007_0029 = 1) and ignore specified prompt and form. Regardless of readers enabled, 0007_0029 controls whether <ENTER CARD> is displayed on the standard swipe forms. The order of the letters does not matter, only whether or not they are included in the string. The reader(s) specified must be enabled in configuration for this message to activate it/them. If any specified readers are not enabled by configuration, '23.6' is returned to indicate such. If no readers are specified, all readers enabled by configuration are activated. To avoid ambiguity, the Form field must be present (even with empty value) in order for the MCS field to be sent.
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character.

87.x On-Guard and KME Card Read Data Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	iConnectEFT Constant = M87_E2EE_CARD_READ Message identifier – ASCII – "87."

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P87_RES_EXIT_TYPE Exit type: • 0 = Good Read • 1 = Bad Read • 2 = Cancelled • 3 = Button Pressed • 4 = Cless Card Floor Limit Exceeded • 5 = Max Cless Floor Limit Exceeded • 6 = Invalid Prompt • 7 = Encryption Failed • 8 = Bad key card • 9 = Bad format of 23. message and/or on-demand 23. message not allowed at this time since on-demand already running • R = At least one specified reader is disabled
5	1	Alphanum	iConnectEFT Constant = P87_RES_CARD_SOURCE Source of card read: • ◦ C = Contactless Reader ◦ E = EMV Contactless ◦ H = Manual entry ◦ M = MSR ◦ Q = Fast quick chip ◦ S = One of the following: ▪ SLE5542 memory card ▪ EMV card ▪ WIC card ◦ A = Account message entry ◦ c = Coupon or key card ◦ m = Mobile ◦ ? = Unknown/invalid card type **Only included in Response if 0013_0014 = 1.**

Offset	Length	Type	Description
6	Variable	Decimal	ConnectEFT Constant = P87_RES_CARD_DATA Encrypted/formatted card data. The format of this field is described in the On-Guard Card Data Encryption Rules section. If the exit type is "Button pressed" then this will be the single byte ID of the key.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

87.x On-Guard and KME Card Read Request using an MSR or Cless card

Offset	Length	Type	Description
4	1	Alphanum	Exit Type
5	1	Alpha	Source of card read
6	Variable	Decimal	Encrypted card data
M	1	Constant	FS (Only if next field “Token value” is available)
M+1	Variable	Alphanum	Token value generated by the terminal (Only if available)

87.x On-Guard and KME Card Read Response using an MSR or Cless card

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “87.”

Offset	Length	Type	Description
4	1	Alphanum	Exit type: • 0 = Good Read • 1 = Bad Read • 2 = Cancelled • 3 = Button Pressed • 4 = Cless Card Floor Limit Exceeded • 5 = Max Cless Floor Limit Exceeded • 6 = Invalid Prompt • 7 = Encryption Failed • 8 = Bad key card • 9 = Bad format of 23. message and/or on-demand 23. message not allowed at this time since on-demand already running • R = At least one specified reader is disabled
5	1	Alpha	Source of card read: • ◦ C = Contactless reader. ◦ E = EMV contactless. ◦ H = Manual entry. ◦ M = MSR. ◦ Q = Fast quick chip ◦ S = One of the following: ▪ SLE5542 memory card ▪ EMV card ▪ WIC card ◦ A = Account message entry. ◦ c = Coupon or key card. ◦ m = Mobile. ◦ ? = Unknown/invalid card type. **Only included in Response if 0013_0014 = 1.**
M	Variable	Decimal	Encrypted card data.
N	1	Constant	ASCII control character – FS (Only if next field “Token value” is available)
N+1	Variable	Alphanum	Token value generated by the terminal (Only if available)
P	1	Constant	ASCII control character – ETX

Offset	Length	Type	Description
P+1	1	Binary	LRC check character

6.2.61 88.x On-Guard and KME Translate Encrypted Card Data Message

This message takes transaction card data encrypted under the local storage key (AES or E2EE) and returns card data encrypted under the appropriate E2EE key. It is typically used in a store-and-forward scenario. The E2EE key location is indicated in the configuration parameter E2EE key slot number.

88.x On-Guard and KME Translate Encrypted Card Data Request Message Format - Single Length Key

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Decimal	iConnectEFT Constant = M88_E2EE_TRANSLATE_ENCR_CARD_DATA Message identifier – ASCII – 88
4	2	Alphanumeric	iConnectEFT Constant = P88_REQ_LENGTH_OF_ENCR_DATA Length of encrypted data field
6	Variable	Alphanumeric	iConnectEFT Constant = P88_REQ_ENCR_DATA AES/TDES/KME-encrypted data field
N	16	Alphanumeric	iConnectEFT Constant = P88_REQ_SINGLE_KME_KEY Optional single-length KME key to re-encrypt card data
N+16	6	Alphanumeric	iConnectEFT Constant = P88_REQ_KCV Key Check Value (KCV)
N+22	1	Constant	ASCII control character – ETX
N+23	1	Binary	LRC check character

88.x On-Guard and KME Translate Encrypted Card Data Request Message Format - Double Length Key

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M88_E2EE_TRANSLATE_ENCR_CARD_DATA Message identifier – ASCII – 88

Offset	Length	Type	Description
4	2	Decimal	iConnectEFT Constant = P88_REQ_LENGTH_OF_ENCR_DATA Length of encrypted data field
6	Variable	Alphanumeric	iConnectEFT Constant = P88_REQ_ENCR_DATA Encrypted data field
N	32	Alphanumeric	iConnectEFT Constant = P88_REQ_DOUBLE_KME_KEY Optional double KME key to re-encrypt card data
N+32	6	Alphanumeric	iConnectEFT Constant = P88_REQ_KCV Key Check Value (KCV)
N+38	1	Constant	ASCII control character – ETX
N+39	1	Binary	LRC check character

88.x On-Guard and KME Translate Encrypted Card Data Response Message Format - KME Mode

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M88_E2EE_TRANSLATE_ENCR_CARD_DATA Message identifier – ASCII – 88
4	1	Decimal	iConnectEFT Constant = P88_RES_STATUS Status. • 0 = Success • 1 = Invalid command • 2 = Encryption not loaded • 8 = Track data decryption error • 9 = Invalid format type for optional translation • A = Key load failure • B = KCV error
5	6	Decimal	iConnectEFT Constant = P88_RES_PAN_FIRST6DIG Clear value of the first six digits of the PAN

Offset	Length	Type	Description
11	4	Decimal	iConnectEFT Constant = P88_RES_PAN_LASTT4DIG Clear value of the last four digits of the PAN
15	2	Decimal	iConnectEFT Constant = P88_RES_PAN_LENGTH PAN length
17	1	Decimal	iConnectEFT Constant = P88_RES_PAN_MOD PAN Mode 10 check flag. • 0 = PAN Mod 10 check failed • 1 = PAN Mod 10 check passed
18	4	Decimal	iConnectEFT Constant = P88_RES_PAN_EXP_DATE Clear value of expiry date
22	3	Decimal	iConnectEFT Constant = P88_RES_PAN_SERVICE_CODE Clear value of service code
25	1	Decimal	iConnectEFT Constant = P88_RES_ENCR_FLAG Card encrypted data flag • 0 = Clear ASCII data • 1 = Encrypted data
26	1	Alphanumeric	iConnectEFT Constant = P88_RES_ENCR_FORMAT Encrypted format type • A = KME format
27	2	Decimal	iConnectEFT Constant = P88_RES_ENCR_DATA_LENGTH Encrypted data length
29	Variable	Alphanumeric	iConnectEFT Constant = P88_RES_ENCR_DATA KME Encrypted data field
M	1	Constant	ASCII control character – ETX
M+1	1	Binary	LRC check character

88.x On-Guard and KME Translate Encrypted Card Data Response Message Format - On-Guard Mode

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M88_E2EE_TRANSLATE_ENCR_CARD_DATA Message identifier – ASCII – 88
4	1	Decimal	iConnectEFT Constant = P88_RES_STATUS Status. • 0 = Success • 1 = Invalid command • 2 = Encryption not loaded • 8 = Track data decryption error
5	6	Decimal	iConnectEFT Constant = P88_RES_PAN_FIRST6DIG Clear value of first 6 digits of the PAN
11	4	Decimal	iConnectEFT Constant = P88_RES_PAN_LASTT4DIG Clear value of last four digits of the PAN
15	2	Decimal	iConnectEFT Constant = P88_RES_PAN_LENGTH PAN length
17	1	Decimal	iConnectEFT Constant = P88_RES_PAN_MOD PAN Mode 10 check flag • 0 = PAN Mod 10 check failed • 0 = PAN Mod 10 check passed
18	4	Decimal	iConnectEFT Constant = P88_RES_PAN_EXP_DATE Clear value of expiry date
22	3	Decimal	iConnectEFT Constant = P88_RES_PAN_SERVICE_CODE Clear value of service code
25	1	Decimal	iConnectEFT Constant = P88_RES_ENCR_FLAG Card encrypted data flag • 0 = Clear ASCII data • 1 = Encrypted data

Offset	Length	Type	Description
26	1	Alphanumeric	iConnectEFT Constant = P88_RES_ENCR_FORMAT Encrypted format type • B = DUKPT format
27	20	Alphanumeric	iConnectEFT Constant = P88_RES_ENCR_DUKPT_KSN DUKPT Key Serial Number (KSN) + reserved value of 0114 KSN = 20 chars; Reserved value = 4 chars
51	2	Decimal	iConnectEFT Constant = P88_RES_ENCR_DATA_LENGTH Length of Ingecrypt-encrypted data
53	Variable	Alphanumeric	iConnectEFT Constant = P88_RES_ENCR_DATA Ingecrypt-encrypted data field
M	1	Constant	ASCII control character – ETX
M+1	1	Binary	LRC check character

This message is used with On-Guard or KME encryption enabled only.

6.2.62 89.x On-Guard and KME Register BIN Record Message

This message allows the BIN table to be updated by the application. To change a BIN table record, the new record must be MAC'ed using the CEFMK key.

89.x On-Guard and KME Register BIN Record Request Message Format - Change BIN

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M89_E2EE_BIN_RECORD Message identifier – ASCII – 89.
4	1	Decimal	iConnectEFT Constant = P89_REQ_COMMAND Request flag • 0 = Change index

Offset	Length	Type	Description
5	2	Decimal	iConnectEFT Constant = P89_REQ_BIN_INDEX BIN record index • 00 - 3F
7	17	Decimal	iConnectEFT Constant = P89_REQ_BIN_RECORD BIN record
24	8	Hexadecimal	iConnectEFT Constant = P89_REQ_BIN_MAC Record MAC value • MAC'ed using CEFMK key
32	1	Constant	ASCII control character – ETX
33	1	Binary	LRC check character.

89.x On-Guard and KME Register BIN Record Request Message Format - Delete BIN Index

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M89_E2EE_BIN_RECORD Message identifier – ASCII – 89.
4	1	Decimal	iConnectEFT Constant = P89_REQ_COMMAND Request flag • 1 = Delete index
5	2	Decimal	iConnectEFT Constant = P89_REQ_BIN_INDEX BIN record index • 00 - 3F
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character

89.x On-Guard and KME Register BIN Record Request Message Format - Delete all BIN Indexes

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M89_E2EE_BIN_RECORD Message identifier – ASCII – 89.
4	1	Decimal	iConnectEFT Constant = P89_REQ_COMMAND Request flag • 2 = Delete all
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character

89.x On-Guard and KME Register BIN Record Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M89_E2EE_BIN_RECORD Message identifier – ASCII – 89.
4	1	Decimal	iConnectEFT Constant = P89_RES_STATUS Request flag • 0 = Success • 1 = Failed • 2 = Invalid MAC
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character

This message is only used with On-Guard or KME encryption enabled.

6.2.63 90.x P2PE Data Message

The 90.x P2PE Data Message sends and receives cardholder P2PE data. It also receives encryption public keys from the POS for dynamic RSA OAEP public key updates and to select or delete RSA-OAEP public keys. Refer to the following sections for more detailed information on how to use the 90.x message:

Voltage Encryption

- 90.0 - Generating a Key On Demand

- 90.1 - Getting Encryption Transmission Block (ETB)

RSA-OAEP Encryption

- 90.5 - Dynamically Updating RSA-OAEP Public Keys
- 90.6 - Deleting Public Keys from the Terminal
- 90.7 - Selecting Public Keys from the Terminal

AES Encryption

- 90.8 - See Using 90.x P2PE Data Messages with AES Encryption

6.2.63.1 Voltage Encryption – Generating a Key On Demand

This function generates a new key and returns the encryption transmission block (ETB) for the key. Because generating the new ETB takes about 10 seconds, the response is delayed. If the public data has not been sent to the terminal, no ETB is generated.

90.0 Voltage Generate Key Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_REQ_FUNCTION Function – ASCII – 0
5	1	Constant	ASCII Control Character – ETX
6	1	Binary	LRC check character.

90.0 Voltage Generate Key Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_RES_FUNCTION Function – ASCII – 0

Offset	Length	Type	Description
5	1	Decimal	iConnectEFT Constant = P90_RES_STATUS Status: • 0 = Successful • 1 = Invalid function
6	Variable	Alphanum	iConnectEFT Constant = P90_RES_ETB Encryption Transmission Block (ETB) Base64 encoded
M	1	Constant	ASCII Control Character – ETX
M+1	1	Binary	LRC check character

6.2.63.2 Voltage – Getting Encryption Transmission Block (ETB)

This function returns the encryption transmission block (ETB) for the current Voltage encryption key.

90.1 Voltage Get Encryption Transmission Block Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_REQ_FUNCTION Function – ASCII – 1
5	1	Constant	ASCII Control Character – ETX
6	1	Binary	LRC check character

90.1 Voltage Get Encryption Transmission Block Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P90_RES_FUNCTION Function – ASCII – 1
5	1	Decimal	iConnectEFT Constant = P90_RES_STATUS Status. • Successful • Invalid function
6	Variable	Alphanum	iConnectEFT Constant = P90_RES_ETB Encryption Transmission Block (ETB) Base64 encoded
M	1	Constant	ASCII Control Character – ETX
M + 1	1	Binary	LRC check character.

6.2.63.3 RSA OAEP Encryption - Dynamically Updating RSA-OAEP Public Keys

The 90.5 message is used for dynamically updating RSA-OAEP encryption public keys from the POS. The request message sent from the POS contains the key name, key file data, and signature data. The terminal uses the signature and the signature verification public key to validate the new encryption public key. Security parameter 0091_0032 (Public Key for Signature Verification) must be set prior to sending the request message. Once validated, the encryption public key data will be stored in a file in the `/F_SECURITY_APP/RSAKEYS` directory with the file name provided in the Key Name field of the request message. A .PEM extension will be appended to the file. If the signature is not validated then the encryption public key data will be discarded.

The response message from the terminal returns the success status of the key update request. Upon receiving a valid request, the terminal will check for a key with an identical name and return an error response if found. If the signature verification key could not be loaded or if parameter 0091_0032 is not set, an invalid response code will be returned. If no key with the same name exists, then the key data will be verified with the signature in memory prior to storing the key data to its file. This ensures that unnecessary writes to the flash drive are prevented. The following tables provide the formats for the 90.5 Receive Encryption Public Key request and response messages.

90.5 RSA-OAEP Public Key Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_RES_FUNCTION Function – ASCII – 5 – New RSA Public Key function.

Offset	Length	Type	Description
5	4		Reserved. This field may be used for more/last flag and packet number if required in future applications.
9	Variable	Alphanum	Key Name. • Maximum 11 characters.
M	1	Constant	ASCII control character – FS (0x1C)
M + 1	Variable	Alphanum	RSA Key File Data. • PEM format.
N	1	Constant	ASCII control character – FS (0x1C)
N + 1	Variable	Alphanum	Signature Data. • Base64 encoded format.
O	1	Constant	ASCII Control Character – ETX
O + 1	1	Binary	LRC check character.

90.5 RSA-OAEP Public Key Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_RES_FUNCTION Function – ASCII – 5 – New RSA Public Key function.
5	1	Alphanum	Result Code. • 0 = Success • 1 = Invalid request • 2 = Key with same name already exists • 3 = Signature check failed • 4 = Failed to store key (file system failure)
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC check character.

6.2.63.4 RSA OAEP Encryption - Deleting Public Keys from the Terminal

The 90.6 message is used by the POS to delete RSA-OAEP encryption public keys from the terminal. Upon receiving this message, the terminal deletes the public key. The file to be deleted is located in the `/F_SECURITY_APP/RSAKEYS` directory and contains a .PEM file extension. If the signature verification key could not be loaded or if parameter 0091_0032 is not set, an invalid response code is returned. If no key matching the Key Name is found, then an error message is returned. In either case, the encryption public key currently in use must not be deleted. An error response is returned if an attempt is made to delete this key. The following tables describe the 90.6 Delete RSA-OAEP Public Key request and response messages.

90.6 Delete RSA-OAEP Public Key Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_RES_FUNCTION Function – ASCII – 6
5	Variable	Alphanum	Key Name. • Maximum 11 characters.
M	1	Constant	ASCII Control Character – ETX
M + 1	1	Binary	LRC check character.

90.6 Delete RSA-OAEP Public Key Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_RES_FUNCTION Function – ASCII – 6

Offset	Length	Type	Description
5	Variable	Alphanum	Result Code. • 0 = Success • 1 = Invalid request • 2 = No key was found • 3 = Key is currently in use • 4 = Failed to delete key (file system failure)
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC check character.

6.2.63.5 RSA OAEP Encryption - Selecting Public Keys from the Terminal

The 90.7 message is used by the POS to select RSA-OAEP encryption public keys from the terminal. Upon receiving this message, the terminal locates and loads the public key stored under the file name provided in this request message. The file to be loaded is located in the `/F_SECURITY_APP/RSAKEYS` directory and contains a .PEM file extension. Upon successfully loading this key, RSA-OAEP encryption is fully enabled using the selected key. Security parameter 0091_0033 is updated to preserve this key selection following reboots. If the selected key is not successfully loaded, then the key currently in use continues to be used for card data encryption. If there is no encryption public key loaded, then when this request message is received, the terminal enters the offline state, and returns an error message when encryption is attempted.

The following tables describe the 90.6 Delete RSA-OAEP Public Key request and response messages.

90.7 Select RSA-OAEP Public Key Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_RES_FUNCTION Function – ASCII – 7 – New RSA Public Key function
5	Variable	Alphanum	Key name. Maximum 11 characters
M	1	Constant	ASCII Control Character – ETX
M + 1	1	Binary	LRC check character

90.7 Select RSA-OAEP Public Key Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	iConnectEFT Constant = M90_MSR_ENCRYPTION Message Identifier – ASCII – 90.
4	1	Alphanum	iConnectEFT Constant = P90_RES_FUNCTION Function – ASCII – 7 – New RSA public key function
5	Variable	Alphanum	Result Code. • 0 = Success • 1 = Invalid request • 2 = No key found • 3 = Invalid public key • 4 = Failed to select key (file system failure)
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC check character

6.2.63.6 Using 90.x P2PE Data Messages with AES Encryption

With AES encryption, 90.8 messages are used to request key exchange, exchange encryption keys, and begin encryption.

6.2.63.6.1 Enabling AES Encryption

Enabling AES Encryption requires setting to 0091_0042 = 1 and a series of 90.8 messages. The command types for 90.8 messages are:

- 90.80 = Start key exchange
- 90.81 = Start encryption
- 90.82 = Stop encryption
- 90.83 = Encrypted data packet

The POS sends each of the above messages in order. The terminal sends a response to each message, echoing the command type. The responses include a status immediately following the command type:

- 0 = Success
- 1 = Invalid message format
- 2 = Invalid command
- 3 = Missing data
- 8 = The terminal received an encrypted message but AES is not enabled. AES key exchanged is required. Only sent as a 90.838 error response to any 90.8x message.
- 9 = The terminal received a message in clear but expect encrypted data. AES key exchanged is required. Only sent as a 90.839 error response to any 90.8x message.

If all 90.8 messages are successful, subsequent messages are encrypted with AES encryption.

Key Exchange Required

With AES encryption enabled, RBA rejects messages other than 90.83 messages by sending either 90.839 or 90.838 until a new key exchange is completed in the following scenarios:

- Communications are lost when using a session-based method such as Ethernet or Wi-Fi.
- The terminal sends an unencrypted 90.80 message.

If the POS receives either 90.839 or 90.838, restart the key exchange and stop processing encrypted messages.

6.2.63.6.2 Message Formats

90.80 Key Exchanges are the first step in initiating AES encryption. The following tables describe the request and response formats:

90.80 Key Exchange Request

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	Message Identifier – ASCII – 90.
4	1	Constant	AES with Diffie-Hellman – ASCII – 8
5	1	Constant	Key exchange – ASCII – 0
6	Variable	Alphanum	g= followed by data
M	1	Constant	ASCII Control Character – FS
M+1	Variable	Alphanum	p= followed by data
N	1	Constant	ASCII Control Character – FS
N+1	Variable	Alphanum	public= followed by data
O	1	Constant	ASCII Control Character – ETX
O+1	1	Binary	LRC Check Character

90.80 Key Exchange Response

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	Message Identifier – ASCII – 90.

Offset	Length	Type	Description
4	1	Constant	AES with Diffie-Hellman – ASCII – 8
5	1	Constant	Key exchange – ASCII – 0
6	1	Decimal	Status: • 0 = Successful • 1 = Invalid format • 2 = Missing data • 3 = Invalid parameter
7	Variable	Constant	If Status = 0, this field contains the terminals public key. Otherwise, this field is omitted.
M	1	Constant	ASCII Control Character – ETX
M + 1	1	Binary	LRC Check Character

90.81 Encryption Start is final step in initiating AES encryption. The following tables describe the request and response formats:

90.81 Encryption Start Request

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	Message Identifier – ASCII – 90.
4	1	Constant	AES with Diffie-Hellman – ASCII – 8
5	1	Constant	Encryption Start – ASCII – 1
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC Check Character

90.81 Encryption Start Response

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX

Offset	Length	Type	Description
1	3	Constant	Message Identifier – ASCII – 90.
4	1	Constant	AES with Diffie-Hellman – ASCII – 8
5	1	Constant	Encryption Start – ASCII – 1
6	1	Decimal	Status: • 0 = Successful
7	1	Constant	ASCII Control Character – ETX
8	1	Binary	LRC Check Character

The POS can send and receive 90.83 Encrypted Data Packet messages after initiating AES encryption. The following tables describe the request and response formats:

90.83 Encrypted Data Packet Request/Response

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	Message Identifier – ASCII – 90.
4	1	Constant	AES with Diffie-Hellman – ASCII – 8
5	1	Constant	Encrypted data packet – ASCII – 3
6	Variable	ASCII	Encrypted data
M	1	Constant	ASCII Control Character – ETX
M+1	1	Binary	LRC Check Character

The terminal sends 90.838 and 90.839 error messages in response to any incorrect or unexpected message. See Using 90.x P2PE Data Messages with AES Encryption.

6.2.63.6.3 Disabling AES Encryption

AES encryption can be disabled by sending a 90.82 message as described in the following tables:

90.82 Encryption Disable Request

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	Message Identifier – ASCII – 90.
4	1	Constant	AES with Diffie-Hellman – ASCII – 8
5	1	Constant	Encryption Disable – ASCII – 2
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC Check Character

90.82 Encryption Disable Response

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX
1	3	Constant	Message Identifier – ASCII – 90.
4	1	Constant	AES with Diffie-Hellman – ASCII – 8
5	1	Constant	Encryption Disable – ASCII – 2
6	1	Decimal	Status: • 0 = Successful
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC Check Character

6.2.64 91.x Print Message

Unique to the iWL250 terminal, the 91.x Printer message controls receipt printing in the RBA application. A receipt is printed by sending one or more messages that contain the information to be printed on the receipt. Each receipt is treated as a single job with a response message sent from the terminal to the POS when the job is complete or when an error is encountered.

The Print Request message is sent from the POS to the terminal to indicate that the POS has information to print. The print request follows one of these two processes:

- If the message type is 0, the entire print job is in a single message

- If the receipt data is more than 1015 characters, more than one message must be used. The first message must be type 1. If more than 2030 characters are required, type 2 messages are sent until 1015 characters or less are remaining. The final message must be type 3, which instructs the terminal to print the receipt.. To cancel a partial receipt, the POS sends a type C message.

Refer to the 91.x Barcode Printing section for a description of the commands used to configure Barcode Printing parameters.

6.2.64.1 91.x Printer Message Format

The following tables describe the format for the 91.x Print Request Message and 91.x Print Response message. The Print Response message is sent from the terminal to the POS when a print job is complete or when an error is received from the printer.

91.x Print Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX – 0x02
1	3	Constant	iConnectEFT Constant = M91_PRINTER Message Identifier – ASCII – 91.
4	1	ASCII	iConnectEFT Constant = P91_REQ_MSG_TYPE Message Type • 0 = Start new print job with all data in this message • 1 = Start a new print job with first block of data in this message • 2 = Continue printing with this data • 3 = Finish printing with this data • C = Cancel pending print job
5	Variable	ASCII	iConnectEFT Constant = P91_REQ_DATA Data to be sent to the printer (Up to 1015 ASCII characters)
M	1	Constant	ETX – 0x03
M+1	1	Binary	LRC check character

91.x Print Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII Control Character – STX – 0x02
1	3	Constant	iConnectEFT Constant = M91_PRINTER Message Identifier – ASCII – 91.

Offset	Length	Type	Description
4	1	ASCII	iConnectEFT Constant = P91_RES_RESULT Result • 0 = Successful • 3 = Paper out • 4 = Printer error • 5 = Request error (Continue or Finish message received without a Start) • N = No printer found
5	1	Constant	ETX – 0x03
6	1	Binary	LRC check character

The data portion of the message is composed of ASCII characters to be printed. Each line is terminated by a <Carriage Return> and <Line Feed>. These are the only control characters supported; all other ASCII control characters will be ignored. The data may also include special tags to control special features of the printer. The table below lists the supported tags.

Supported Tags for Printers

Start Tag	End Tag	Description
<@Cut>		Cut receipt; advances the paper of couple of lines so that the paper can be torn off.
<@Bitmap>	<@Bitmap>	Example: `91.0Please<@Bitmap>AD2.BMP<@Bitmap>on your way out` where AD2.BMP is the filename of the bitmap to print. The bitmap is printed in line with the text so text can be printed before and after the bitmap on the same line. If the bitmap is taller than a line of text, the text that follows the bitmap command could overwrite the bitmap. Line feeds and/or spaces must be used to ensure proper spacing.
<@BoldOn>	<@BoldOff>	Bolds text
<@ReverseOn>	<@ReverseOff>	White text surrounded by black
<@DoubleCharOn>	<@DoubleCharOff>	Double-wide characters

Start Tag	End Tag	Description
<@BigCharOn>	<@BigCharOff>	Very Large font Three line feeds must follow a line of big characters to prevent any overlap with the next characters or lines that must display.
<@BarCodeOn>	<@BarCodeOff>	Print a bar code (format 39) – Not supported

Number of Characters Allowed by Tag Type

Font	Default	<@DoubleCharOn>	<@BigCharOn>
Normal	40	20	6
Bold	40	20	n/a

6.2.64.2 91.x Barcode Printing

Unique to the iWL250 terminal, the following sections describe commands to configure Barcode Printing parameters such as format, size, orientation and alignment.

Unless the default values are used, the printing should be configured at least once after each terminal reboot.

6.2.64.2.1 Barcode Format

The following table lists the commands used to configure the barcode printing format. The default format is Code 39.

Commands Used to Configure Barcode Printing Format

Command	Description	Example
<@BarCode39>	Switch to Code 39 Format (Default)	<@BarCode39><@BarCodeOn>123456<@BarCodeOff>
<@BarCode128>	Switch to Code 128 Format	<@BarCode128><@BarCodeOn>123456<@BarCodeOff>
<@BarCode25>	Switch to Code 25 Format	<@BarCode25><@BarCodeOn>123456<@BarCodeOff>
<@BarCodeEAN8>	Switch to EAN-8 Format (7 digits)	<@BarCodeEAN8><@BarCodeOn>1234567<@BarCodeOff>

Command	Description	Example
<@BarCodeEAN13>	Switch to EAN-13 Format (12 digits)	<@BarCodeEAN13><@BarCodeOn>123456789012<@BarCodeOff>

6.2.64.2.2 Barcode Size

The following table lists the commands used to configure the barcode size. The default height x width of a single bar is 50x 2 pixels.

Commands Used to Configure Barcode Size

Command	Description	Example
<@BarCodeHeightNNN>	Configure the height of the barcode to NNN pixels. Theformat of NNN should be 3 digits with prefix '0' if needed. (Default is 50)	<@BarCodeHeight050><@BarCodeOn>123456<@BarCodeOff>
<@BarCodeWidthNNN>	Configure the width of 1 bar of the barcode. The format of NNN should be 3 digits with prefix '0' if needed.(*Default is 002*)	<@BarCodeWidth002><@BarCodeOn>123456<@BarCodeOff>

6.2.64.2.3 Barcode Orientation

The following table lists the commands used to configure the barcode printing orientation. The default orientation is horizontal.

Commands Used to Configure Barcode Printing Orientation

Command	Description	Example
<@BarCodeHorizontal>	Switch to horizontal layout. (Default)	<@BarCodeHorizontal><@BarCodeOn>123456<@BarCodeOff>
<@BarCodeVertical>	Switch to vertical layout.	<@BarCodeVertical><@BarCodeOn>123456<@BarCodeOff>

6.2.64.2.4 Barcode Alignment

The following table lists the commands used to configure the barcode printing alignment. The default alignment is center.

Commands Used to Configure Barcode Printing Alignment

Command	Description	Example
<@BarCodeAlignCenter>	Barcode will be printed in the center of the receipt. (Default)	<@BarCodeAlignCenter> <@BarCodeOn> 123456<@BarCodeOff>

Command	Description	Example
<@BarCodeAlignLeft>	Barcode will be printed at the left of the receipt.	<@BarCodeAlignLeft> <@BarCodeOn> 123456<@BarCodeOff>
@<BarCodeAlignRight>	Barcode will be printed at the right of the receipt.	<@BarCodeAlignRight> <@BarCodeOn> 123456<@BarCodeOff>

6.2.64.2.5 Starting and Stopping Barcode Printing

The following table lists the commands used to start and stop barcode printing.

Commands Used to Start and Stop Barcode Printing

Command	Description	Example
<@BarCodeOn>	Start barcode printing. All data after this will be printed in barcode format.	<@BarCodeOn> 123456<@BarCodeOff>
<@BarCodeOff>	Stop barcode printing. All data buffered to this point will be sent to the printer. All data after this command will be printed as text.	<@BarCodeOn> 123456<@BarCodeOff>

6.2.65 92.x Light Control

The POS sends a 92.x Light Control request to the terminal to illuminate terminal lights specified in the request. Depending on the terminal light(s) specified, colors and other attributes, such as pattern or intensity, can also be specified. Field values are not case-sensitive.

The following example request/response pair illuminates red smart card reader lights, green contactless lights, blinking green MSR reader lights, and a white display.

```
92.SMC:COLOR=Red(FS)CLESS1:color=Green(FS)MSR:color=GREEN,attribute=blink(FS)Display:color=White
```

```
92.0(FS)0(FS)0
```

The response gives status for each light specified in the order they were requested. In this example, all return 0 = success.

> If a light is listed twice in the 92.x request, only the second occurrence in the request is applied.

92.x Light Control Request Format

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td>0</td><td>1</td><td>Constant</td><td>ASCII control character – STX</td></tr>
<tr><td>1</td><td>3</td><td>Constant</td><td>Message identifier – ASCII – 92.</td></tr>
<tr><td>4</td><td>Variable</td><td>Alphanum</td><td>Light to illuminate:
• SMC - Smart card reader lights
• MSR - Magnetic stripe reader lights
• Cless1 - Contactless lights pattern 1
• Cless2 - Contactless lights pattern 2
• Cless3 - Contactless lights pattern 3
• Cless4 - Contactless lights pattern 4
• Keyboard - Keyboard lights
• Display - Display backlight</td></tr>
<tr><td>M</td><td>1</td><td>Constant</td><td>: (colon) symbol. Included only with optional color and/or intensity field(s).</td></tr>
<tr><td>M+1</td><td>Variable</td><td>Alpha</td><td>Color. Optional. If unspecified, color is treated as ON. The following table lists valid colors for each light:
<table>
<tr><th>Light</th><th>Valid colors</th></tr>
<tr><td>SMC</td><td>iUN: Off, ON, Red, Green, Blue, White or RGB in 0xRRGGBB format</td></tr>
<tr><td></td><td>Other terminals: Off, ON, Red, Green</td></tr>
<tr><td>MSR</td><td>Off, ON, Red, Green</td></tr>
<tr><td>Cless1</td><td>Off, ON, Green, Blue</td></tr>
<tr><td>Cless2</td><td>Off, ON, Green, Yellow</td></tr>
<tr><td>Cless3</td><td>Off, ON, Green</td></tr>
<tr><td>Cless4</td><td>Off, ON, Green, Red</td></tr>
<tr><td>Keyboard</td><td>Off, ON</td></tr>
<tr><td>Display</td><td>iUN: Off, ON, Red, Green, Blue, White, RGB in 0xRRGGBB format</td></tr>
<tr><td></td><td>Other terminals: Off, ON</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td>N</td><td>1</td><td>Constant</td><td>, (comma) symbol. Included only between optional color and intensity field(s).</td></tr>
<tr><td>N+1</td><td>Variable</td><td>Alphanum</td><td>Attribute (Optional). The following table describes the attributes for each light:
<table>
<tr><th>Light</th><th>Attributes</th></tr>
<tr><td rowspan="2">SMC</td><td>Intensity:<ul><li>0 - 100</li></ul></td></tr>
<tr><td>Blink</td></tr>
<tr><td>MSR</td><td>Blink in the following direction:<ul><li>l2r = left to right</li><li>r2l = right to left</li></ul></td></tr>
<tr><td>Cless1-Cless4</td><td>Ignored</td></tr>
<tr><td>Keyboard</td><td>Intensity. If no intensity is given, 80% is used.<ul><li>0 - 100</li></ul></td></tr>
<tr><td>Display</td><td>Intensity. If no intensity is given, value of 0007_0036 is used.<ul><li>0 - 100</li></ul></td></tr>
</table></td></tr>
<tr><td>O</td><td>1</td><td>Constant</td><td>ASCII control character – FS
Optional. Included only if followed by another light field.</td></tr>
<tr><td colspan="4"><b>Optional additional Light:color,attribute strings</b></td></tr>
<tr><td>P</td><td>1</td><td>Constant</td><td>ASCII control character – ETX</td></tr>
<tr><td>Q</td><td>1</td><td>Binary</td><td>LRC check character.</td></tr>
</table>

92.x Light Control Response Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	Message identifier – ASCII – 92.
4	1	Numeric	Status for the first light specified: • 0 = Success. Light changed • 1 = Light not supported by this terminal • 2 = Invalid color or attribute given. Light not changed • 3 = System error. Error changing light. Light not changed • 9 = Parsing error. Terminal unable to parse 92.x request
5	1	Constant	ASCII control character – FS Optional. Included only if followed by another Status field.
4	1	Numeric	Optional. Status for next light specified (see offset 4 for values)
Optional Additional [FS] Status Pairs			
M	1	Constant	ASCII control character – ETX
M+1	1	Binary	LRC check character

Available lights and attributes differ by terminal. The following table describes the values supported by each terminal:

Colors and Attributes Supported per Terminal

Device	SMC	MSR	CLESS1	CLESS2	CLESS3	CLESS4	KEYBOARD	DISPLAY
iPP320	OFF, ON, GREEN Always blinks	OFF, ON, GREEN Always blinks	OFF, ON, GREEN, BLUE	OFF, ON, GREEN, YELLOW	OFF, ON, GREEN	OFF, ON, GREEN, RED	Not supported - always on	OFF, ON, percent
iPP350	OFF, ON, GREEN Always blinks	OFF, ON, GREEN Always blinks	OFF, ON, GREEN, BLUE	OFF, ON, GREEN, YELLOW	OFF, ON, GREEN	OFF, ON, GREEN, RED	Not supported - always on	OFF, ON, percent
iSC250	Not supported	Not supported	OFF, ON, GREEN	OFF, ON, GREEN	OFF, ON, GREEN	OFF, ON, GREEN	Not supported - controlled with display	OFF, ON, percent (0% is not completely off)

Device	SMC	MSR	CLESS1	CLESS2	CLESS3	CLESS4	KEYBOARD	DISPLAY
iSC350	OFF, ON, red, green, blink	RED, green, on, blink, trace	OFF, ON, GREEN, BLUE	OFF, ON, GREEN, YELLOW	OFF, ON, GREEN	OFF, ON, GREEN, RED	OFF, ON, percent (0007_0013 can force backlight to stay on)	OFF, ON, percent (0% is not completely off)
iSC480	OFF, ON, GREEN	OFF, ON, GREEN, blink, trace (r2l = up, l2r = down)	OFF, ON, GREEN, BLUE	OFF, ON, GREEN, YELLOW	OFF, ON, GREEN	OFF, ON, GREEN, RED	OFF, ON, percent (0007_0013 can force backlight to stay on)	OFF, ON, percent (0% is not completely off)
iWL220, iWL228	Not supported	Not supported	OFF, ON, GREEN	OFF, ON, GREEN	OFF, ON, GREEN	OFF, ON, GREEN	Not supported - controlled with display	OFF, ON, percent
iWL250, iWL258	Not supported	Not supported	OFF, ON, GREEN, BLUE	OFF, ON, GREEN, YELLOW	OFF, ON, GREEN	OFF, ON, GREEN, RED	Not supported - controlled with display	OFF, ON, percent (0% is not completely off)
iMP350	Not supported	Not supported	OFF, ON, GREEN, BLUE	OFF, ON, GREEN, YELLOW	OFF, ON, GREEN	OFF, ON, GREEN, RED	Not supported	OFF, ON, percent
iMP352	Not supported	Not supported	OFF, ON, GREEN, BLUE	OFF, ON, GREEN, YELLOW	OFF, ON, GREEN	OFF, ON, GREEN, RED	Not supported	OFF, ON, percent
iCM122	Not supported	Not supported	OFF, ON, GREEN	OFF, ON, GREEN	OFF, ON, GREEN	OFF, ON, GREEN	Not supported	Not supported

Device	SMC	MSR	CLESS1	CLESS2	CLESS3	CLESS4	KEYBOARD	DISPLAY
iUP250, iUR250	OFF, ON, GREEN, RED, BLUE, YELLOW, WHITE, or RGB value, percent	Not supported (Same light as SMC)	OFF, ON, GREEN (Also turns on contactless logo on iUC150 or iUC150B)	OFF, ON, GREEN (Also turns on contactless logo on iUC150 or iUC150B)	OFF, ON, GREEN (Also turns on contactless logo on iUC150 or iUC150B)	OFF, ON, GREEN (Also turns on contactless logo on iUC150 or iUC150B)	Not supported	(iUP only) OFF, ON, GREEN, RED, BLUE, YELLOW, WHITE, or RGB value, percent
iUC285	OFF, ON, BLUE	OFF, ON, BLUE	OFF, ON, GREEN (Also turns on contactless logo below screen)	OFF, ON, GREEN (Also turns on contactless logo below screen)	OFF, ON, GREEN (Also turns on contactless logo below screen)	OFF, ON, GREEN (Also turns on contactless logo below screen)	Not supported	OFF, ON, percent (0% is not completely off)
iMP 650	OFF, ON, White, Blink	Not supported	OFF, ON, GREEN	OFF, ON, GREEN	OFF, ON, GREEN	OFF, ON, GREEN	Not supported	OFF, ON, percent (0% is not completely off)

6.2.66 93.x Terminal Authentication Messages

The 93.x Terminal Authentication messages are proprietary messages which authenticate communications between the payment terminal's RBA application and the POS. These messages are used to unlock a terminal following reboot. This is implemented using the compatible iConnectEFT. A new parameter has been added to the security.dat file to facilitate the terminal locking function. The user will need to edit the config.dfs file and generate a new security.dat file which will include the new '0091_0018' parameter used to implement this feature.

When the '0091_0018' Terminal Authentication flag is set to '1' in the security.dat file, the terminal is locked following each reboot. It must be unlocked before any transactions can be processed. The terminal will respond with a '00.9300' message indicating that it is locked when any message is sent from the POS, including the 01.x Online Message, until it is successfully unlocked. In order to unlock the terminal, a 93.x Terminal Authentication message must be sent to initiate the challenge/response sequence. The terminal will then respond with a '93.0' message indicating the unlocked status.

If the POS sends additional messages to the terminal before the POS has successfully replied to the terminal's 93.x message, then only the most recent message from the POS is processed (replied to) once authentication is completed. In other words, if multiple messages are sent from the POS before the challenge/response sequence has successfully completed, then only the last message is queued and replied to. Refer to the following challenge/ response sequence example.

1. Terminal boots up in locked mode.
2. 01.x Online Message is sent from the POS to the terminal.
3. Terminal responds with a '00.9300' message indicating to the POS that it is locked.
4. POS sends a '93.9C3FDC8AAA27597DDBEBE9299219EA23F3FFCA0D' message.
5. Terminal responds with a '93.0' message indicating "OK", and processes the most recent message preceding the 93.x message, which is the 01.x online message.
6. Terminal goes online.

When the terminal is powered down or rebooted, it will again boot up in locked mode and can only be unlocked using the 93.x challenge/response sequence. In order to enable or disable the locking function, a 60.x Configuration Write message must be used to set or reset the '0091_0018' Terminal Authentication flag in the security.dat file. In order to retain this setting permanently following reboot, a 00.x Offline Message or 01.x Online Message must be sent after the configuration has been changed via the 60.x message. Refer to the following tables which describe the 93.x Terminal Authentication Request message and 93.x Terminal Authentication Response message.

93.x Terminal Authentication Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M93_CHALLENGE Message Identifier – ASCII – "93."
4	1	Alphanum	iConnectEFT Constant = P93_REQ_TYPE Request type.
5	3	Alphanum	iConnectEFT Constant = P93_REQ_OPTIONS Request options.
8	Variable	Alphanum	iConnectEFT Constant = P93_REQ_DATA Request data.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

93.x Terminal Authentication Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M93_CHALLENGE Message Identifier – ASCII – “93.”
4	3	Alphanum	iConnectEFT Constant = P93_RES_OPTIONS Response options. • 0 = OK. • 1 = Failed. • 2 = Not needed.
7	Variable	Alphanum	iConnectEFT Constant = P93_RES_DATA Response data.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

6.2.67 94.x and 95.x Barcode Configuration Messages

6.2.67.1 Barcode Configuration Messages

Two messages provide barcode support:

- The 94.x messages are used to write barcode configuration, or to set barcode actions.
- The 95.x messages are used to read barcode configuration, receive current status of barcode actions, or to send barcode data.

Both 94.x and 95.x type messages follow one of three formats:

- Request messages, which are sent from the POS to the terminal.
- Response messages, which are sent from the terminal to the POS.
- Barcode data messages, which are asynchronously sent to the POS each time a barcode is scanned.

Anytime a barcode request message is sent to the terminal by the POS, the terminal should send a barcode response message. If no barcode scanner is configured in the barcode.dat file, then the terminal should respond to all barcode request messages with a 'Barcode Disabled Response Error' message (`ECR_MSG_ERR_DISABLED`). Additionally, exclusive barcode control/access must be coordinated with other RBA logic (e.g., for Bluetooth pairing/unpairing). If the application has exclusive control of the barcode, then all barcode request messages should be answered with a 'Barcode Unavailable Response Error' message (`BCR_MSG_ERR_UNAVAIL`).

The following table lists all barcode configurations. This table maps the 94.x and 95.x Barcode Configuration messages to their corresponding 60.x and 61.x configuration messages (see Note 1).

Barcode Configurations

Barcode Configuration	94.x Configuration Set	95.x Configuration Get	60.x Configuration Write	61.x Configuration Read
Configure Terminal	--	--	(see Note 1)	'61.15[GS] 1'
Configure Keyboard (see Note 2)	--	--	(see Note 1)	'61.15[GS] 2'
Scan Mode (see Note 3)	'94.10'	'95.10'	(see Note 1)	'61.15[GS] 3'
Image Mode (see Note 3)	'94.11'	'95.11'	(see Note 1)	'61.15[GS] 4'
Illumination Mode (see Note 3)	'94.12'	'95.12'	(see Note 1)	'61.15[GS] 5'
Lightning Mode (see Note 3)	'94.13'	'95.13'	(see Note 1)	'61.15[GS] 6'
Trigger Enabled (see Note 3)	'94.20'	'95.20'	(see Note 1)	'61.15[GS] 7'
Symbologies Enabled	'94.30'	'95.30'	(see Note 1)	'61.15[GS] 8'
Symbology Encryption Enabled	--	'95.31'	(see Note 1)	'61.15[GS] 9'
Symbology Encryption Type	--	'95.32'	(see Note 1)	'61.91[GS] 19'
RSA Encryption Key	--	--	(see Note 1)	'61.91[GS] 20'
RSA Encryption Exponent	--	--	(see Note 1)	'61.91[GS] 21'
RSA Encryption Key ID	--	--	(see Note 1)	'61.91[GS] 22'

Note 1: 60.x configuration write messages are currently disabled for barcode configuration.
Note 2: Currently applies to USB barcode scanners only.
Note 3: Currently applies to iSMP barcode scanners only.

Barcode Actions

The following table lists all barcode action messages.

Barcode Action Messages

Barcode Action	94.x Set Action	95.x Get Action Status
Reset	'94.00'	--
Power	'94.01'	'95.01'
Scan	'94.02'	'95.02'

Barcode Action	94.x Set Action	95.x Get Action Status
Bulk Scan	'94.04'	'95.04'
Data	--	'95.09' (see Note 4)

> Note 4: The '95.09' Barcode Data messages are sent from the terminal for each barcode scanned. The '95.09' messages cannot be sent to the terminal to request barcode data.

6.2.67.1.1 Barcode Bulk Scanning

Currently only available on the iSMPc, bulk scanning allows the Terminal to store barcodes in memory after each scan rather than pushing them out to the POS after each. Afterwards, the POS can request all barcodes at once.

Barcode bulk scan is enabled with default barcode configurations (symbology, illumination mode etc.) The trigger is enabled by default for bulk scan, and returns to its previous setting when bulk scan is disabled. Scan mode is set to “Single Scan Mode” and also returns to its previous mode when bulk scan is disabled. Only the following messages are supported during bulk scan:

- '94.0400' for Barcode Disable
- 61.x Parameter Read Message
- 11.x Status Message
- 07.x/08.x Unit Data/ Health Status Message
- 97.x Reboot Message
- 95.x Barcode Get Message

After a barcode is scanned, scanned data is displayed on the screen on the first 3 lines with a clear entry field for entering quantity.

- If <Cancel> is pressed on this screen, the barcode is discarded.
- If no quantity is submitted before a 94.0400 Barcode Cancel message, the last item scanned is saved with a quantity of 1.
- If a new barcode is scanned, the existing barcode is saved with default quantity of '1' and the new barcode is displayed on the screen.
- If <Enter> is pressed after entering any value, the barcode is saved with the entered quantity and the screen goes back to the initial “Barcode Bulk Scanning” screen.
- If a Barcode Read error happens, the existing barcode is saved and the error message “Barcode Read Error” is displayed on the screen for 3 seconds. The screen then goes back to the initial “Barcode Bulk Scanning” screen.

Barcode undo operation can be achieved by pressing “Prev Scan” Button on the initial Bulk Scan screen.

- If there is no stored scan, on press of "Prev Scan" button, “No Records Found” is displayed for 3 seconds and Screen goes back to the initial “Barcode Bulk Scanning” screen.
- If there is at least one stored scan, on press of “Prev Scan” button a new form is displayed with last scanned barcode data with its quantity. “Undo” and “Keep” buttons are displayed on the form.
    - On press of “Undo”, the stored scan is deleted and the screen goes back to the initial “Barcode Bulk Scanning” screen.

  - On press of “Keep” or Cancel Key, Undo operation is discarded and the screen goes back to the initial “Barcode Bulk Scanning” screen.

6.2.67.2 *Barcode Message Tables*

The following sections provide configuration message tables for the 94.x and 95.x barcode messages, including; character offset, character length, character type, and description:

- Barcode Data Message
- Barcode Encryption Messages
- Barcode Illumination Messages
- Barcode Image Messages
- Barcode Lighting Messages
- Barcode Power Messages
- Barcode Reset Messages
- Barcode Scan Messages
- Barcode Symbology Messages
- Barcode Trigger Messages

6.2.67.3 *Barcode Data Message*

Barcode Data Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 09 = Barcode data read.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 9 = No error; barcode data returned in plain text. • 8 = No error; barcode data returned encrypted. • 2 = Generic error. • R = barcode data read error. • E = Encryption error.

Offset	Length	Type	Description
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code: • 09 = Data read.
9	Variable	Decimal	iConnectEFT Constant = P95_RES_SYMBOLOGY_LIST Barcode symbology (see Note 1)
M	Variable	Alphanumeric	iConnectEFT Constant = P95_RES_BARCODE_DATA Data, if no errors (see Note 2 and Note 3).
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

Note 1: Returned symbology should be a (positive) decimal code corresponding to the scanned barcode's symbology type. Symbology code will (currently) be returned as a two-digit decimal value. '-1' should be returned if symbology of scanned barcode is unknown (or for any barcode read errors).
Note 2: All barcode data for all symbologies, whether encrypted or plain text, is always encoded in Base64 ASCII (in case of binary data). However, for any error(s), no barcode data is returned.
Note 3: If there is any error, then Data will not be returned (but Barcode Symbology is returned even if '-1').

6.2.67.4 Barcode Encryption Messages

6.2.67.4.1 **Barcode Configure Symbology Encryption Messages**

Barcode Configure Symbology Encryption Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 31 = Symbology encryption.

Offset	Length	Type	Description
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 01 = Enabled. Configuration only enables encryption for listed symbologies.
8	Variable	Alphanumeric	iConnectEFT Constant = P94_REQ_SYMBOLOGY_LIST Symbology list: 'XX' Decimal barcode symbology codes(s), comma separated (see Note 2).
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Symbology list may include only comma-separated, (non-negative) decimal codes corresponding to desired barcode symbologies to enable. ‘0’/’00’ may be used as a solitary symbology code to enable all symbologies. Each symbology configuration message overwrites the previously configured/enabled symbology list.

Barcode Configure Symbology Encryption Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 31 = Symbology encryption.
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error Code: • 0 = No error. • C = Configuration error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

6.2.67.4.2 Barcode Read Symbology Encryption Configuration Messages

Barcode Read Symbology Encryption Configuration Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 31 = Symbology encryption.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Read Symbology Encryption Configuration Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 31 = Symbology encryption.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error. • C = Configuration error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note 3): • 01 = Enabled (see Note 1).

Offset	Length	Type	Description
M	Variable	Alphanumeric	iConnectEFT Constant = P95_RES_SYMBOLOGY_LIST Symbology list, if no errors (see Note 3): 'XX' Decimal barcode symbology codes(s), comma separated (see Note 2).
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

Note 1: Configuration read only returns all symbologies with encryption enabled.
Note 2: Returned symbology lists comma-separated, (positive) decimal codes corresponding to currently enabled barcode symbologies. Example: "13,23,33,41" indicates Code39, Code128, PDF417, and QR barcode symbologies are enabled to encrypt barcode data.
Note 3: If there is any error, then the Action Code and Symbology List will not be returned.

Barcode Read Encryption Type Configuration Messages

Barcode Read Encryption Type Configuration Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 32 = Encryption type.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Read Encryption Type Configuration Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 32 = Encryption type.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error. • C = Configuration error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note): • 00 = Encryption disabled. • 09 = RSA encryption.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

If there is any error,then the Action Code will not be returned.

6.2.67.5 Barcode Illumination Messages

6.2.67.5.1 **Barcode Configure Illumination Mode Messages**

Barcode Configure Illumination Mode Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character - STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td>4</td><td>2</td><td>Decimal</td><td>iConnectEFT Constant = P94_REQ_TYPE_CODE
Type code:<ul><li>12 = Illumination mode.</li></ul></td></tr>
<tr><td>6</td><td>2</td><td>Decimal</td><td>iConnectEFT Constant = P94_REQ_ACTION_CODE
Action code:
<table>
<tr><th>Action Code</th><th>Scan LED Lights</th><th>Aimer LED Lights</th></tr>
<tr><td>'00'</td><td>OFF</td><td>OFF</td></tr>
<tr><td>'01'</td><td>ON</td><td>OFF</td></tr>
<tr><td>'02'</td><td>OFF</td><td>ON</td></tr>
<tr><td>'03'</td><td>ON</td><td>ON</td></tr>
</table></td></tr>
<tr><td>8</td><td>1</td><td>Constant</td><td>ASCII control character - ETX</td></tr>
<tr><td>9</td><td>1</td><td>Binary</td><td>LRC check character.</td></tr>
</table>

Barcode Configure Illumination Mode Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character - STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 12 = Illumination mode.
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • C = Configuration error.
7	1	Constant	ASCII control character - ETX

Offset	Length	Type	Description
8	1	Binary	LRC check character.

6.2.67.5.2 Barcode Read Illumination Mode Configuration Messages

Barcode Read Illumination Mode Configuration Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character - STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 12 = Illumination mode.
6	1	Constant	ASCII control character - ETX
7	1	Binary	LRC check character.

Barcode Read Illumination Mode Configuration Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character - STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 12 = Illumination mode.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error. • C = Configuration error.

Offset	Length	Type	Description
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note): <table><tr><th>Action Code</th><th>Scan LED Lights</th><th>Aimer LED Lights</th></tr><tr><td>'00'</td><td>OFF</td><td>OFF</td></tr><tr><td>'01'</td><td>ON</td><td>OFF</td></tr><tr><td>'02'</td><td>OFF</td><td>ON</td></tr><tr><td>'03'</td><td>ON</td><td>ON</td></tr></table>
M	1	Constant	ASCII control character - ETX
M + 1	1	Binary	LRC check character.

If there is any error, then the Action Code will not be returned.

6.2.67.6 Barcode Image Messages

6.2.67.6.1 **Barcode Configure Image Mode Messages**

Barcode Configure Image Mode Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 11 = Image mode.

Offset	Length	Type	Description
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 01 = 1D mode. • 02 = 2D mode. • 03 = 2D mode for bright environment. • 04 = 2D mode for shiny/reflective surfaces.
8	1	Constant	ASCII control character – ETX
9	1	Binary	LRC check character.

Barcode Configure Image Mode Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character– STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 11 = Image mode.
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • C = Configuration error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

6.2.67.6.2 Barcode Read Image Mode Configuration Messages

Barcode Read Image Mode Configuration Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 11 = Image mode.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Read Image Mode Configuration Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 11 = Image mode.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error. • C = Configuration error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note 1): • 01 = 1D mode. • 02 = 2D mode (see Note 2). • 03 = 2D mode for bright environment. • 04 = 2D mode for shiny/reflective surfaces.
M	1	Constant	ASCII control character – ETX

Offset	Length	Type	Description
M +1	1	Binary	LRC check character.

Note 1: If there is any error, then the Action Code will not be returned.
Note 2: All 2D modes also include 1D mode.

6.2.67.7 Barcode Lighting Messages

6.2.67.7.1 **Barcode Configure Lighting Mode Messages**

Barcode Configure Lighting Mode Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 13 = Lighting mode.
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 00 = Shorter exposure time. • 01 = Longer exposure time (for shiny/reflective surfaces; see 'Configure Image Mode,' 'Action Code 04.'
8	1	Constant	ASCII control character – ETX
9	1	Binary	LRC check character.

Barcode Configure Lighting Mode Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."

Offset	Length	Type	Description
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 13 = Lighting mode.
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • C = Configuration error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

6.2.67.7.2 Barcode Read Lighting Mode Configuration Messages

Barcode Read Lighting Mode Configuration Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 13 = Lighting mode.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Read Lighting Mode Configuration Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."

Offset	Length	Type	Description
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 13 = Lighting mode.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error. • C = Configuration error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note 1): • 00 = Shorter exposure time; priority to illumination LEDs for less blurred images. • 01 = Longer exposure time; priority to aperture for shiny/reflective surfaces.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Note 1
If there is any error, then the Action Code will not be returned.

6.2.67.8 Barcode Power Messages

6.2.67.8.1 **Barcode Set Power Messages**

Barcode Set Power Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."

Offset	Length	Type	Description
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 01 = Power.
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 00 = Off. • 01 = On.
8	1	Constant	ASCII control character – ETX
9	1	Binary	LRC check character.

Barcode Set Power Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 01 = Power.
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

6.2.67.8.2 Barcode Get Power Status Messages

Barcode Get Power Status Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 01 = Power.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Get Power Status Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 01 = Power.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note 1): • 00 = Off. • 01 = On.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Note 1
If there is any error, then the Action Code will not be returned.

6.2.67.9 Barcode Reset Messages

Barcode Reset Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 00 = Reset (see Note 1).
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 00 = None (see Note 2).
8	1	Constant	ASCII control character – ETX
9	1	Binary	LRC check character.

Barcode Reset Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 00 = Reset.

Offset	Length	Type	Description
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

> Note 1: Barcode Reset Request is a 94.x write-only message that restores RBA's default `barcode.dat` configuration and powers off the barcode reader.
> Note 2: No (other) Action codes are currently supported.

6.2.67.10 Barcode Scan Messages

6.2.67.10.1 **Barcode Set Scan Messages**

Barcode Set Scan Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 02 = Scan.
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 00 = Stop. • 01 = Start.
8	1	Constant	ASCII control character – ETX'
9	1	Binary	LRC check character.

Barcode Set Scan Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 02 = Scan.
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

6.2.67.10.2 Bulk Scan Messages

Barcode Bulk Scan Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 04 = Bulk Scan.

Offset	Length	Type	Description
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 00 = Stop. • 01 = Start. • 02 = Get count of stored scans • 03 = Read all records. RBA will send one response per 94.0403 request. • 04 = Delete all records. Once read record is sent by POS, the terminal responds with a 94.x Message containing barcode data: • If Barcode record is below 4k length then 1 94.x response is sent per barcode record. • If Barcode record is more than 4k length then multiple 94.x responses are sent per record.
8	1	Constant	ASCII control character – ETX'
9	1	Binary	LRC check character.

Barcode Bulk Scan Response Message (to '94.0401', '94.0400', and '940404')

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 04 = Bulk Scan.
6	1	Alphanumeric	Status code: • 0 = Success. • 1 = Failed.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

Barcode Bulk Scan Response Message (to '94.0402')

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 04 = Bulk Scan.
6	2	Alphanumeric	Action code: • 02 = Return stored scan count.
8	1	Decimal	Status code: • 0 = Success. • 1 = Failed.
9	Variable	Decimal	Scan count.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

RBA will send one response per 94.0403 request.

Barcode Bulk Scan Read All Record Response Message (to '94.0403')

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 04 = Bulk Scan.
6	2	Alphanumeric	Action code: • 03 = Read all records.

Offset	Length	Type	Description
8	1	Decimal	Status code: • P = Partial. There are more packets to come for this record. • F = Failed due to bad memory. • C = Either data is complete for a short record, or this is the last packet of a long record. • N = No records. • E = End of all records.
9	4	Decimal	Packet Number.
13	4	Decimal	Total number of packets.
17	2	Alphanumeric	Symbology.
19	4	Decimal	Barcode data length.
23	Variable	Alphanumeric	Base64 encoded barcode data.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Below example shows details of the Read All Record Response:

If there were three scans, all with symbology 13, as follows -

- Data (in Base64) is AAAAAA (length 6), quantity 5
- Data (in Base64) is BBBBB….BBBBB (length 6000), quantity 7
- Data (in Base64) is CCCCCCCC (length 8), quantity 9

Then RBA would send these responses -

- 94.0403C00010005130006AAAAAA
- 94.0403P00020007136000BBBBBB…BBBBBBBBBBBBBBBBBBB
- 94.0403C00030007136000BBBBBB…BBB
- 94.0403E00040009130008CCCCCCCC

6.2.67.10.3 Barcode Get Scan Messages

Barcode Get Scan Status Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 02 = Scan.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Get Scan Status Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 02 = Scan.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note 1) • 00 = Stop. • 01 = Start.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Note 1: If there is any error, then the Action Code will not be returned.

6.2.67.10.4 Barcode Configure Scan Mode Messages

Barcode Configure Scan Mode Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 10 = Scan mode.
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 01 = Single scan mode. • 02 = Multi-scan mode.
8	1	Constant	ASCII control character – ETX
9	1	Binary	LRC check character.

Barcode Configure Scan Mode Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 10 = Scan mode.

Offset	Length	Type	Description
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • C = Configuration error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

6.2.67.10.5 Barcode Read Scan Mode Configuration Messages

Barcode Read Scan Mode Configuration Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 10 = Scan mode.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Read Scan Mode Configuration Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 02 = Scan mode.

Offset	Length	Type	Description
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error. • C = Configuration error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note 1) • 00 = Stop. • 01 = Start.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Note 1
If there is any error, then the Action Code will not be returned.

6.2.67.11 Barcode Symbology Messages

6.2.67.11.1 Barcode Configure Symbologies Messages

Barcode Configure Symbologies Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 30 = Symbologies.
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 01 = Enabled (see Note 1).

Offset	Length	Type	Description
8	Variable	Alphanumeric	iConnectEFT Constant = P94_REQ_SYMBOLOGY_LIST Symbology list: XX Decimal barcode symbology code(s), comma separated (see Note 2)
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Barcode Configure Symbologies Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 30 = Symbologies.
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • C = Configuration error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

Note 1: Configuration only enables listed symbologies.
Note 2: Symbology list should include only comma-separated, (non-negative) decimal codes corresponding to desired barcode symbologies to enable. ‘0’/’00’ may be used as a solitary symbology code to enable all symbologies. Each symbology configuration message overwrites the previously configured/enabled symbology list.
Example: "13,23,33,41" enables Code39, Code128, PDF417, and QR barcode symbologies.

6.2.67.11.2 Barcode Read Symbologies Configuration Messages

Barcode Read Symbologies Configuration Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 30 = Symbologies.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Read Symbologies Configuration Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 30 = Symbologies.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error. • C = Configuration error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note 2) • 01 =Enabled (see Note 1).

Offset	Length	Type	Description
M	Variable	Alphanumeric	iConnectEFT Constant = P95_RES_SYMBOLOGY_LIST Symbology list, if no errors (see Note 3): 'XX' Decimal barcode symbology code(s), comma separated (see Note 2).
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

Note 1: Configuration read only returns all enabled symbologies.
Note 2: Returned symbology lists comma-separated, (positive) decimal codes corresponding to currently enabled barcode symbologies. Example: "13,23,33,41" indicates Code39, Code128, PDF417, and QR barcode symbologies enabled.
Note 3: If there is any error, then the Action Code and Symbology List will not be returned.

6.2.67.11.3 Barcode Symbology Codes

The following table provides a description for each of the symbology codes

Code	Description
-1	Unknown
0	All symbologies
1	EAN13
2	EAN8
3	UPCA
4	UPCE
5	EAN13_2
6	EAN8_2
7	UPCA_2
8	UPCE_2
9	EAN13_5
10	EAN8_5
11	UPCA_5
12	UPCE_5

Code	Description
13	Code 39
14	*** N/A ***
15	Interleaved 2 of 5
16	Standard 2 of 5
17	Matrix 2 of 5
18	*** N/A ***
19	CodeBar
20	AmesCode
21	MSI
22	Pleassey
23	Code 128
24	Code 16k
25	Code 93
26	Code 11
27	Telepen
28	Code 49
29	Code 39_ItalianCPI
30	Codablock A
31	Codablock F
32	Codablock 256
33	PDF417
34	GSI_128
35	ISBT128
36	MicroPDF
37	GSI_DataBarOmni
38	GSI_DataBarLimited
39	GSI_DataBarExpanded

Code	Description
40	DataMatrix
41	QRCode
95	GSI DataBar Omni-Dir Composite (CC-A)
44	GSI DataBar
45	GS1 DataBar Expanded Composite (CC-A)
46	GS1 Composite/GS1-128 Composite (CC-A)
47	EAN-13 Composite (CC-A)
48	EAN-8 Composite (CC-A)
49	UPC-A Composite (CC-A)
50	UPC-E Composite (CC-A)
51	GS1 DataBar Omni-Dir Composite (CC-B)
52	GS1 DataBar Limited Composite (CC-B)
53	GS1 DataBar Expanded Composite (CC-B)
54	GS1 Composite/GS1-128 Composite (CC-B)
55	EAN-13 Composite (CC-B)
56	EAN-8 Composite (CC-B)
57	UPC-A Composite (CC-B)
58	UPC-E Composite (CC-B)
59	GS1 Composite/GS1-128 Composite (CC-C)
60	ISBN
61	Postnet
62	Planet
63	BPO
64	Canada Post
65	Australian Post
66	Japan Post
67	Dutch Post

Code	Description
68	China Post
69	Korean Post
70	TLC39
71	Trioptic
72	ISMN
73	ISSN
74	Aztec
75	Sweden Post
76	Infomail
77	Multicode
78	Incomplete Multicode

6.2.67.12 Barcode Trigger Messages

6.2.67.12.1 **Barcode Configure Trigger Messages**

Barcode Configure Trigger Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_REQ_TYPE_CODE Type code: • 20 = Trigger.
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 00 = Disabled. • 01 = Enabled.
8	1	Constant	ASCII control character – ETX

Offset	Length	Type	Description
9	1	Binary	LRC check character.

Barcode Configure Trigger Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M94_BARCODE_SET ASCII Message identifier, "94."
4	2	Decimal	iConnectEFT Constant = P94_RES_TYPE_CODE Type code: • 20 = Trigger.
6	1	Alphanumeric	iConnectEFT Constant = P94_RES_STATUS Status/Error code: • 0 = No error. • C = Configuration error.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

6.2.67.12.2 Barcode Read Trigger Configuration Messages

Barcode Read Trigger Configuration Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_REQ_TYPE_CODE Type code: • 20 = Trigger.
6	1	Constant	ASCII control character – ETX
7	1	Binary	LRC check character.

Barcode Read Trigger Configuration Response Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M95_BARCODE_GET ASCII Message identifier, "95."
4	2	Decimal	iConnectEFT Constant = P95_RES_TYPE_CODE Type code: • 20 = Trigger.
6	1	Alphanumeric	iConnectEFT Constant = P95_RES_STATUS Status/Error code: • 0 = No error. • 1 = (reserved; do not use). • 2 = Generic error. • C = Configuration error.
7	2	Decimal	iConnectEFT Constant = P95_RES_ACTION_CODE Action code, if no errors (see Note 1). • 00 = Disabled. • 01 =Enabled.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

Note 1
If there is any error, then the Action Code will not be returned.

6.2.68 97.x Reboot

This message reboots the terminal.

Reboot Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
1	3	Constant	iConnectEFT Constant = M97_REBOOT Message Identifier – ASCII – “97.”
M	1	Constant	ASCII control character – ETX
M + 1	Variable	Binary	LRC check character.

Info
The 62.x File Write message is used to load files to the terminal. In the event the terminal does not automatically reboot after a file update, a manual reboot should be performed. The following table outlines which file types require a manual reboot:

File Type Updated	Automatic Reboot: Terminal reboots automatically after update	Manual Reboot: Send 97.x Message or press Terminal key combination for manual reboot
.OGZ	X	
.PGZ		X
.TGZ		X
.K3Z		X

The 62.x File Write message is intended to update single files whereas larger files can be uploaded much more rapidly via IBMEFT download or by using TMS.

7 Configuring Encryption

This section describes point-to-point encryption and lists the supported encryption types and how to configure them.

7.1 P2PE Card Data Encryption

7.1.1 Enabling Point-to-Point Encryption in RBA

RBA users can enable P2PE by adjusting the following parameters, found in the Security Parameters (security.dat) section of `config.dfs.` Additional parameters in that section are used to set parameters for the individual encryption methods.

Parameters for Enabling MSR Encryption

config.dfs Parameter	Description
0091_0001	Set the value according to the table in Supported Encryption Methods
0091_0002	Set the value to key slot index that contains the data encryption key (if applicable).

> Neither of these values can be set using a 60.x Configuration Write message. To set these, users will have to edit the `security.dat` file in config.dfs directly, and obtain signature and a new .PGZ file prior to implementation. See Signing Requirements for .DAT File Changes for details.

7.1.2 Supported Encryption Methods

The following table shows the encryption methods which are supported by RBA. The encryption type is selected using parameter 0091_0001 in `SECURITY.DAT`.

Some encryption methods use a DUKPT key, which requires an injected key slot configured in parameter 0091_0002.

Other methods use a public/private key system with configuration details noted in the description for each method.

Encryption Methods' Support Status

Encryption Method	Support Status	0091_0001 Encryption Type number	Uses injected DUKPT key?
Magtek MagneSafe™ POS	Not supported	1	
On-Guard	√	2	Yes

Encryption Method	Support Status	0091_0001 Encryption Type number	Uses injected DUKPT key?
EPS	√	3	Yes
Voltage TEP1	√ (Cannot be used with TailGate)	4	No
Voltage TEP2	√ (Cannot be used with TailGate)	5	No
Voltage TEP4	√ (Cannot be used with TailGate)	6	No
Monetra CardShield	√	7	Yes
Mercury Payment Systems (MPS)	Not supported	8	
RSA-OAEP	√	9	No
TransArmor	√	10	No
TDES DUKPT Generic	√	11	Yes
S1	√	12	Yes
ECC	Not supported.	13	
P2PE Encryption for NCR/Retalix	√	14	Yes
Voltage TEP1x	√ (Cannot be used with TailGate)	15	No
Voltage TEP2x	√ (Cannot be used with TailGate)	16	No

Encryption or masking cards with PANs containing fewer than nine digits is not supported (minimum of 12 digits for Voltage encryption types). Merchants should either whitelist these cards or disable non-standard card encryption.

7.1.3 Encryption Requirements

7.1.3.1 Encryption Requirements

With certain encryption types set, the RBA will encrypt only if certain conditions are met. The table below explains the minimum requirements for each encryption type supported and used by the RBA:

Encryption Requirements by Encryption Type

Encryption Method	Minimum Requirements	Description
EPS P2PE	- Track 1 or Track 2 must be read successfully. - PAN must include at least 9 characters.	EPS (Element Payment Systems) P2PE Encryption
Monetra CardShield	- Track 1 or Track 2 must be read successfully. - PAN must include at least 9 characters.	Monetra CardShield Encryption
On-Guard	- Track 2 with sentinels for swiped or MSD card. - Track 2 Equivalent or PAN for EMV card. - PAN followed by expiration date. - PAN must be at least 9 and at most 37 characters.	On-Guard Encryption Configuration
RSA-OAEP	- Track 1, Track 2, and Track 3 data. - If manually entered, then the input must include the account number, expiration date and CVV2. - PAN must include at least 9 characters.	RSA-OAEP and TransArmor Encryption
S1	- Track 2 with both account number and expiration date field - Minimum 3 bytes of discretionary data.	S1 Encryption
TDES DUKPT Generic TDES DUKPT Encryption for NCR/Retalix	- Track 1 or Track 2 must be read successfully, or data must be manually entered. - PAN must include at least 9 characters.	Generic TDES DUKPT Encryption TDES DUKPT Encryption for NCR/Retalix

Encryption Method	Minimum Requirements	Description
TransArmor	• Raw Track 1 or Track 2 data, or manually entered PAN. • PAN must include at least 9 characters.	RSA-OAEP and TransArmor Encryption
Voltage TEP1, TEP1x	• PAN must include at least 12 digits. • Track data must include at least one complete PAN.	Voltage TEP1 and TEP2 Encryption Voltage TEP1x, TEP2x, and TEP4 Encryption
Voltage TEP2, TEP2x, TEP4	• PAN must include at least 12 digits. • PAN must be successfully read from track data or manually entered data.	Voltage TEP1 and TEP2 Encryption Voltage TEP1x, TEP2x, and TEP4 Encryption

7.1.4 Encryption Processing

When Point-to-Point Encryption (P2PE) is enabled, cardholder data are encrypted immediately after being captured. This happens regardless of the source of the data: magnetic stripe, contactless magstripe, contact EMV or contactless EMV, or manual entry. In the case of manual entry, RBA constructs a dummy Track 2 containing the keyed information, which is then used as if a magnetic stripe had been read.

By default, in addition to providing encrypted data blocks to the POS, RBA also provides the first six digits and the last four digits of card numbers in the clear when encryption is enabled. The middle digits are masked. To modify these settings, parameters '0091_0003' (unmasked leading digits) and '0091_0004' (unmasked trailing digits) can be changed.

Depending on the encryption method and the method of card entry (especially EMV versus non-EMV), the masked data and encrypted data are returned to the POS in different fields of RBA messages and in different EMV tags. The specifics are described in the sections for each encryption method.

The contents of some RBA variables are also affected. For example, when P2PE is enabled, the account number returned in variable 398 (Card read On-Demand account number) or 401 (Payment card account number) contains the masked account number. See Retrieving card information using the 29.x (Get Variable) Message for details.

Also note that the account number (PAN) is required to create a PIN block. With P2PE enabled, the POS does not have the actual PAN; it can substitute the masked PAN instead. See Requesting the PIN Block Using the Masked PAN for details.

If BIN range checking is enabled, specific card types can be included/excluded from encryption using the corresponding flag for each BIN range. To set up this process, Security BIN (secbin.dat) parameter '0092_0001' should be set to '1' (enabling security bin table checking). Also, parameters '0092_0002' through '0092_0030' should be modified as needed for inclusion/exclusion of card types and BIN ranges.

The "Spin-the-Bin" process works as usual, except that the account number in the 19.x response has all but the first 6-9 digits masked. `config.dfs` parameter '0005_0008' sets the number of digits.

If P2PE is attempted and the operation fails (for example, an injected key is not present), RBA shuts down and the terminal goes offline.

Icon

Encryption or masking of cards with PANs containing less than 9 digits is not supported. Merchants should either whitelist these cards or disable non-standard card encryption.

7.1.5 Retrieving card information using the 29.x (Get Variable) Message

The 29.x Get Variable Request can be used to retrieve the following pieces of information. The Description column shows the information that will be returned when P2PE is enabled:

Using the 29.x Message to Retrieve Cardholder Data

Information Type	Variable	Description
Mod-10 check value	396	Mod-10 check digit in card read transaction flow. Set to 'F' if the Mod-10 check fails.
Mod-10 check value	397	Mod-10 check digit in 23.x Card Read Request (On-Demand). Set to 'F' if the Mod-10 check fails.
Masked PAN (with first 6 and last 4 digits in the clear)	398	Used to hold the masked PAN for cards read from the "card read request" form
	401	Used to hold the masked PAN for cards read the "swipe" form
Name	399	Used to hold the name for cards read from the "card read request" form
	402	Used to hold the name for cards read from the "swipe" form
Expiration Date	400	Used to hold the expiration date for cards read from the "card read request" form
	403	Used to hold the expiration date for cards read from the "swipe" form
Service Code	413	Service code. This variable is always available for card type verification whether encryption is enabled or not. The first digit of the Service Code is 2 or 6 for an EMV card. The POS can use this to determine if a card that was swiped is actually an EMV card, and should be inserted instead. It is generally up to the POS to control this process.
Track 1	406	Used to retrieve masked Track 1 data
Track 2	407	Used to retrieve masked Track 2 data

Information Type	Variable	Description
Track 3	411	Used to retrieve Track 3 data

7.1.6 Requesting the PIN Block Using the Masked PAN

With P2PE enabled, the clear PAN is never provided to the POS, but it is required to generate a PIN block.

To accommodate this, RBA allows the POS to request a PIN block using the masked PAN. This is because RBA stores the original PAN during a transaction, and and associates it with the masked PAN that is sent to the POS.

When the POS sends a 31.x PIN Request message with the PAN masked, RBA compares that masked PAN to the masked PAN that it stored.

- If the two masked PANs match, the terminal uses the original PAN to encrypt the PIN block
  -or-
- If the PANs do not match, the terminal encrypts the PIN block using the masked PAN coming from the POS. (This normally results in a failed decryption of the PIN block).

7.1.7 Mod-10 Checking

The application uses the Mod-10 (Luhn) formula to verify the account number (PAN) using a check digit. The check digit verifier is included in the application to identify PANs that are incorrectly entered (miskeyed), allowing the application to prompt for re-entry of the PAN. This is very important in the case of network-down situations, wherein the POS logic stands in for the authorization and forwards the account information upon network return.

The Mod-10 calculation will occur regardless of any configuration parameter settings. Specific Mod-10 BIN settings only determine whether cards with invalid Mod-10 check digits are accepted or rejected (such as in the case where the last digit of PAN does not match the Mod-10 calculation).

If DFS data index '0099_0001' is set to a value of ‘1’ (to enable BIN range checking), the MOD-10 flag in each “BINx.DAT” file must be set to a value of ‘1’. Currently, the MOD-10 calculation is the only supported card status validation check. Refer to the Card Transaction Codes table in BIN Processing (allBins.dat, bin0.dat - bin20.dat) for information about enabling Mod-10.

The following Mod-10 check digit variables are always updated:

- Variable number 396 (Mod-10 check digit in card read transaction flow).
- Variable number 397 (Mod-10 check digit in 23.x Card Read Request (On-Demand)).

7.1.8 Loading Key Serial Number (KSN) Data

Before MSR Encryption processing can begin, the terminal needs to be loaded with keys (also known as key injection). During the loading process, an administrator injects key serial number (KSN) data into the terminal. Not all encryption types (such as Voltage) require key injection.

> Terminal KSN data is visible next to KSN indexes 0 and 6 in the terminal’s TSA application. See your terminal’s **Operations Guide** for information on accessing terminal applications other than RBA.

7.1.9 Selecting Specific Cards to be Encrypted

By default, MSR encryption encrypts all cards. To specify only certain types of cards to be encrypted, configure the Security BIN (secbin.dat) file for the appropriate bin ranges. See RSA-OAEP and TransArmor Encryption for additional information specific to those encryption types.

7.1.10 Signing Requirements for .DAT File Changes

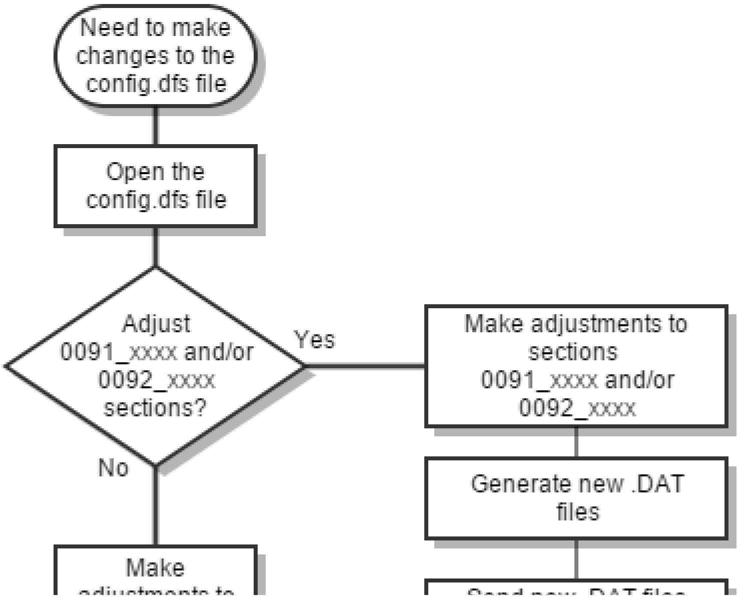
If your organization requires changes to the `security.dat` and/or `secbin.dat` files, you must follow the procedure outlined in this section. Changes to these .DAT files require specific Ingenico signature and approval prior to your implementation.

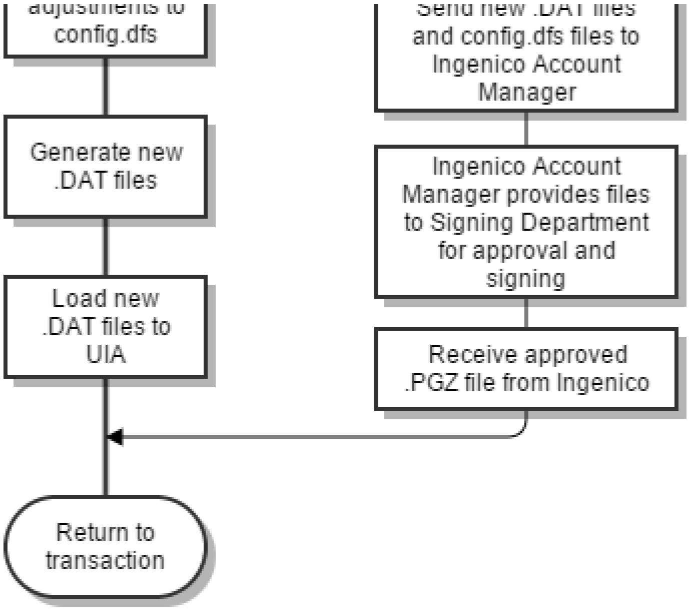
- **`Security.dat`** - Security parameters. These parameters must be signed and cannot be changed by messages. See section Security Parameters (security.dat)for parameter details.
- **`Secbin.dat`** - Security BIN table. These parameters must be signed and cannot be changed by messages. See also RSA-OAEP and TransArmor Encryption for additional information specific to RSA-OAEP and TransArmor encryption types. Security BIN (secbin.dat) for parameter details.

After approval and signature of any changes to your `security.dat` and/or `secbin.dat` files, you will receive a new .PGZ file from Ingenico. You will be unable to implement your changes to the `security.dat` and `secbin.dat` files without a signed .PGZ file from Ingenico. If you implement your changes prior to receipt of the new .PGZ file, your Telium terminals may appear to run properly, however, your terminals will actually be running as previously configured, without your changes. See the process diagram on the following page for approval process information.

> All parameters except for those located in the `security.dat` and `secbin.dat` files may be changed using the 60.x Configuration Write Message.

Contact your Ingenico Account Manager with any questions you may have about the signing process.


Need to make changes to the config.dfs file
Open the config.dfs file
Adjust 0091_xxxx and/or 0092_xxxx sections?
Yes
No
Make adjustments to sections 0091_xxxx and/or 0092_xxxx
Generate new .DAT files
Make adjustments to
Send new .DAT files


adjustments to config.dfs
Generate new .DAT files
Load new .DAT files to UIA
Send new .DAT files and config.dfs files to Ingenico Account Manager
Ingenico Account Manager provides files to Signing Department for approval and signing
Receive approved .PGZ file from Ingenico
Return to transaction

MSR Encryption Configuration Process

7.1.11 Encryption Whitelisting

In RBA there are two methods to whitelist cards so they are not encrypted:

- Security BIN (secbin.dat) sets RBA-level whitelisting (applies to all encryption types)
- E2EBIN sets E2EE Application-level whitelisting (applies only to On-Guard and KME encryption types)

7.1.11.1 Loading E2EBIN

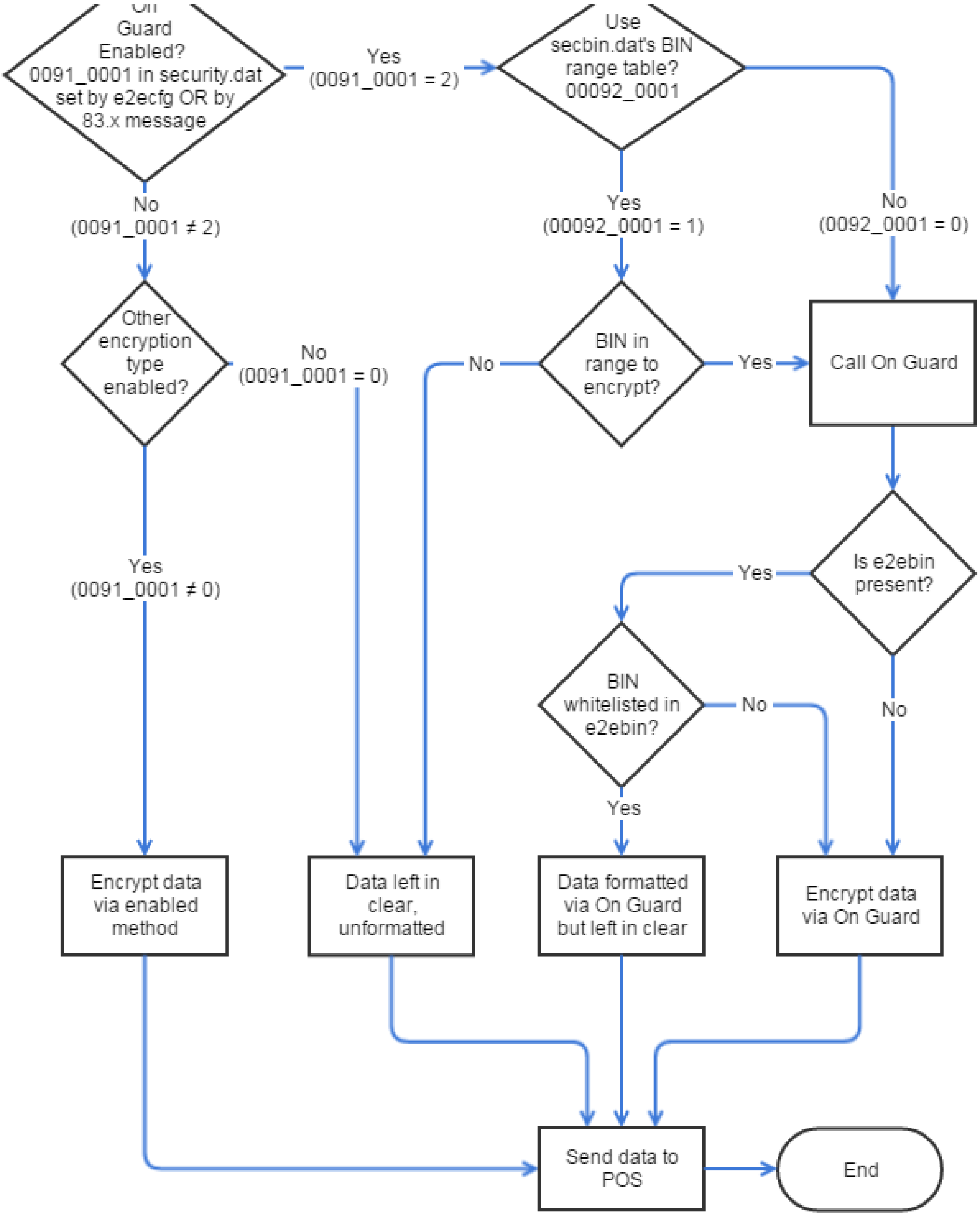
This section only applies to On Guard and KME encryption types.

If the country setting is US, the E2EBIN file must be signed before it can be uploaded. If the country setting is CA, the E2EBIN file must be MAC'ed using the terminal's CEFMK key.

7.1.11.2 Whitelist Interaction

This section applies to On-Guard and KME encryption types only.

The `SECBIN.DAT` file is always consulted first. It can specify exceptions to ranges that `E2EBIN` encrypts, but not vice versa. See the full table of considerations in the **Avoiding Vulnerabilities** section. The following flow diagram illustrates the encryption process using On-Guard as an example. KME behaves similarly in its consultation of the whitelists.


On Guard Enabled? 0091_0001 in security.dat set by e2ecfg OR by 83.x message
Yes (0091_0001 = 2)
Use secbin.dat's BIN range table? 00092_0001
No (0091_0001 ≠ 2)
Yes (00092_0001 = 1)
No (0092_0001 = 0)
Other encryption type enabled?
No (0091_0001 = 0)
No
BIN in range to encrypt?
Yes
Call On Guard
Is e2ebin present?
Yes
Yes (0091_0001 ≠ 0)
BIN whitelisted in e2ebin?
No
No
Yes
Encrypt data via enabled method
Data left in clear, unformatted
Data formatted via On Guard but left in clear
Encrypt data via On Guard
Send data to POS
End

On Guard Encryption flow

7.1.11.3 Avoiding Vulnerabilities

This section applies to On Guard and KME encryption types only.

In a few use cases, a merchant might attempt to set a range to encrypt or leave in the clear, but those settings are overlooked by the flow as described in the following table:

If a merchant wants to		But	Then
Specify cards to encrypt via...	The loaded **e2ebin**	- the card's BIN is whitelisted by secbin.dat *and/or* - On Guard/ KME is not enabled in `security.dat` (0091_0001)	The card data specified in e2ebin to encrypt is sent in the clear. The e2ebin file is not called in the flow.
	The loaded **secbin.dat**	- 0092_0001 = 0 in secbin.dat *and/or* - On Guard/ KME is not enabled in `security.dat` (0091_0001)	The card data specified in secbin.dat to encrypt is sent in the clear. The table is not called in the flow.
Whitelist cards (send card data in the clear) via...	The loaded **e2ebin**	another encryption type is enabled in `security.dat` (0091_0001)	The card data is sent encrypted by another method's specifications.
	The loaded **secbin.dat**	another encryption type is enabled in `security.dat` (0091_0001)	The card data is sent encrypted by another method's specifications.

If a merchant wants to		But	Then
		- 0092_0001 = 0 in secbin.dat *and* - e2ebin	The card is encrypted via On-Guard/KME because the whitelist is not enabled.

7.2 Encryption Methods

7.2.1 EPS (Element Payment Systems) P2PE Encryption

This section provides information to assist in the configuration and understanding of EPS (Element Payment Systems) encryption. EPS is a structure-preserving, DUKPT based encryption type.

When using EPS, the Key Sequence Number changes for every encryption. Refer to the following table for configuration parameters.

Parameters used by EPS Encryption

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption	0091_0001	0	Specify this value as 3.
Specify Encryption Key Slot (Key Index)	0091_0002	4	Valid slot values that may be used for KP4 keys include 0-5. This value must match the slot where the key was injected.
Configure Leading PAN Digits in the Clear	0091_0003	6	Configure the number of leading digits to be displayed in the clear. - Maximum = 6.
Configure Trailing PAN Digits in the Clear	0091_0004	4	Configure the number of trailing digits to be displayed in the clear. - Maximum = 4.
Masking the PAN	0091_0012	0	Specify the character to use for masking the PAN. Use one of the following for EPS encryptions: - '0' = (zero). - '*' = (asterisk).

For examples of EPS card swipes and EPS encryption refer to the following sections.

- EPS P2PE Card Swipe Examples
- EPS P2PE Encryption Processing Examples

Track 1 is not supported when performing manual entry for EPS encryption. It will remain blank for the manual entry transaction.

As of RBA release 3.3.0, EPS is PCI3-compliant and uses only KP4 keys.

Encryption or masking of cards with PANs containing less than 9 digits is not supported. Merchants should either whitelist these cards or disable non-standard card encryption.

Default values for DFS Data Index '0091_0002' (Specify Encryption Key Slot) differ for KP4 versus non-KP4 keys.The following table shows the default values and valid ranges for KP4 and non-KP4 encryption key slots used for EPS encryption.

Encryption Key Slots for EPS Encryption

	KP4 Keys	Non-KP4 Keys
Default Value	4	6
Range	0 - 5	0 - 9

7.2.1.1 EPS P2PE Card Swipe Examples

With EPS encryption, the first 6 and the last 4 digits are always in the clear, and the remaining middle digits are encrypted. Remember that the Key Sequence Number changes for every encryption when using EPS. The following table illustrates swiped card data examples viewed using a variety of messages when encrypted with EPS.

EPS Encrypted Message Samples

Message	Description of After	Examples of Before (original card number) and After Encryption
19.x (BIN Lookup)	Only 10 of 13 digits are displayed. The first 6 digits and the last 4 digits are in the clear; the 3 middle digits are encrypted.	Before: 4012345678909 After: 4012340008909 19.D11000034012340008909[FS]4B99358C793844C31AC522C764CC5A676 C3DFEE935D522CB613051F2554A9D3B87C09BE4E1A55896E44AB21F4FDA82 8D248F3AE1D1025F3AC935CDB33D1A1AD1:FFFF9876543210E00004[FS]1164 C984EBF0C3FED6A2047073608535C68A1BA050DDB73AAFA03DCC276CAB1 5:FFFF9876543210E00004
23.x (Card Read Request Response)	Only 10 of 13 digits are displayed. The first 6 digits and the last 4 digits are in the clear; the 3 middle digits are encrypted.	Before: 4012345678909 After: 4012340008909 23.04B99358C793844C31AC522C764CC5A676C3DFEE935D522CB613051F25 54A9D3B87C09BE4E1A55896E44AB21F4FDA828D248F3AE1D1025F3AC935C DB33D1A1AD1:FFFF9876543210E00004[FS]1164C984EBF0C3FED6A20470736 08535C68A1BA050DDB73AAFA03DCC276CAB15:FFFF9876543210E00004
29.x (Get Variable Request/ Response)	All 13 digits are displayed. The first 6 digits and the last 4 digits are in the clear; the middle 3 digits are encrypted.	Before: 4012345678909 After: 4012340008909 Track1: 29.200004064B99358C793844C31AC522C764CC5A676C3DFEE935D522CB613 051F2554A9D3B87C09BE4E1A55896E44AB21F4FDA828D248F3AE1D1025F3 AC935CDB33D1A1AD1:FFFF9876543210E00004 Track2: 29.200004071164C984EBF0C3FED6A2047073608535C68A1BA050DDB73AA FA03DCC276CAB15:FFFF9876543210E00004

Message	Description of After	Examples of Before (original card number) and After Encryption
50.x (Authorization Request/Response)	Only 10 of 13 digits are displayed. The first 6 digits and the last 4 digits are in the clear; the 3 middle digits are encrypted.	Before: 4012345678909 After: 4012340008909 Track2: 50.123456789012345678901234567890123456789002070032548000002@D1 164C984EBF0C3FED6A2047073608535C68A1BA050DDB73AAFA03DCC276CA B15:FFFF9876543210E00004[FS]1@[FS]1025[FS]

7.2.1.2 EPS P2PE Encryption Processing Examples

The following table shows EPS encrypted examples of Track 1, Track 2, and manually entered data.

EPS Encrypted Data Samples

Track	Type	Example
Track 1	Original	B4447340101127648^VISA CARDHOLDER/^1209121000000678000000
	Encrypted Track 1	5D97BDCBC8F9E12F4C99D502FB9B30E7715DBC0C8D64C27BE868554F24F176733C3798B96 76A7D68A0F3BFF8256E484AB65A88B4B2C4AB50DFA60B09585B9D57
	KSN	A08B000C000003000023
	Encrypted Track Format	EncryptedTrack1Data:KSN1
Track 2	Original	4447340101127648=1209121000067800000
	Encrypted Track 2	CEAFC2FD0BA3E76E0C062DD2DD4196E111A71424C52561E943142AD271FC0D86C45CC365 B8D7E292
	KSN	A08B000C000003000024
	Encrypted Track Format	EncryptedTrack2Data:KSN2

Track	Type	Example
Track 3		Used with the 23.x message, Track 3 data will be sent in the clear and is only available when the '0003_0010' (Append Track 3) parameter is set to a value of '1' (where 1 = Send Track 1, Track 2 and Track 3).
Manually Entered	Original Track 1 Data	%M4744750029324780^MANUAL ENTRY/ ^1306000000123000000?
	Encrypted Track 1 Data	Track 1 is not supported when performing manual entry for EPS encryption. It will remain blank for the manual entry transaction.
	Original Track 2 Data	4744750029324780=1306=123 The '123' within the Track 2 data in this example represents the CVV number.
	Encrypted Track 2 Data	5EFDACAD0F0B6997EC120D2AE568EDD504A50 214F8871E6C43D3A4CFDC30D4BC:FFFF 9876543210E00003
	KSN	FFFF9876543210E00003
	Encrypted Track Format	EncryptedManuallyEnteredData:KSN Manually entered data will appear as Track 2 data.

12/26/17 WJM: Added material below - it was previously in "Encryption Processing," but didn't belong there because it is specific to EPS. It may not be correct.

The following table shows EPS encrypted examples of Track 1, Track 2, and manually entered data:

Encrypting Sample Data with EPS

Track	Type	Example
Track 1	Original	`%B4447340101127648^VISACARDHOLDER/^1209121000000678000000?`

	Encrypted Track 1	`5D97BDCBC8F9E12F4C99D502FB9B30E7715DBC0C8D64C27BE868554F24F176733C37`
		`98B9676A7D68A0F3BFF8256E484AB65A88B4B2C4AB50DFA60B09585B9D57`

	KSN	`A08B000C000003000023`
Track 2	Original	`4447340101127648=1209121000006780000 0?`
	Encrypted Track 2	`CEAFC2FD0BA3E76E0C062DD2DD4196E111A71424C52561E943142AD271FC0D86C45C` `C365B8D7E292`
	KSN	`A08B000C000003000024`
Manually Entered	Original Track 1 Data	%M4744750029324780^MANUAL ENTRY/^1306000000123000000?
	Encrypted Track 1 Data	Track 1 is not supported when performing manual entry for EPS encryption. It will remain blank for the manual entry transaction.
	Original Track 2 Data	4744750029324780=1306=123 "123" within the Track 2 data in this example represents the CVV.

	Encrypted Track 2 Data	`5EFDACAD0F0B6997EC120D2AE568EDD504A50214F8871E6C43D3A4CFDC30D4BC:`
		`FFFF9876543210E00003`

Track	Type	Example
	KSN	`FFFF9876543210E00003`

7.2.2 Generic TDES DUKPT Encryption

Generic TDES DUKPT encryption (Triple Data Encryption Algorithm with Derived Unique Key Per Transaction) is an encryption method. A unique key is used for each transaction. Generic TDES DUKPT encryption follows DUKPT: 2009 key management defined in X9.24-1 for data keys and uses CBC mode encryption with an Initial Vector of nulls. The current Telium implementation can be used with MSR, contactless, and manually entered data. To enable this feature and set its parameters, edit the `SECURITY.DAT` section of `CONFIG.DFS`. The resulting `SECURITY.DAT` file must be signed and downloaded to the terminal to enable the encryption.

When using generic TDES DUKPT encryption, there are two options for incrementing the key serial number (KSN). It can either be forced to increment, or it will automatically increment after 10 encryptions. RBA uses the automatic advance mode. Additional information can be found in the following subsections:

TDES DUKPT Configuration - Configuration information for Generic TDES DUKPT.

Usage - Data format prior to encryption, data returned to the POS application, and determining the encryption configuration.

7.2.2.1 TDES DUKPT Configuration

7.2.2.1.1 Configuring Security Parameters

Configuration information for Generic TDES DUKPT is contained in two files: `SECURITY.DAT` and `SECBIN.DAT`. These files are used to configure encryption and security in the application. Refer to Configuring the Application in this guide for details. Security parameters for generic TDES DUKPT encryption are described in the following table:

Parameters used by TDES DUKPT Encryption

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption (in `security.dat`)	0091_0001	0	Specify this value as 11 for Generic TDES DUKPT.
Specify Encryption Key Slot (Key Index) (in `security.dat`)	0091_0002	4	Generic TDES DUKPT uses this DUKPT key slot for this feature. (Only slots 0-5 can be used).
Configure Leading PAN Digits in the Clear (in `security.dat`)	0091_0003	6	Generic TDES DUKPT ignores the value of this parameter. Specifies the number of leading digits to be displayed in the clear (Maximum = 6). The default value of 6 is hard-coded for Generic TDES DUKPT.

Parameter Name	DFS Data Index	Default Value	Description
Configure Trailing PAN Digits in the Clear (in `security.dat`)	0091_0004	4	Generic TDES DUKPT ignores the value of this parameter. Specifies the number of trailing digits to be displayed in the clear (Maximum = 4). The default value of 4 is hard-coded for Generic TDES DUKPT.
Masking the PAN (in `security.dat`)	0091_0012	0	Generic TDES DUKPT ignores the value of this parameter. Specifies the character to use for masking the PAN. The default value of 0 (zero) is hard-coded for Generic TDES DUKPT.

Encrypting or masking cards with PANs containing fewer than nine digits is not supported. Either whitelist these cards or disable non-standard card encryption.

7.2.2.1.2 Configuring for Manual Entry

In addition to configuring the security parameters, the "Enter Card" prompt display parameter ('0007_0029') must also be configured for the proper cardholder prompt. When using TDES encryption and manually entering data, the PAN, expiration date and CVV are all required. This parameter must therefore be set to '0' or '1' when using this encryption mode.

The only files used by RBA are `SECURITY.DAT` and `SECBIN.DAT`. These files must be signed by Ingenico and downloaded to the terminal, which prevents an attacker from turning off encryption or changing the settings.

7.2.2.2 Usage

For information on Generic TDES DUKPT encryption data format, communications with the POS, and configuring the encryption, refer to the following subsections:

Data Format Prior to Encryption - Generic TDES DUKPT encryption cases with examples.

Data Returned to the POS Application - Description of set RBA properties which are set once the card is swiped.

Determining the Encryption Configuration - Methods used by the application to determine how the terminal is configured.

7.2.2.2.1 Data Format Prior to Encryption

The input to the encryption process depends on the encryption type. For Generic TDES DUKPT encryption, there are four cases:

- Only Track 1 was read successfully. The string to be encrypted consists of the raw Track 1 data with Start and End Sentinels.
    - Example:
      %B4445222299990007^LAST/VISA^141250254321987 12345Q?
- Only Track 2 was read successfully. The string to be encrypted consists of the raw Track 2 data with Start and End Sentinels.
    - Example:
      ;4445222299990007=141250254321987 12345?
- Data was entered manually. The string to be encrypted consists of concatenated dummy Track 1 and Track 2 data with Start and End Sentinels. The dummy tracks are constructed from the manually-entered PAN, expiration date and CVV2 which are all required when using TDES encryption in manual entry mode.
    - Example:
      %M5444009999222205^MANUALLY/ENTERED^121200000012340000 00?;
      5444009999222205=121200000012340 00?
      In this example, 5444009999222205 is the PAN, 1212 is the expiration date (YYMM), and 1234 is the CVV2. There will always be six 0's between the expiration date and the CVV2. There will always be six 0's after the CVV in Track 1, and three 0's after the CVV in Track 2.
- Both Tracks 1 and 2 were read successfully. The string to be encrypted consists of the concatenated raw Track 1 and Track 2 with Start and End Sentinels.
    - Example:
      %B4445222299990007^LAST/VISA^141250254321987 12345Q?;
      4445222299990007=141250254321987 12345?

> Please refer to Manual Card Data Entry in E2EE Mode for information on programming manual entry of cardholder data when using point-to-point encryption.

7.2.2.2.2 Data Returned to the POS Application

Once a card has been swiped or tapped, or the manual entry process is complete, the following properties are set:

TDES Properties and their Content

Property	Contents when using Generic TDES DUKPT Encryption
Account Number	Masked account number (middle digits are 0)
ExpirationDate	Expiration date (in the clear)
FirstName	First name (in the clear)
MiddleInitial	Middle initial (in the clear)
ServiceCode	Service code (in the clear)

Property	Contents when using Generic TDES DUKPT Encryption
Suffix	Suffix (in the clear)
Surname	Surname (in the clear)
Title	Title (in the clear)
Track1Data	Masked Track 1 data
Track1DiscretionaryData	Masked Track 1 discretionary data
Track2Data	Masked Track 2 data
Track2DiscretionaryData	Masked Track 2 discretionary data
Track3Data	The Track 3 data sent to the POS consists of four items separated by colons (":"): • The KSN of the TDES DUKPT encryption key - 20 bytes ASCII hex characters. • One digit indicating which data were encrypted: 1 = Track 1, 2 = Track 2, 3 = dummy tracks for manually-entered data, 4 = Track 1 and Track 2. • The four digit length (decimal) of the encrypted data block. This is the number of bytes of binary data. • The encrypted data block in ASCII Hex format. Since each byte is represented by two ASCII characters, the length of this string will be twice the length of the binary data block. The following is an example of Track 3 Data for generic TDES DUKPT encryption, where Track 2 data was encrypted: FFFF4900361491E00004:2:0048:16D8BD06F00671AAA4FBA2381EDD239DE03E618FB33 2AEA7524CBB1ED1DBE4FFDEF26740138D5549E08FB7ECD1649169

If the `TransmitSentinels` property is true, then Track1Data, Track2Data and Track3Data will each begin with a start sentinel and end with an end sentinel.

Currently, Track 3 data either in the clear, or masked, are not available to the application.

7.2.2.2.3 Determining the Encryption Configuration

The encryption settings are configured on the terminal and cannot be changed by the application. The application can find out how the terminal is configured by querying certain RBA variables corresponding to the configuration parameters. The general rule is that the variable DFS_xxxx_yyyy contains the value of the parameter with DFS Data Index xxxx_yyyy. Only the variables needed by the POS application are provided. Currently, the following variables are supported and are listed in the below table.

Enabling TDES DUKPT Encryption

Variable Name	Variable Description
DFS_0091_0001	A read-only variable containing the encryption type: • 0 = No encryption • 11 = Generic TDES DUKPT encryption.

7.2.3 Monetra CardShield Encryption

7.2.3.1 Introduction

Monetra CardShield encryption is built with design principles similar to DUKPT key management and TDES ciphers. Monetra encryption may be used in both the standard RBA flow, and with the 23.x message.

Configuration Parameters (in config.dfs)

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption	0091_0001	0	Specify this value as '7'.
Specify Encryption Key Slot (Key Index)	0091_0002	4	Valid values include '0' – '6'. • Only slots '0' - '5' can be used. • This value must match the slot where the key was injected.
Configure Leading PAN Digits in the Clear	0091_0003	6	Configure the number of leading digits to be displayed in the clear. Maximum = '6'.
Configure Trailing PAN Digits in the Clear	0091_0004	4	Configure the number of trailing digits to be displayed in the clear. Maximum = '4'.

Parameter Name	DFS Data Index	Default Value	Description
Masking the PAN	0091_0012	0	Specify the character to use for masking the PAN. Monetra uses ONLY the '*' (asterisk). By always using '*', the POS will have a reliable way to identify whether the data returned in the 23.x message is encrypted or not.

> Encryption or masking of cards with PANs containing less than 9 digits is not supported. Merchants should either whitelist these cards or disable non-standard card encryption.

7.2.3.2 Data to be Encrypted if Track 3 is Available

If present, the original Track 3 data will be encrypted. The other track data will be handled as specified by the above parameters, and as explained below.

7.2.3.3 Data to be Encrypted if Track 3 is Unavailable

If Track 3 is unavailable, Track 1 and Track 2 are encrypted together as a single block of data. Dummy card data is not created for empty track data in the card, and no sentinels will be sent in that case. If a track is blank, then that track will be returned to the POS as blank. Only track data which is present will be encrypted. Depending on which tracks are present, the format of the data fed to the ciphering algorithm will be in one of the following formats:

Encrypting per Valid Track Data

Tracks Present and Valid	Algorithm
1 & 2	%BTRACK1?;TRACK2?
1 only	%BTRACK1?;?
2 only	%B?;TRACK2?

7.2.3.4 Manual Entry

For manual entry, the data input to the ciphering algorithm will be in the form

PAN|EXPDATE|CVV2

If EXPDATE or CVV2 are not entered manually, they will be empty. Regardless, the two vertical bar characters in the data will remain. The decrypted data may contain trailing null characters as required for padding.

7.2.3.5 *Encryption Does Not Occur If…*

- If neither Track 1 nor Track 2 has been read successfully (e.g., if there is an encryption error, or an invalid card exists), then encryption does not occur; the data is then sent in the clear.
- If one of the tracks is present and valid, then encryption will occur with only the present and valid track.
- If Track 3 has been read successfully, then encryption does not occur, and again, the data is sent in the clear. This is because Track 3 is used to send back the encrypted data and the KSN.

7.2.3.6 *Encryption Data Returned to the POS – Tracks 1 & 2*

The encrypted data will always be sent to the POS using the 23.x message. Sentinels are included in the 23.x message.

The data received can be identified as encrypted when it has these two characteristics:

1. Track 1 or Track 2 contains one or more instances of the masking character '*' (asterisk).
2. Track 3 contains a ':' (colon) which is used to separate the encrypted data from the KSN.

> Either both characteristics or neither characteristic will exist at any one time.

Depending on which tracks were encrypted, you will receive one of the following three message formats with the 23.x message:

Encrypted Tracks Returned

Tracks Present and Valid	Format
1 & 2	23.0%BMASKEDTRACK1?[FS];MASKEDTRACK2?[FS]ENCRYPTEDDATA:KSN
1 only	23.0%BMASKEDTRACK1?[FS];?[FS]ENCRYPTEDDATA:KSN
2 only	23.0%B?[FS];MASKEDTRACK2?[FS]ENCRYPTEDDATA:KSN

7.2.3.7 *Configuring for Manual Entry using the 23.x Message*

If selected, manual entry will follow the same rules as manual entry from the swipe screen.

To load either of the manual entry forms, specify the `CCOD.K3Z` form for contactless terminals or the `COD.K3Z` form for non-contactless terminals in the 23.x message, in addition to the prompt or prompt index. The button for manual entry is hidden when the Display "Enter Card" Prompt (configuration parameter '0007_0029') is set to '0'. In order for the manual entry button to be visible, this configuration parameter must be set to a value of '1' to '4' as described in the following table.

Configuring for Manual Entry

'0007_0029'	"Enter Card" Button	Enter Card Number	Enter Expiration Date	Enter CVV
0	Not Displayed			
1	Displayed	Yes	Yes	Yes

'0007_0029'	"Enter Card" Button	Enter Card Number	Enter Expiration Date	Enter CVV
2	Displayed	Yes	Yes	No
3	Displayed	Yes	No	Yes
4	Displayed	Yes	No	No

If no manual entry configuration is necessary, load either of the default forms, issue the 23.x message as normal (e.g., no form name specified, but the prompt or prompt index still needs to be sent).

7.2.3.8 Initiating an Encrypted Swipe/Tap/Manual Entry

Monetra encryption support has been added for card swipes during the standard flow "swipe" screen. Previously, only the 23.x on-demand card swipes were allowed when using Monetra encryption, while card swipes from the standard flow "swipe" screen were flagged as errors which caused the terminal to go offline. With card swipes now being supported during the standard flow "swipe" screen, this action is no longer being processed as an error.

7.2.3.9 Monetra CardShield Encryption Examples

7.2.3.9.1 Monetra Data Encryption Examples

Content	PAN	Example Value After Encryption
PAN Only	7195388662093300010	**Track 1**: 7195388662093300010 **Track 2**: 7195388662093300010
Track 1 Only	6011086910623514	**Track 1**: %B601108*****3514^INGENICO/ TEST CARD ^0705101************? **Track 2**: (BLANK) **Track 3**: 212A38115DC56F04850BEB5E91014F401A 277713FAB73BE3C66C4893 9BF7289A3CC4154DD23A145D00CD0358 52CFCFA97EECE48FF9658F8F47F27A A0CED16B4C:0000000200000B00000161
6 Digit PAN	475767	**Track 1**: 475767 **Track 2**: 475767
Track 2 Only	21110000075272	**Track 1**: (BLANK) **Track 2**: 21110000075272
Track 2 Only	6001760817150245351	**Track 1**: (BLANK) **Track 2**: 6001760817150242351=3712

7.2.3.9.2 Manual Entry Transaction

Variable IDs `399` and `402` (Account Name) return "Manual Entry" for manual entry when Monetra encryption is enabled.

manualAccountName is replaced with 'msg23MsrName' in a 23.x message during an On Demand flow manual entry. Variable `399` then returns "Manual Entry" in the 29.x message as shown in the table below.

Monetra Manual Entry in On-Demand Flow Example

<table>
<tr><th>Step</th><th>Notes</th></tr>
<tr><td>Enable TransArmor encryption.</td><td></td></tr>
<tr><td>Enable '0007_0029'.</td><td>Set to any value '1' through '4'.</td></tr>
<tr><td>Send 23.x message while terminal displays a "Please slide card" form.</td><td></td></tr>
<tr><td>Press ENTER CARD button.</td><td></td></tr>
<tr><td>Enter PAN+Expiry Date+CVV value as per the '0007_0029' settings.</td><td></td></tr>
<tr><td>The terminal displays "card accepted" form and sends a 23.x response.</td><td></td></tr>
<tr><td>The POS prompts the terminal for variables with 29.x messages.
During a card swipe or contactless transaction in a normal transaction flow (not On-Demand), the '29.00000399' request would still return a card name in the response, e.g., '29.20000399TESTCARD/TEST'.</td><td><b>Sample 29.x Requests and Responses</b>
<table>
<tr><th>29.x Request</th><th>29.x Response</th></tr>
<tr><td>29.00000398</td><td>29.200003984445220000000007</td></tr>
<tr><td>29.00000399</td><td>29.60000399Manual Entry</td></tr>
<tr><td>29.00000400</td><td>29.200004000000</td></tr>
</table></td></tr>
</table>

7.2.4 On-Guard Encryption

7.2.4.1 Overview

This section describes support for On-Guard point-to-point encryption. On-Guard encryption uses an injected DUKPT key. On-Guard encryption and KME encryption methods are handled similarly and are referred to collectively as the *E2EE feature* (End-to-End Encryption). On-Guard and KME encryption methods can process card data read by:

- Magnetic stripe reader (MRS)
- Smartcard reader (SCR) for EMV cards
- Manual entry

The E2EE solution isolates the POS system (electronic cash register or host device) from processing clear-text card data, and reduces the impact of PCI DSS reviews.

On-Guard encryption and KME encryption can be enabled, disabled, and configured by a signed configuration file loaded into the SYSTEM drive on the terminal.

- The text file containing the configuration information is named "e2ecfg".
- The corresponding signed configuration file is "829651xxxx.PGN" where xxxx is the version number.

This configuration file contains the *enable* mode (KME or On-Guard) as well as the index of the encryption key in the secret area. After the application reads and parses this file, it will be deleted. The configuration extracted from this fileis saved to the application disk. The E2EE configuration information is incorporated into the existing security.dat file described in the Security Parameters (security.dat) section of this document.

An E2EE activate command simplifies the use of this feature. The terminal can be loaded with the required software and keys. When the POS and network are ready to process the encrypted card data, the POS can send a single command to enable the feature.

Note: After E2EE encryption is enabled by either configuration file or by command, it cannot be disabled. The application has a mechanism to prevent the reverse operation. The only way to disable E2EE encryption is to erase the terminal and reload all components.

By default, E2EE encrypts data from all cards. If you require only some cards to be encrypted, you must configure the *e2ebin* file for the appropriate BIN ranges. This file contains a list of BIN ranges, utilizing low and high ranges of the first 6 digits of the PAN. If there is a match between the first 6 PAN digits of the card data read and the BIN table, then the card data read is returned in the clear. The *e2ebin* file must be signed before it can be loaded onto a terminal. As an expected change to the general-release version, the BIN ranges are defined in the existing *secbin.dat* file.

Refer to the following sections for more information about On-Guard encryption:

- On-Guard Configuration
- On-Guard Card Data Encryption Rules
- Handling Existing RBA Messages
- E2EE Card Data Encryption

7.2.4.2 On-Guard Configuration

7.2.4.2.1 On-Guard Encryption Configuration

As an expected change to the general-release version, the E2EE configuration information may be incorporated into the existing “security.dat” file described in the Security Parameters (security.dat) section. Accordingly, the configuration file described in this section may no longer be used. The E2EE configuration file “e2ecfg” must be signed as (829651xxxx.PGN) before it can be used by the RBA. The “e2ecfg” file is a text file whose first line contains the following values, separated by commas:

- W
- X
- Y
- Z
- A

The function of these variables is described in the following table.

On-Guard Encryption Variables

Item	Length	Type	Description
W	1	Alphanumeric	E2EE mode. • 1 = KME E2EE mode is enabled. • 2 = On-Guard E2EE mode is enabled. • D = E2EE is disabled.
X	1	Alphabetic	Output format type. • A = Type A formatting, will return Track 2 only and support Base 24 framing. • B = Type B formatting used by On-Guard, will return track 2 only (with no framing) and the KSN for the E2EE DUKPT cryptogram.
Y	1	Alphabetic	Type of key used. • M = Master/Session key (used by KME). • D = DUKPT (used by On-Guard encryption).
Z	1	Numeric	Key slot where the encryption key has been injected. Key Pattern 4 (KP4) must be used. • Must be in the range from 0 to 5. Commonly, • Slot ‘2’ is used for KME keys. • Slot ‘5’ is used for On-Guard keys.
A	1	Numeric	Optional. Specifies the key number of the optional TDES local storage data encryption key. • Value for key number is from 0 to 9. This is not used for On-Guard encryption. The format of the LS data block is always that of the manual entry definition.

Encryption or masking of cards with PANs containing less than 9 digits is not supported. Merchants should either whitelist these cards or disable non-standard card encryption.

Because signed files have a minimum size, padding is added after the above information to meet those size requirements. The following example shows this padding:

```
1, A, M, 2,
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
```

0091_0001 must be set to '2' in order to use On-Guard encryption.

The E2EE configuration file can be signed and downloaded, changing the E2EE feature from being disabled to enabled. In this case, the E2EE feature will be activated and the RBA application will encrypt all data that is required to be encrypted. The feature will remember being enabled such that downloading a configuration file to disable E2EE will be ignored.

If the E2EE configuration file is loaded into the terminal with the E2EE mode parameter set to 'enabled' ('1' or '2'), then the `config.dfs` parameter '0091_0001' (Encrypt track data) will have no effect even if it has been enabled. Its value will be reset to '0'. Note that if the RBA starts and the E2EE mode has not been configured either by file or by command, then the RBA displays the message "No Config file" and will not run. (By default, the RBA installation package includes a version of the configuration file that disables E2EE, so this situation will not typically occur.)

Some terminals require E2EE to be enabled. In these cases, if the E2EE mode is set to Disabled, the RBA will display the message "E2EE Not Enabled" and will not run.

7.2.4.2.2 On-Guard BIN Table Configuration

The BIN table "E2EBIN" provides a means to specify that cards in certain BIN ranges will not be encrypted. This file must be signed and downloaded to the terminal in order to be recognized by RBA. As an expected change to the general-release version, the BIN range information will be incorporated into the existing `secbin.dat` file described in the RBA documentation, and the configuration file described in this section will no longer be used. The unsigned BIN table is a text file which contains lines in the following format:

Unsigned BIN Table Content

Line	Content
1	Low bin (6 digits)
2	"-"
3	High bin (6 digits)
4	";"
5	Service code

In the service code, an “x” can be used to match any digit. A service code of “mmm” is applicable to BIN checking of Manual Entry card data only. Consider the following sample BIN table:

Service Codes with x and mmm Examples

Sample	Content
1	000000-999999;110
2	000000-299999;x6x
3	130000-299999;xxx
4	800000-999999;x20
5	130000-299999;mmm

The service codes 120 and 220 are designated for Debit Card only, so they will be excluded from wildcard service codes such as “xxx” or “x2x”, etc. If a service code of "120" or "220" is explicitly entered in the BIN table, then matching entries will not be encrypted.

The BIN file must be signed and loaded onto the HOST drive in the terminal. The RBA application, at terminal boot up, parses this file and then deletes it.

When determining the PAN for use with On-Guard encryption, the RBA follows these rules:

- For MSR and manual entry, look for an “=” sign or the end of the card data to terminate the PAN, up to a maximum of 37 characters.
- For card data from smart cards (contact or contactless), it is assumed that the PAN will be 19 or fewer digits.

7.2.4.3 On-Guard Card Data Encryption Rules

This section describes how data will be formatted prior to E2EE encryption.

7.2.4.3.1 Type A Formatting

Type A formatting applies to KME and will be described in a separate document.

7.2.4.3.2 Type B Formatting

Type B formatting applies to On Guard encryption. This section describes how card data will be formatted for On Guard encryption. Note that any data processed by On Guard (even if whitelisted in `E2EBIN`) will be formatted, whether returned in clear or encrypted. PANs whitelisted by an enabled `secbin.dat` will not be formatted or encrypted.

A 'Start Sentinel' character and 'End Sentinel' character will be present in all cases. For manual entry, the 'Start Sentinel' character is “M”. For all other cases, the 'Start Sentinel' character is “;”. The End Sentinel is always “?”. The following applies to type B formatting

Type B formatting (On Guard only)

Data/source	Data to be encrypted, if any	Additional formatting performed
Track 2 data read from a swiped or MSD card	Track 2 data converted into ASCII	None
Chip card data (contact or contactless) with Track 2 Equivalent Data (EMV tag `T57` or equivalent) available	ASCII-converted data	After encryption, a separator (‘=’ , hex 0x3D) is added after the PAN if not already present.
Chip card data (contact or contactless) with Track 2 Equivalent Data (EMV tag `T57` or equivalent) **not** available	• PAN • Application Expiration (YYMM), from tag T5F24 • Service Code, from tag T5F30	After encryption, a separator (“=”) is added after the PAN, followed by the encrypted Application Expiration date, followed by the encrypted Service Code.
Manually captured cardholder data	PAN followed by the expiration date (YYMM format) in ASCII	After encryption, the ASCII character ‘M’ will be added to the beginning of the buffer, and a separator (“=”) will be added between the PAN and the expiry date. The application responsible for decrypting this data will have to parse the data into its component encrypted PAN and Expiry Date before handing off to the Decryption Appliance for decryption.

7.2.4.4 Handling Existing RBA Messages

7.2.4.4.1 Message Handling with E2EE Encryption Enabled

With E2EE enabled, a different set of messages is used in place of existing RBA messages for certain functionality. As an expected change to the general release, messages are more consistent between E2EE and non-E2EE configurations. The following table describes the alternate message process that occurs when E2EE is enabled.

Alternate Process for On-Guard E2EE

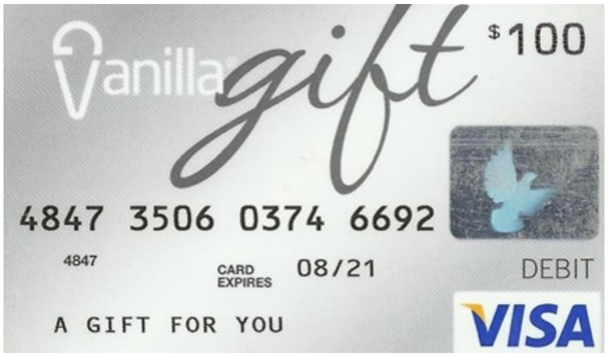
Message	When E2EE Is Enabled
12.x Account Message	Not supported.
18.x Non-Payment Card Message	The 85.x On-Guard and KME Non-Payment Card Message is sent in place of this message.
19.x BIN Lookup Message	The 86.x On-Guard and KME BIN Lookup (PIN Encouragement) Message is sent instead. A 19.x message from the POS is ignored.
23.x Card Read Request (On-Demand)	The 87.x On-Guard and KME Card Read Data is sent instead. The 23.x card read request message is disabled and returns an invalid command response with the error code 9 (declined).

7.2.4.4.2 31.x PIN Entry Command Message in E2EE Mode

On-Guard encryption does not use a masked PAN as a workaround to enable On-Demand PIN. The terminal creates a masked PAN and saves it in variable 398 sothe POS and terminal can identify the account number the PIN is created with when the PAN cannot be compromised in the clear. Thus, the 31.x message and variable 398 use the masked PAN.

7.2.4.5 MSR Encryption Example

In the following example, an MSR card is successfully read and card data is encrypted using On-Guard encryption. Refer to the following image for the card used in this example.

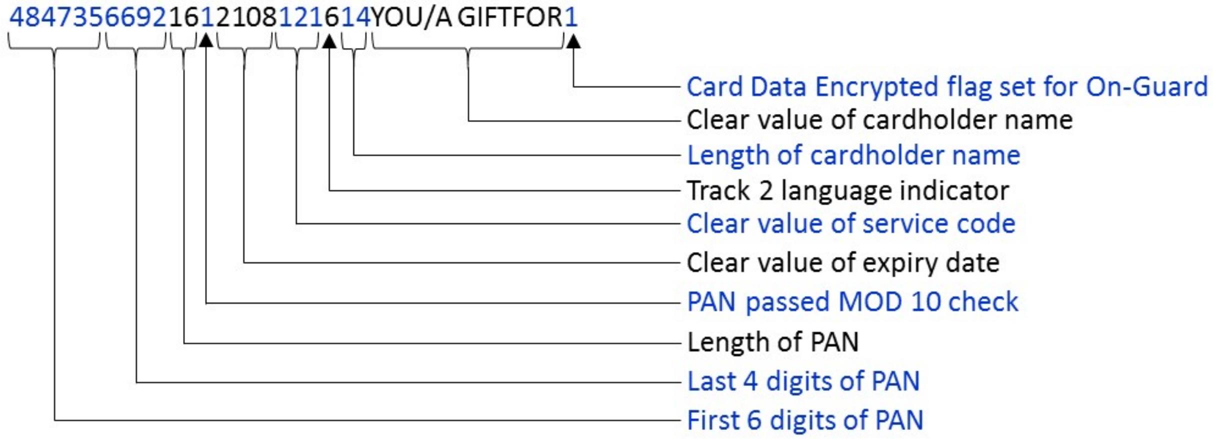


Example MSR Card

The following data will be returned in the 87.x On-Guard and KME Card Read Data:

- Exit type = '0' indicating a good card read.
- Card Data = '48473566921612108121614YOU/A GIFTFOR1
  BFFFF98765432292000050114328438595666498708=00823303648413132DE47
  E75CB39344F75AA4B82BDFF0AA2264683140DF3D6C473E2C74A7BE9B8B
  59D5A11512B5361AB781A382B768E69BE9C02'

FOR MSR and contactless MSD, the data in this example is formatted as follows:


48473566921612108121614YOU/A GIFTFOR1
Card Data Encrypted flag set for On-Guard
Clear value of cardholder name
Length of cardholder name
Track 2 language indicator
Clear value of service code
Clear value of expiry date
PAN passed MOD 10 check
Length of PAN
Last 4 digits of PAN
First 6 digits of PAN

MSR and Contactless MSD Data Format

With On-Guard encryption enabled, the following data is formatted as described in the following table for Data Sent with On-Guard Encryption Enabled.

BFFFF98765432292000050114328438595666498708=00823303648413132DE47E75CB39344

F75AA4B82BDFF0AA2264683140DF3D6C473E2C74A7BE9B8B59D5A11512B5361AB781A

382B768E69BE9C02

Data Sent with On-Guard Encryption Enabled

Data	Value
Specifies IngeCrypt data format	B
DUKPT Key Serial Number (KSN)	FFFF9876543229200005
Reserved; always the same value	0114
Decimal length of IC encrypted card data	32
IC encrypted card data if reading is valid	8438595666498708=008233036484131
Decimal length of AES encrypted PAN field	32
Decimal length of AES encrypted PAN field	DE47E75CB39344F75AA4B82BDFF0AA22
Decimal length of AES/TDES encrypted card data field	64
AES/TDES encrypted card data field	683140DF3D6C473E2C74A7BE9B8B59D5A11512B5361AB781A382B768E69BE9C0
First digit of Track-2 card language indicator	2

7.2.4.6 E2EE Card Data Encryption

If E2EE is enabled, the application encrypts the card data, and sends it to the host for decryption and processing. The format of the Track-2 field in the 50.x Authorization Request message is changed: encrypted data replaces the MSR data typically incorporated at offset 62 in the 50.x message. The encrypted card data varies, depending on the card-entry mode. Refer to the following sections for more information:

- MSR and Contactless MSD in E2EE Mode
- Manual Card Data Entry in E2EE Mode
- EMV Contact and Contactless in E2EE Mode

7.2.4.6.1 MSR and Contactless MSD in E2EE Mode

For MSR and contactless MSD transactions, data is formatted as follows:

MSR and Contactless MSD Format

Data	Length	Type	Description
PAN, first six digits	6	ASCII	Clear value of the first six digits of the PAN
PAN, last four digits	4	ASCII	Clear value of the last four digits of the PAN
PAN length	2	ASCII	Decimal length of the PAN
PAN Mod-10 check flag	1	ASCII	• 0 = PAN failed MOD 10 check • 1 = PAN passed MOD 10 check
Expiry date (see Note 1)	4	ASCII	Clear value of the Track-2 expiry date (*YYMM*)
Service code (see Note 1)	3	ASCII	Clear value of the Track-2 service code
Language code (see Note 1)	1	ASCII	Track-2 card language indicator
Cardholder name length	2	ASCII	Decimal length of the cardholder name
Cardholder name	n	ASCII	Clear value of the cardholder name
Card Data Encrypted flag (see Note 2)	1	ASCII	• 0 = Clear ASCII data. Only occurs if whitelisted by `E2EBIN` • 1 = Encrypted ASCII data

Note 1
When Track 1 and Track 2 are requested with E2EE enabled, Service code, Language code, and Expiry Date are filled from Track-2 data only. If only Track-1 data is available, then these fields contain only 0 values.

Note 2
When Track 1 and Track 2 or only Track 1 are requested with E2EE enabled, and only Track 1 is available, then the Card Data Encrypted flag is set, but the encrypted field lengths are zero.

The remaining fields depend on the value of the Card Data Encrypted flag.

Fields with Card Data Encrypted Flag set to 0

Data	Length	Type	Description
Track 1 length	2	ASCII	Decimal length of ISO 1 field.
ISO 1 field	n	ASCII	ISO 1 track if reading is valid.
Track 2 length	2	ASCII	Decimal length of ISO 2 field.
ISO 2 field	n	ASCII	ISO 2 track if reading is valid.
Extended Language code	1	ASCII	First digit of Track 2 card language indicator.

Fields with Card Data Encrypted Flag set to 1

Data	Length	Type	Description
Encrypted Format Type	1	ASCII	• B = IngeCrypt (IC) data format
IC KSN	20	ASCII	DUKPT Key Serial Number (KSN)
Reserved	4	ASCII	0114
IC card data length	2	ASCII	Decimal length of IC-encrypted card data
IC card data field	n	ASCII	IC-encrypted card data if reading is valid
AES PAN length	2	ASCII	Decimal length of AES-encrypted PAN field
AES PAN field	n	ASCII	AES-encrypted PAN field if reading is valid.
LS Card data length	2	ASCII	Decimal length of AES/TDES-encrypted card data field
LS Card data field	n	ASCII	AES/TDES-encrypted card data field
Extended Language code	1	ASCII	First digit of Track-2 card language indicator

7.2.4.6.2 Manual Card Data Entry in E2EE Mode

For manual card data entry, the data is formatted as follows:

Manual-Entry Data Format

Data	Length	Type	Description
PAN, first six digits	6	ASCII	Clear value of the first six digits of the PAN
PAN, last four digits	4	ASCII	Clear value of the last four digits of the PAN
PAN length	2	ASCII	Decimal length of the PAN
PAN Mod-10 check flag	1	ASCII	• 0 = PAN failed MOD 10 check • 1 = PAN passed MOD 10 check
Expiry date	4	ASCII	Clear value of the expiry date (YYMM)
Service code	3	ASCII	00
Language code	1	ASCII	D0
Card Data Encrypted flag	1	ASCII	• 0 = Clear ASCII data. Only occurs if whitelisted by `E2EBIN` • 1 = Encrypted data

The remaining fields depend on the value of the Card Data Encrypted flag. If set to 0 for clear ASCII data, then the following fields are sent:

Fields with Card Data Encrypted Flag set to 0

Data	Length	Type	Description
Card data length	2	ASCII	Decimal length of the data field
Card data field	n	ASCII	Data field
Extended language code	1	ASCII	First digit of Track-2 card language indicator

If the Card Data Encrypted flag is set to 1, then the following fields are sent:

Fields with Card Data Encrypted Flag set to 1

Data	Length	Type	Description
Encrypted Format Type	1	ASCII	B = IngeCrypt (IC) data format
IC KSN	20	ASCII	DUKPT Key Serial Number (KSN)
Reserved	4	ASCII	0114
IC card data length	2	ASCII	Decimal length of IC-encrypted card data
IC card data field	n	ASCII	IC-encrypted card data if reading is valid

Data	Length	Type	Description
AES PAN length	2	ASCII	Decimal length of AES-encrypted PAN field
AES PAN field	n	ASCII	AES-encrypted PAN field if reading is valid
LS Card data length	2	ASCII	Decimal length of AES/TDES-encrypted card data field
LS Card data field	n	ASCII	AES/TDES-encrypted card data
Extended Language code	1	ASCII	First digit of Track-2 card language indicator

7.2.4.6.3 EMV Contact and Contactless in E2EE Mode

During an EMV transaction, the EMV 33.03.x Authorization Request Message is used in place of the 50.x Authorization Request message. In this message, the following tags are substituted with masked values:

- T5A (PAN)
- T57 (Track-2-equivalent data)

In the EMV 33.02.x Track 2 Equivalent Data Message and EMV 33.05.x Authorization Confirmation Response Message, the Track-2 values are replaced by the masked value.

EMV Tags Affected by On-Guard E2EE

When E2EE encryption is enabled, certain EMV tags function as follows:

Tag	Tag Name	With E2EE Encryption Enabled
DFF1D		Tag DFF1F is created and contains the encrypted Track-2-equivalent data. For On-Guard encryption, Tag DFF1D contains the masked PAN. See the format in the following topic, *Creating Tag DFF1D*.
T57	Track-2-equivalent data	Masked Track-2-equivalent data
T5A	PAN	If present: • First six digits and last four digits are returned in the clear • All other digits are masked with 0 if the PAN in tag 5A and tag 57 match
T5F24	Expiry date	Returned in the clear
T5F30	Service code	Returned in the clear
T56		All zeros

> The output AES PAN cryptogram is limited to a clear PAN maximum of 30 digits.

Creating Tag DFF1D

Data	Length	Type	Notes
PAN, first six digits	6	ASCII	Clear value of the first six digits of the PAN
PAN, last four digits	4	ASCII	Clear value of the last four digits of the PAN
PAN length	2	ASCII	Decimal length of the PAN
PAN Mod-10 check flag	1	ASCII	• 0 = PAN failed MOD 10 check • 1 = PAN passed MOD 10 check
Expiry date (see Note 1)	4	ASCII	Clear value of the Track-2 expiry date (YYMM)
Service code (see Note 1)	3	ASCII	Clear value of the Track-2 service code
Language code (see Note 1)	1	ASCII	Track-2 card language indicator
Cardholder name length	2	ASCII	Decimal length of the cardholder name
Cardholder name	n	ASCII	Clear value of the cardholder name
Card Data Encrypted flag (see Note 2)	1	ASCII	• 0 = Clear ASCII data. Only occurs if whitelisted by `E2EBIN` • 1 = Encrypted ASCII data

7.2.5 RSA-OAEP and TransArmor Encryption

RSA-OAEP encryption is RSA public key encryption with a key length of 2048 bits, and OAEP padding. TransArmor encryption is a special case of RSA-OAEP encryption, which uses the same encryption algorithm but differs in other details. This section applies to both encryption types unless noted otherwise.

See http://en.wikipedia.org/wiki/RSA_(algorithm) for algorithm details.

The application supports both MSR and contactless card data. Using the 23.x message to request manually entered card data is supported.

For both encryption types, the public key consists of a 2048-bit modulus and an exponent, which is typically set to '010001'. These values are configured in the `security.dat` section of `config.dfs`. The resulting `security.dat` file must be signed and downloaded to the terminal to enable the encryption.

7.2.5.1 Configuration Parameters (in config.dfs)

The relevant parameters in `config.dfs` include the following:

Parameters for RSA and TransArmor Encryption

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption (in `security.dat`)	0091_0001	0	Specify this value as '9' for RSA-OAEP, '10' for TransArmor.
Specify Encryption Key Slot (Key Index) (in `security.dat`)	0091_0002	6	RSA-OAEP/TransArmor ignores the value of this parameter. Encryption key slots are not used.
Configure Leading PAN Digits in the Clear (in `security.dat`)	0091_0003	6	RSA-OAEP/TransArmor ignores the value of this parameter. Specifies the number of leading digits to be displayed in the clear (Maximum = 6). The default value of 6 is hardcoded for RSA-OAEP/TransArmor.
Configure Trailing PAN Digits in the Clear (in `security.dat`)	0091_0004	4	RSA-OAEP/TransArmor ignores the value of this parameter. Specifies the number of trailing digits to be displayed in the clear (Maximum = 4). The default value of '4' is hardcoded for RSA-OAEP/TransArmor.

Parameter Name	DFS Data Index	Default Value	Description
Masking the PAN (in `security.dat`)	0091_0012	0	RSA-OAEP/TransArmor ignores the value of this parameter. Specifies the character to use for masking the PAN. The default value of '0' (zero) is hardcoded for RSA-OAEP/TransArmor.
Public Key encoded in Base64 (in `security.dat`)	0091_0013	(392-character string)	Made up of a 2048-bit modulus and an exponent (normally '65537'). See **Web page links** noted above for more information. This Public Key should be encoded in ASN.1 Base64 format, which will result in a 392-character string value for this parameter.
Exponent Value for RSA-OAEP/TransArmor (in `security.dat`)	0091_0014	010001	Specify this value as the default value = 010001. This overrides the exponent value from the public key in parameter '0091_0013'. This value is in binary format and should generally be set to the default (where 010001 = 65537), but may need to be modified – check with your key authority if you are unsure.
Key ID (in `security.dat`)	0091_0015	12345678901	For TransArmor, the 11-byte Key ID corresponding to the Public Key. Not needed for RSA-OAEP.
Encryption Target (in `security.dat`)	0091_0016	2	For TransArmor, set to 1 or 2 to indicate whether Track 1 or Track 2 data should be used in the encrypted data block. Ignored for RSA-OAEP. • 1 = Track 1 • 2 = Track 2 (Default)
Terminal ID (in `store.dat`)	0004_0013	12345678	For TransArmor, the Terminal ID to be used in the encrypted data block. Must be eight digits or fewer. Not needed for RSA-OAEP.

Encryption or card masking with PANs containing fewer than nine digits is not supported. Either whitelist these cards or disable non-standard card encryption.

7.2.5.2 Encryption Data Returned to the POS

The encrypted data (2048 bits or 256 bytes) is returned in Track 3 to the POS. The encrypted data is Base64-encoded, resulting in a 344-byte string.

For RSA-OAEP, the input to the encryption process consists of the concatenated raw Track 1, Track 2 and Track 3 data. If data is manually entered, then the input is the account number, expiration date and CVV2.

For TransArmor encryption, the input consists of the Terminal ID (padded to 8 bytes with leading 0's), followed by either the raw Track 1, raw Track 2, or manually-entered PAN, depending on the data available for the current transaction. The Track 3 data sent to the POS consists of three items separated by colons (":"):

- The 344-byte Base64 string of encrypted data.
- One digit indicating which data is encrypted:
  - 1 = Track 1.
  - 2 = Track 2.
  - 3 = PAN for manually-entered data.
- Key ID from the security.dat file.

Example of Track 3 Data for TransArmor Encryption (where Track 2 data is encrypted with the Key ID of 12345678901):

```
h7S2Qv71zutAc/6my+V3XaKQv62sQowIhnv2yhogDKylNchR28kv26ZfRrQCqyTkne7nTFjxiES5j0n
FJRax3xhO0EKwlohpDikEi4roStHvF80sY9KwJ+5Ugu0XC+YfubQacSKtZ2ic5ATLwqo0WhNkjgTB
to0yZNhiDRVWok7LGNMx9plqOXlG5nvzONkzLak72hbxjRH452QYN+qC+XcJKgSsQdxziMhNSyg
dUY7HcfQ1KQ0gkkZtwz5Ei+HFrVPKhheAivhJkOwrBa6w6humyvg+2A1VATGIZUkgXwYqRxf0/1R
SSgH29lHUXxmCn/MAa2/Ui34diQUnaolMLg==:2:12345678901
```

Track 1 and Track 2 in the RBA messages will contain masked Track 1 and Track 2 data.

7.2.5.3 RSA-OAEP and TransArmor Encryption Examples

7.2.5.3.1 Manual Entry Transactions

Variable IDs `399` and `402` (Account Name) return "Manual Entry" for manual entry when TransArmor encryption is enabled.

manualAccountName is replaced with 'msg23MsrName' in a 23.x message during an On Demand flow manual entry. Variable `399` then returns "Manual Entry" in the 29.x message as shown in the table below.

TransArmor Manual Entry in On-Demand Flow Example

Step	Notes
Enable TransArmor encryption.	
Enable '0007_0029'.	Set to any value '1' through '4'.
Send 23.x message while terminal displays a "Please slide card" form.	

<table>
<tr><th>Step</th><th>Notes</th></tr>
<tr><td>Press ENTER CARD button.</td><td></td></tr>
<tr><td>Enter PAN+Expiry Date+CVV value as per the '0007_0029' settings.</td><td></td></tr>
<tr><td>The terminal displays "card accepted" form and sends a 23.x response.</td><td></td></tr>
<tr><td>The POS prompts the terminal for variables with 29.x messages.
With RSA encryption enabled, or during a card swipe or contactless transaction in a normal transaction flow (not On-Demand), the '29.00000399' request would still return a card name in the response, e.g., '29.20000399TESTCARD/TEST'.</td><td><b>Sample 29.x Requests and Responses</b>
<table>
<tr><th>29.x Request</th><th>29.x Response</th></tr>
<tr><td>29.00000398</td><td>29.200003984445220000000007</td></tr>
<tr><td>29.00000399</td><td>29.60000399Manual Entry</td></tr>
<tr><td>29.00000400</td><td>29.200004000000</td></tr>
</table></td></tr>
</table>

7.2.6 S1 Encryption

7.2.6.1 S1 Encryption Overview

S1 encryption uses an injected TDES DUKPT key and has the following unique characteristics:

- Three data blocks are provided to the POS:
    - Sensitive Data Key Block - specifies key attributes used by the host to identify or derive the decryption key, including the KSN, and the "obfuscation scheme" which identifies whether the data were encrypted or whitelisted.
    - Volatile Encrypted Sensitive Data Block - contains an encrypted set of tag-value pairs for items such as EMV Track 2 Equivalent Data, magstripe track data, etc. The set of tags varies depending on the situation (magstripe, EMV, contactless, or manual entry).
    - Persisted Encrypted Sensitive Data Block - also contains an encrypted set of tag-value pairs, such as the PAN.
- S1 provides data origin authentication on all encrypted data transmitted from the terminal to the host. This is implemented by appending a secure 8-byte MAC (Message Authentication Code) to each of the encrypted data blocks. The host is required to validate the secure MAC prior to initiating decryption.
- When cards are whitelisted, a hash of the whitelist is provided so that the host can verify that the whitelist has not been altered.
- This requires the whitelist (or "encryption control list") to be formatted differently from the usual SECBIN.DAT. An XML file named S1LIST.XML is used instead. The terminal compares the card BIN to the whitelist to determine whether P2PE is mandated for the card type, and determines what obfuscation to apply to unencrypted card information returned to the POS.

7.2.6.2 Configuration Parameters (in config.dfs)

The following parameters relate to S1 encryption:

Parameters used by S1 Encryption

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption (in `security.dat`)	0091_0001	0	Encrypt track data sent in messages. - 12 = S1
Specify Encryption Key Slot (Key Index) (in `security.dat`)	0091_0002	4	Used to specify the terminal key slot containing the data DUKPT key to use for S1 P2PE. Valid values include 0 - 5.
Configure Leading PAN Digits in the Clear (in `security.dat`)	0091_0003	6	S1 defines its own obfuscation schemes, so it ignores the value of this parameter.
Configure Trailing PAN Digits in the Clear (in `security.dat`)	0091_0004	4	S1 defines its own obfuscation schemes, so it ignores the value of this parameter.
Masking the PAN (in `security.dat`)	0091_0012	0	S1 defines its own obfuscation schemes, so it ignores the value of this parameter.
Enable Security BIN Table (`secbin.dat`)	0092_0001	0	For S1 encryption, set this value to 0 (zero, the default) to enable use of the S1 Whitelist (instead of the SecBIN table).
Enable BIN range checking	0099_0001	0	Special note about BIN range checking and MOD-10: S1 P2PE requires MOD-10 checking. - If '0099_0001' is set to a value of '0' (zero = disables BIN range checking), then the MOD-10 flag in '0100_0005' (for `BIN0.DAT` file) must be set to '1' (one). - If '0099_0001' is set to a value of '1' (one = enables BIN range checking), then the MOD-10 flag in each `BINx.DAT` file must also be set to '1' (one).

7.2.6.3 S1 and MOD-10

S1 P2PE encryption requires Mod-10 checking. Refer to the Mod-10 Checking section for more information. Also refer to the "Enable BIN Range Checking" parameter in the table for Card Transaction Codes in the BIN Processing (allBins.dat, bin0.dat - bin20.dat) section for more information about enabling Mod-10.

7.2.6.4 Configuring the Encryption Control List, S1LIST.XML

The encryption control list specifies which PAN BIN ranges to encrypt, and which to not encrypt.

The S1 encryption control list is contained in an .XML file named `S1LIST.XML`. This file is optionally used when implementing S1 encryption- if not available, all PANs/BIN ranges are S1-encrypted. Note that S1LIST.XML is a user-friendly format; RBA internally translates it into the standard S1 format, that is hashed to protect the list from modifications.

The `S1LIST.XML` file must be packaged and signed by Ingenico as a .PGZ file before being loaded onto the terminal in the HOST directory. Loading may be performed using the 62.x File Write message, LLT, or any of the other standard Ingenico terminal loading and updating methods.

The fields in each entry are:

- BIN - the BIN to be checked
- BINLength - length of the BIN
- PANLength - the PAN length to be used when matching card numbers against the BIN
- Scheme - either "S1" for S1 encryption, or "N1" for no encryption (whitelisted card)

The control list is searched to find the best-fit BIN-range/PAN-length entry. Best fit is determined by matching PAN lengths, if possible, and most matching BIN range digits:

- If more than one entry with BIN range and exact PAN length matches the card's PAN, then the entry with the longest BIN range length is the best-fit match.
- If more than one entry with BIN range matches the card's PAN but no matching PAN length, then the entry with the longest BIN range length is the best-fit match.
- A PAN length of 0 indicates the least specific match, and can be used as a "wildcard" entry for PAN lengths that are not specifically listed.

```
<entry BIN="600649" BINLength="6" PANLength="19" Scheme="N1"
 Name="DOPGC,DOPPR,DOPZP,DOPCV"></entry>
<entry BIN="600649" BINLength="6" PANLength="16" Scheme="S1"
 Name="DOPGC,DOPPR,DOPZP,DOPCV"></entry>
<entry BIN="600649" BINLength="6" PANLength="18" Scheme="S1"
 Name="DOPGC,DOPPR,DOPZP,DOPCV"></entry>
<entry BIN="60064923" BINLength="8" PANLength="18" Scheme="N1"
 Name="DOPGC,DOPPR,DOPZP,DOPCV"></entry>
```

In this example, all four entries are for BIN range "600649" but the first "N1" entry indicates to whitelist (i.e. **NOT** encrypt) the PANs with this BIN range.

The middle two entries ensure that PANs with BIN range "600649" but with either 16 or 18 digits **will NOT** be whitelisted but **will** be encrypted – **except** that the last entry ensures that an 18-digit PAN with BIN range "60064923" will whitelist (i.e. **NOT** encrypt) the PANs with this more specific BIN range.

7.2.6.5 *Data Returned to the POS*

After a card is read, the S1 data blocks are returned in the "Track 3" field of RBA messages. The data blocks are concatenated together, separated by ':'s (colons). (If the actual Track 3 on a card is non-empty, the Track 3 data are obfuscated and then appended to the end of the sensitive data following a ':' (colon).)

Thus, the "Track 3" field of RBA messages will be formatted as follows:

SensitiveDataKeyBlock:VolatileEncryptedSensitiveDataBlock:PersistedEncryptedSensitiveDataBlock:Obfuscated track3DataFromCard

As with any binary data sent in an RBA message, the data blocks are in hex-ASCII format in accordance with the RBA message protocol.

> RBA formats sensitive data the same for whitelisted cards, but does not encrypt the data.

7.2.6.6 *Error Handling*

If there is an error with the encryption and/or sensitive data, a '23.7' response message is returned to indicate that an encryption error has occurred. If the card does not contain valid data or if there is any other error in the track data that would prevent encryption, then a '23.9' response message is returned. In both cases, a “card read error” message is displayed for a few seconds on the terminal.

7.2.7 TDES DUKPT Encryption for NCR/Retalix

7.2.7.1 *Overview*

With TDES DUKPT encryption for NCR/Retalix, the data format differs from standard TDES DUKPT encryption in that Track 1 and Track 3 data are suppressed, meaning they are not sent from the terminal to OpenEPS. The PIN and Track 2 data are fully encrypted before being sent to OpenEPS using P2PE encryption as implemented in the SCAT interface. The application will provide OpenEPS with the necessary data elements to process a transaction, including:

- First 6 digits and last 4 digits of PAN
- PAN length
- Expiration date
- Mod-10 check pass/fail status
- Track 2 length
- Any other data fields as applicable to the Ingenico - NCR interface

7.2.7.2 *NCR/Retalix Manual Entry*

If external BIN searching is enabled via the host (parameter '0005_0002' is set to '1') and the encryption method is selected as P2PE for NCR/Retalix (Parameter '0091_0001' is set to '14'), then a special manual 19.x message will be sent after the PAN is entered which will enable the POS to control the expiration date and security code entry portions of manual entry. Refer to the 19.x BIN Lookup Message section for more information.

7.2.7.3 *Implementation*

To facilitate this new encryption format, the Security Parameters (security.dat) configuration file has been updated as follows:

Parameter	DFS Data Index	Default	Description of Update
Enable Track Data Encryption	0091_0001	0	- 14 = P2P Encryption for NCR/Retalix.
Target Track to Encrypt	0091_0035	2	New parameter added to indicate the target track to encrypt for TDES DUKPT encryption for NCR/Retalix. - 1 = Track 1. - 2 = Track 2. - 3 = Track 3. - 4 = All available tracks.

When selecting NCR/Retalix P2P encryption, parameter '0091_0001' must be set to '14'. The target track to encrypt is by default set to track 2 (parameter '0091_0035' = '2'). If track 2 is the target encryption track and track 2 data is invalid, an error will be returned as opposed to falling back on track 1 data.

To enable this encryption mode and set its parameters, edit the SECURITY.DAT section of CONFIG.DFS. The resulting SECURITY.DAT file must be signed and downloaded to the terminal to enable the encryption. When using Generic TDES DUKPT Encryption, there are two options for incrementing the Key Serial Number (KSN). It can either be forced to increment, or it will automatically increment after 10 encryptions. Currently, RBA uses the automatic advance mode. Refer to TDES DUKPT Configuration for NCR/Retalix for a list of parameters to be configured for this encryption mode.

> The `secbin.dat` configuration file is referenced to indicate the account number that is to be whitelisted.

> Mod-10 checking is always performed for NCR/Retalix P2P encryption- enabling the Mod-10 flag in the bin range is neither necessary nor recommended.

7.2.7.4 Examples

As an example, data sent from the terminal to OpenEPS for a swiped card with '0091_0035' = '2' would be

```
[FS]5115080015637716=1312121633[FS]
```

Encrypted track 2 data is sent from the terminal to OpenEPS while track 1 and track 3 are suppressed.

For manually entered card data, the format is as follows:

```
M<account number>[FS]<exp data 'YYMM'>[FS]<CVV>
```

As an example; a card with an account number of '5115080015637716', and expiration date of 12/15, and CVV of '124' will be

sent as

```
M5115080015637716[FS]1512[FS]124
```

The preceding 'M' character defines the data as manually entered card data.

7.2.7.5 TDES DUKPT Configuration for NCR/Retalix

7.2.7.5.1 Configuring Security Parameters

Configuration information for TDES DUKPT is contained in two files: SECURITY.DAT and SECBIN.DAT. These are the same files used to configure encryption and security in the Telium RBA application. The specific security parameters relevant to TDES DUKPT encryption for NCR/Retalix are listed in the following table.

Parameters in security.dat used by TDES DUKPT Encryption for MTX

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption	0091_0001	0	Specify this value as '14' for TDES DUKPT encryption for NCR/Retalix.
Specify Encryption Key Slot (key Index)	0091_0002	4	Generic TDES DUKPT uses this DUKPT key slot for this feature. • Only slots 0-5can be used.
Configure Leading PAN Digits in the Clear	0091_0003	6	Specifies the number of leading digits to be displayed in the clear. • Maximum = 6. • The default value of 6 is hardcoded for Generic TDES DUKPT. Generic TDES DUKPT ignores the value of this parameter.
Configure Trailing PAN Digits in the Clear	0091_0004	4	Specifies the number of trailing digits to be displayed in the clear. • Maximum = 4. • The default value of 4 is hardcoded for Generic TDES DUKPT. Generic TDES DUKPT ignores the value of this parameter.
Masking the PAN	0091_0012	0	Specifies the character to use for masking the PAN. • The default value of 0 (zero) is hardcoded for Generic TDES DUKPT. Generic TDES DUKPT ignores the value of this parameter.

Parameter Name	DFS Data Index	Default Value	Description
Target Track to Encrypt	0091_0035	2	This parameter indicates the target track to encrypt for TDES DUKPT encryption for NCR/ Retalix. • 1 = Track 1. • 2 = Track 2. • 3 = Track 3. • 4 = All available tracks.
NCR/Retalix Encryption Suppress Data	0091_0038	1	When this parameter is enabled the following data will be suppressed from the encrypted data: • Track 1 data • Track 2 data • Cardholder name The settings for this parameter are as follows: • 0 = Disabled; do not suppress data. • 1 = Enabled; suppress data.

> Encryption or masking of cards with PANs containing less than 9 digits is not supported. Merchants should either whitelist these cards or disable non-standard card encryption.

7.2.7.5.2 Configuring for Manual Entry

In addition to configuring the security parameters, the "Enter Card" prompt display parameter ('0007_0029') must also be configured for the proper cardholder prompt. When using TDES encryption and manually entering data, the PAN, expiration date and CVV are all required. This parameter must therefore be set to '0' or '1' when using this encryption mode.

> The only files used by RBA are SECURITY.DAT and SECBIN.DAT. These files must be signed by Ingenico and downloaded to the terminal. This prevents an attacker from turning off encryption or modifying the settings.

7.2.8 Voltage TEP1 and TEP2 Encryption

7.2.8.1 Overview of Voltage TEP1 and TEP2 Encryption

Voltage encryption in RBA is an implementation of the Hewlett-Packard Enterprise Voltage SecureData Payments solution.

Voltage TEP1 encryption is whole-track encryption, while TEP2 encryption is structure-preserving encryption.

7.2.8.2 PAN Encryption Guidelines

- When Voltage TEP2 encryption is enabled, the PAN must contain a minimum of 12 digits. Up to six leading digits, as well as the last four digits are preserved and sent in the clear.
- For PANs containing 14 or more digits, the number of encrypted digits are the PAN length less 10. As an example, a PAN containing 15 digits has five digits encrypted.
- A PAN containing 16 digits has six digits encrypted. In all cases, a minimum of four digits, including the Luhn value, are encrypted. PANs embedded in Track 1 and Track 2 data are processed similarly.The track-embedded PAN length is shorter than the original PAN length, but the leading and trailing digits are preserved. The Luhn check generates the same result as the plaintext version.
- Additional track data is encrypted using Base64 alphabet and stored in the discretionary data field with the ciphertext of the original discretionary data value.
- When using Voltage TEP1 encryption, the Base64 character set is used and the Luhn value is disregarded. All digits in the PAN are encrypted. The PAN length is a minimum of 12 digits, and can be as long as the original unencrypted PAN. When performing manual entry, the encrypted PAN included in the 50.x Authorization Request message is always be the same length as the PAN entered using the terminal PIN pad. The following tables summarize Voltage TEP1 and TEP2 encryption of the PAN for a swiped card.
- Encrypting or masking cards with PANs containing fewer than nine digits is not supported. Whitelist these cards or disable non-standard card encryption.
- Voltage encryption handles account messages containing only the PAN. The PAN is always encrypted, regardless of whether or not the expiry date and CVV are also included in the message.

Voltage TEP1 Encryption of PAN for Swiped Card

Message Type	After Voltage TEP2 Encryption
19.x BIN Lookup Message	All digits of the PAN are encrypted. The encrypted PAN length included in the message is longer than the original PAN length.
23.x Card Read Request (On-Demand) 29.x Get Variable Request 50.x Authorization Request	All digits of the PAN are encrypted.

Voltage TEP2 Encryption of PAN for Swiped Card

Message Type	After Voltage TEP2 Encryption	PAN Before and After Encryption
19.x BIN Lookup Message 23.x Card Read Request (On-Demand) 50.x Authorization Request	• The encrypted PAN length included in the message is less than the original PAN length. • The first six and the last four digits of the PAN are sent in the clear. • The remaining middle digits of the PAN are encrypted.	5444009999222205 5444008062205
29.x Get Variable Request	• The PAN length is preserved, all digits are included in the message. • The first 6 and the last four digits of the PAN are sent in the clear. • The remaining middle digits of the PAN are encrypted.	5444009999222205 5444004114072205

Refer to Voltage TEP1 and TEP2 Encryption Examples for examples of TEP1 and TEP2 encryption. Also refer to the following table that describes the relevant parameters for Voltage Tep1 and TEP2 in the `config.dfs` file.

Relevant Parameters for Voltage TEP1 and TEP2 in config.dfs

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption (in `security.dat`)	0091_0001	0	Enable encryption for the encryption type: • 4 = Voltage TEP1 • 5 = Voltage TEP2
Max Number of Transactions with Same Key (in `security.dat`)	0091_0005	0	Maximum number of transactions with the same key. • 0 = Do not change keys based on transaction count. • 1 - 65000 = Change key after this many transactions with the same key.

Parameter Name	DFS Data Index	Default Value	Description
Periodically Change Keys (in `security.dat`)	0091_0006	0	Periodically change keys (Requires setting the terminals date and time). Enter all letters in UPPER CASE. • 0 = Disabled • D = Daily • SU = Change every Sunday • MO = Change every Monday • TU = Change every Tuesday • WE = Change every Wednesday • TH = Change every Thursday • FR = Change every Friday • SA = Change every Saturday • 01-31 = Change on the XX day of the month
Preserve Keys During Power Failure (in `security.dat`)	0091_0007	0	Preserve keys during power failure. • 0 = A new key is generated at power up. • 1 = Keys are saved when generated and restored at power up.

Parameter Name	DFS Data Index	Default Value	Description
Append ETB to 50.x Authorization Request (in security.dat)	0091_0008	0	Append ETB to 50.x Authorization Request Message. An ETB is created when a new key is created. • 0 = Do not append. • 1 = Append. Format is: <SS><Track 3 Card Data><RS><Encrypted track 1><RS><Encrypted track 2><RS><Encrypted PAN><RS><ETB><ES> Where: • <SS> = start sentinel ";" • <Track 3 Card Data> = Track 3 as read from card (could be empty) • <RS> = record separator 0x1E • <Encrypted track 1> = Voltage encrypted and base64 encoded (empty for manual entry) • <Encrypted track 2> = Voltage encrypted and base64 encoded (as described in section 4.3.2 for manual entry) • <Standalone PAN> = Voltage encrypted primary account number base64 encoded • <ETB> = Voltage ETB base64 encoded (this is optional based on 0091_0008) • <ES> = end sentinel "?"
Identity String (in security.dat)	0091_0009	id@sample.com	Identity String provided by the authorizer. "id@sample.com" is sample data, not for production.
Identity State (in security.dat)	0091_0010	*	Identity State. • Use format mmddyyyy. • If set to *, the terminal's current date will be used. Be sure to set the date and time via the 28.x Set Variable Request.
Parameter Data Encoded in base64 (in security.dat)	0091_0011		Provided by the authorizer.

Parameter Name	DFS Data Index	Default Value	Description
Masking Character	0091_0012	0	Character to use for masking. • For 0091_0012, only 0 and * are valid.
Length of Encrypted CVV in Voltage Encryption	0091_0017	8	For Voltage encryption, this value is the length of the encrypted CVV. Used for manual entry only. • Valid lengths are 7 - 23.

7.2.8.3 Voltage TEP1 and TEP2 Encryption Examples

7.2.8.3.1 Voltage Encryption Examples

Depending on the message used to view the encrypted card data, a variety of leading- and ending-digits will be in the clear and/or portions of the card data will be encrypted when using Voltage TEP1 or TEP2 encryption types.

> Remember that to enable the 19.x (BIN Lookup) message for use with either TEP1 or TEP2 encryption types, the following parameters must be set: '0005_0002' = ‘1’ or ‘2’, and '0005_0004' = ‘1’.

7.2.8.3.2 Voltage TEP1 Encryption Examples

The following table illustrates examples of Voltage TEP1 encryption.

Voltage TEP1 Encryption Examples

Content/Parameter	Example Value
Security Parameter Setting	Parameter '0091_0001' in the security.dat configuration file is set to '4' for Voltage TEP2.
Card in Clear	4077140046854992
PAN Encryption	+++++++Ab/KESest
Track 1 Encrypted Data	qrZ7cYgqSxh/LgKCEId4EqkL4bLOP+CjpFY6qlb/ mdibwen0oeVDIZoISWUro+15
Track 2 Encrypted Data	AFdnvAz0VP2kIFHCt/wX+vikfpg
23.x Card Read Request (On-Demand)	23.0I3yA1C2FsGSrt5UU8tKoaJiqFosOulZQ1yuXO+gbyw afLVwhNuPeiLznET9Os W8zwzfIeSDOV6p[FS]

Content/Parameter	Example Value
PAN Only Sent in 12.x Account Message	PAN in clear = 4012345678909 12.4012345678909 50.12345678901234567890123456789012345678900208020453600001@A401234 5678909[FS]1@[FS]12345[FS] Note that the PAN is sent in the clear when only the PAN is included in the 12.x Account message.
PAN + Expiry Date Sent in 12.x Account Message	PAN in clear = 4012345678909 12.4012345678909=1212 50.12345678901234567890123456789012345678900208020453600002@T+++++/ IakLV2q=1212[FS]1@[FS]3523[FS]
PAN + Expiry Date + CVV Sent in 12.x Account Message	PAN in clear = 4012345678909 12.4012345678909=1212[FS]333 50.12345678901234567890123456789012345678900208020453600003@A1LSak AaNvUH2WG9eX+CPAED4Zaf[FS]1@[FS]12345[FS]

1 Voltage TEP1 Encryption Examples

The following examples illustrate swiped card PAN encryptions viewed using various messages when encrypted with Voltage TEP1.

Voltage TEP1 Card Swipe PAN Encryption Examples

Message	PAN	Encrypted/Masked PAN
19.x BIN Lookup Message	5444009999222205	**Encrypted Track** 1: wHcwtWNPmagNhPLrAuITygvL0dbplOG HRKhEUj/SqdO6JSlRh7Cd 6biblSPn **Encrypted Track 2**: 8Pun/ 4ZxdghAfy0rnjVWXhzk4ye • All digits of the PAN are encrypted. • More than 16 digits/characters are displayed.
23.x Card Read Request (On-Demand)	341111597242000	++++++/85P8N0ru All 15 digits of the PAN are encrypted.
29.x Get Variable Request	341111597242000	++++++/85P8N0ru All 15 digits of the PAN are encrypted.
50.x Authorization Request	341111597242000	50.123456789012345678901234567890 1234567890020201684660002@D/ u2vCFy3nvM/ zximFEeA9CZAqYD[FS]1@[FS]1025[FS] All 15 digits of the PAN are encrypted.

With Voltage TEP1 encryption, the encrypted portion of the data is always unreadable.

7.2.8.3.3 Voltage TEP2 Encryption Examples

The following table illustrates examples of Voltage TEP2 encryption. With Voltage TEP2 encryption, the first 6 and the last 4 digits are always in the clear, and the remaining middle digits are always numerically encrypted.

Voltage TEP2 Encryption Examples

Content/Parameter	Description
Security Parameter Setting	Parameter '0091_0001' in the security.dat configuration file is set to '5' for Voltage TEP2.
Card in Clear	4077140046854992

Content/Parameter	Description
PAN Encryption	4077147429874992 - The middle 6 digits are encrypted.
Track 1 Encrypted Data	B4077141064992^YOU/A GIFT FOR^2106521Z8enOVxOyfA4ryC31EpKp5haA0h
Track 2 Encrypted Data	4077141064992=2106521T25N21u/dXMR9dcU
23.x Card Read Request (On-Demand)	23.0B4077141064992^YOU/A GIFT FOR^2106521Z8enOVxOyfA4ryC31EpKp5haA0h[FS]4077141064992=2106521T25N21u/dXMR9dcU
PAN Only Sent in 12.x Account Message	PAN in clear = 4012345678909 12.4012345678909 50.12345678901234567890123456789012345678900207012689400001@A4012345678909[FS]1@[FS] 18250[FS] Note that the PAN is sent in the clear when only the PAN is included in the 12.x Account message.
PAN + Expiry Date Sent in 12.x Account Message	PAN in clear = 4012345678909 12.4012345678909=1212 50.12345678901234567890123456789012345678900207012689400003@T4012316398909=1212[FS] 1@[FS]18250[FS]
PAN + Expiry Date + CVV Sent in 12.x Account Message	PAN in clear = 4012345678909 12.4012345678909=1212 50.12345678901234567890123456789012345678900207012689400003@T4012316398909=1212[FS] 1@[FS]18250[FS]

The following examples illustrate swiped card PAN encryptions viewed using various messages when encrypted with Voltage TEP2.

Voltage TEP2 Card Swipe PAN Encryption Examples

Message	PAN	Encrypted/Masked PAN
19.x BIN Lookup Message	5444009999222205	5444008062205 • Only 13 of 16 digits are displayed. • The first 6 and the last 4 digits are in the clear; the 3 middle digits are encrypted.
23.x Card Read Request (On-Demand)	5444009999222205	5444008062205 • Only 13 of 16 digits are displayed. • The first 6 and the last 4 digits are in the clear; the 3 middle digits are encrypted.
29.x Get Variable Request	5444009999222205	5444004114072205 • All 16 digits are displayed. • The first 6 and the last 4 digits are in the clear; the middle 6 digits are encrypted.
50.x Authorization Request	5444009999222205	5444008062205 • Only 13 of 16 digits are displayed. • The first 6 and the last 4 digits are in the clear; the 3 middle digits are encrypted.

> The encrypted portion of the TEP2 data is always numeric.

Manual Entry

Voltage encryption will encrypt in the same format for any manual entry whether parameter '0007_0029' (Display "Enter Card" Prompt) is set to '1', '2', '3', or '4'. The only changes are what information is appended after the PAN, if any (the '0007_0029' setting determines whether a customer is prompted for CVV and/or expiration date). The manual entry process is illustrated in the table below, using TEP2 encryption for the examples:

Voltage TEP2 Manual Entry Details

0007_0029 Value	Information Prompted	Input Example	Messages Sent with Data Encrypted
'1'	PAN, expiration date, and CVV.	• 4445222299990007 • 1512 • 123	• 23.0[FS]4445228903600007=1512=10571712 • 50.1234567890123456789012 3456789012345 67890020803443880009@T44 4522890360000 7=1512=10571712[FS]1@[FS]100 [FS]
'2'	PAN and expiration date.	• 6011000090102317 1 • 1512	• 23.0[FS]6011000090102317 1=1512 • 50.1234567890123456789012 3456789012345 67890020803443880009@T44 4522890360000 7=1512[FS]1@[FS]100[FS]
'3'	PAN and CVV.	• 6011000090102317 1 • 369	• 23.0[FS]4445228903600007==99236979 • 50.1234567890123456789012 3456789012345 67890020803443880009@T44 4522890360000 7==10571712[FS]1@[FS]100[FS]
'4'	PAN only.	• 6011000090102317 1	• 23.0[FS]4445228903600007 • 50.1234567890123456789012 3456789012345 67890020803443880009@T44 4522890360000 7[FS]1@[FS]100[FS]

Variable IDs 399 and 402 (Account Name) return "Manual Entry" for manual entry when Voltage encryption is enabled.

manualAccountName is replaced with 'msg23MsrName' in a 23.x message during an On Demand flow manual entry. Variable 399 then returns "Manual Entry" in the 29.x message as shown in the table below.

Voltage Manual Entry in On-Demand Flow Example

<table>
<tr><th>Step</th><th>Notes</th></tr>
<tr><td>Enable TransArmor encryption.</td><td></td></tr>
<tr><td>Enable '0007_0029'.</td><td>Set to any value '1' through '4'.</td></tr>
<tr><td>Send 23.x message while terminal displays a "Please slide card" form.</td><td></td></tr>
<tr><td>Press ENTER CARD button.</td><td></td></tr>
<tr><td>Enter PAN+Expiry Date+CVV value as per the '0007_0029' settings.</td><td></td></tr>
<tr><td>The terminal displays "card accepted" form and sends a 23.x response.</td><td></td></tr>
<tr><td>The POS prompts the terminal for variables with 29.x messages.
During a card swipe or contactless transaction in a normal transaction flow (not On-Demand), the '29.00000399' request would still return a card name in the response, e.g., '29.20000399TESTCARD/TEST'.</td><td><b>Sample 29.x Requests and Responses</b>
<table>
<tr><th>29.x Request</th><th>29.x Response</th></tr>
<tr><td>29.00000398</td><td>29.200003984445220000000007</td></tr>
<tr><td>29.00000400</td><td>29.200004000000</td></tr>
<tr><td>29.00000399</td><td>29.60000399Manual Entry</td></tr>
</table>
</td></tr>
</table>

7.2.9 Voltage TEP1x, TEP2x, and TEP4 Encryption

Voltage TEP1x and TEP2x function similarly to Voltage TEP1 and TEP2 respectively. The following exceptions apply:

- A 3072-bit encryption key is used.

Voltage TEP4 is a format-preserving encryption type that behaves identically to TEP2, with the following exceptions:

- 3072-bit key is used
- Only encrypts the middle six digits, leaving the leading and trailing digits in the clear.

7.3 Managing Keys

This section describes key injection and public key management.

7.3.1 Offline Remote Key Injection (RKI) Support

7.3.1.1 Overview

This section describes the Remote Key Injection feature, which permits keys for point-to-point encryption and PIN entry to be injected or updated without returning the terminal to a secure injection facility. Offline remote key injection uses a symmetric key to protect the keys to be injected. Keys are loaded on the terminal by downloading key bundle files with the .RKI extension.

7.3.1.1.1 Process

1. The merchant determines the terminals that require key injection and the key(s) to be injected to each terminal.
2. The merchant provides this information to Ingenico. The terminals are identified by their injected serial numbers. (While the set of keys can vary from terminal to terminal, typically the same set of keys is applied to all terminals in the list.)
3. Ingenico creates a key bundle file (.RKI file) and provides it to the merchant.
4. The merchant uses any available mechanism to download the .RKI file to the HOST directory on the terminals.
5. Upon reboot, each terminal processes the file and performs any key injections required. Status messages are displayed on the screen and also written to a log file in the HOST directory.
6. After the remote key injection process completes, RBA deletes the .RKI file.

7.3.1.1.2 Prerequisites

- To be eligible for RKI, a terminal must have an injected serial number. (This step is handled automatically prior to shipment from the factory in almost all cases.)
- The merchant needs a key bundle file with keys for injection.
- The terminal must be enabled for RKI via an RKI.XML file signed by Ingenico.

7.3.1.2 Enabling/Disabling Remote Key Injection

Remote key injection can be enabled or disabled by setting the parameter `RKIENABLE` in the **signed** `RKI.XML` file, and downloading it to the terminal. The following table summarizes the requirements for enabling the remote key injection feature:

Requirements for Enabling Remote Key Injection

RKI.XML File	RKIENABLE	Remote Key Injection
Not Present		Disabled
Present	0	Disabled
Present	1	Enabled

7.3.1.2.1 Key Bundle (.RKI File) Format

A key bundle file contains a series of data records. Each record contains:

- The injected serial number of the terminal to be injected
- The key data for the key to be injected
- The type of key (P2PE, PIN debit, and so on)
- An identifier specifying the key slot to be injected with that key

Within the file, keys are protected and encrypted according to the TR-31 standard for secure key exchange.

7.3.1.2.2 Initiating RKI

Download the key bundle file, and reboot the terminal. The terminal displays messages showing the RKI operations. These messages are also written to a text file, `RKILog.TXT`, in the HOST directory.

When the process is completed, the key bundle file is deleted.

7.3.1.2.3 Functional Limitations

- DUKPT keys cannot be re-injected.
- A maximum of 1000 DUKPT key injections is supported. Further RKI attempts are blocked, and the message, *Exceeded ORKI limit, return for refresh*, is displayed. A terminal must be returned to Ingenico to be reset.

7.3.1.2.4 Setting the RKIVERSION

The 62.x File Write messages can set the RKIVERSION during a file download.

The current RKIVERSION value can be retrieved with the 29.x Get Variable message with RBA variable 256.

7.3.2 Dynamically Updating RSA-OAEP Public Keys

7.3.2.1 Overview

Dynamic updating of RSA-OAEP Public Keys enables merchants to update encryption keys without requiring Ingenico to sign a new SECURITY.DAT every time a key is updated. This feature can be used with either generic RSA-OAEP encryption or TransArmor encryption.

The merchant must establish a **signing key** that RBA will use to verify any new keys as they are updated. Once the signing key is in place, RBA messages can be used to update the encryption key at any time.

The feature uses these subtypes of the 90.x P2PE Data Message:

- 90.5 RSA-OAEP Public Key Request/Response message.
- 90.6 Delete RSA-OAEP Public Key Request/Response message.
- 90.7 Select RSA-OAEP Public Key Request/Response message.

The feature also uses these parameters in the Security Parameters (security.dat) configuration file:

- 0091_0032 - Public Key for Signature Verification
- 0091_0033 - Public Key for Data Encryption

*7.3.2.2 **Procedure to establish a signing key***

1. The customer generates a **Signing** Public and Private key pair. The **Signing** Public key should be in the form of a PEM file, for example VERIFY1.PEM.
2. The customer also updates their SECURITY.DAT so that parameter 0091_0032 contains the name of the PEM file.

3. The customer sends the PEM file and the updated SECURITY.DAT to Ingenico for signing.
4. Ingenico returns corresponding PGZ files, i.e. VERIFY1.PGZ and SECURITY.PGZ.
5. The customer downloads the PGZ file containing the Signing Public Key (for example VERIFY1.PGZ) to the terminal, so that the **Signing Public key** is applied. The terminal must be rebooted to store the Signing Public Key.
6. The customer then downloads the updated SECURITY.PGZ and reboots the terminal.

These steps only need to be done once, as long as the Signing key pair does not change.

Once these steps are done for the first time, any previous RSA-OAEP encryption key is no longer in effect. The customer must load the desired encryption key through the update process.

*7.3.2.3 **Procedure to dynamically load or update an encryption key***

These steps assume that the signing key has been established in the terminal.

1. The customer generates an **Encrypting** Public and Private key pair.
2. The customer signs the **Encrypting Public key** with the **Signing Private key** (see below for details on this step).
3. The customer sends an RBA 90.5 message to the terminal containing the **Signed Encrypting Public key**.
4. The terminal receives the **Signed Encrypting Public key** from the POS and validates the signature using the **Signing Public key** referenced in the `SECURITY.PGZ` file.
5. Once the signature of the **Signed Encrypting Public key** is validated by the terminal, it is then stored and can be selected for encrypting cardholder data.
    a. If the signature is invalid then an error message will be returned to the POS.
    b. If the new **Signed Encrypting Public key** is downloaded but not signature-validated, then the RBA will respond with an error message and the new key will not be stored and cannot be used for encrypting cardholder data.
    c. A reboot is not required for this process.

The 90.5 request message sent from the POS to the terminal includes a key name, key file data, and signature data. When this message is received, the terminal uses the signature data and the signature verification public key to verify the new encryption public key. Once the signature is verified, the encryption public key is stored in the terminal and a 90.5 response message is returned to the POS with a result code (e.g., success, invalid request). Upon validation, the encryption public key data is stored in the terminals private area with the file name provided in the Key Name field of the request message. A ".PEM" extension will be appended to the file. If the signature is not validated then the encryption public key data will be discarded.

*7.3.2.4 **Other operations***

The 90.6 request message can be used by the POS to delete an encryption public key from the terminal.

The 90.7 request message can be used to select an encryption public key that has already been loaded on the terminal. Once a key has been selected, it is stored in the 0091_0033 parameter location (Public Key for Data Encryption) so that it can be reloaded upon rebooting the terminal. For more information on encryption support messages, refer to the 90.x P2PE Data Message section in this manual.

*7.3.2.5 **Procedure to sign an encrypting public key with the Signing Private Key***

The following code example illustrates how to generate a public key signature and convert it to base64 format for sending to the terminal:

```
#! /bin/sh

# generate a signature for a public key.
openssl sha1 -sign signing_private_key.pem -out signature.bin encryption_public_key.pem

# convert the signature data to base64 for transmission to terminal
base64 signature.bin | tr -d "\012" >signature.b64
```

'`| tr -d "\012"`' strips the linefeed characters from the output of base64. Deleting these linefeed characters is required due to the way RBA processes the signature data. If the base64 command generates carriage return/line feed character sequences to mark the end of lines then both characters will need to be deleted from the base64 output.

The resulting signature.b64 file contains the signature data that can be sent along with the key name and public key data (in encryption_public_key.pem from above commands) in the 90.5 message.

Important: Once the signature is generated and base64 encoded, neither the public key data that was signed nor the base64 encoded signature data can be altered. This would cause the verification to fail.

8 Implementing EMV

This section describes how Ingenico payment terminals process EMV transactions. EMV is the acronym for Europay, MasterCard, and Visa, co-developers of global standards for chip card transaction technology. These standards are managed by EMVCo, a company jointly owned by MasterCard, Visa, American Express, Discover, JCB, and UnionPay. EMV transaction specifications ensure global compatibility between EMV cards, payment terminals, and ATMs.

Refer to the following sections for more in-depth information on EMV transactions:

- Introduction to EMV Transactions
- EMV Transaction Sequence
- EMV Host Interface Messages
- EMV Transaction Flow
- EMV On-Demand Flow
- EMV Configuration and Flow
- EMV with P2PE Enabled

Additional information pertaining to EMV configuration parameters and MAC messages is provided in the following sections:

- EMV Configuration Parameters
- MAC Messages (Canada Only)
- Configuring the EMV Application

8.1 Introduction to EMV Transactions

EMV cards, also referred to as smart cards or chip cards, contain an embedded microchip which is configured to provide a more secure transaction than a standard magnetic stripe card transactions. Whereas a magnetic stripe card contains secure payment information which does not change, an EMV card can encrypt data differently with each transaction. See the following figure for an illustration of an EMV card.

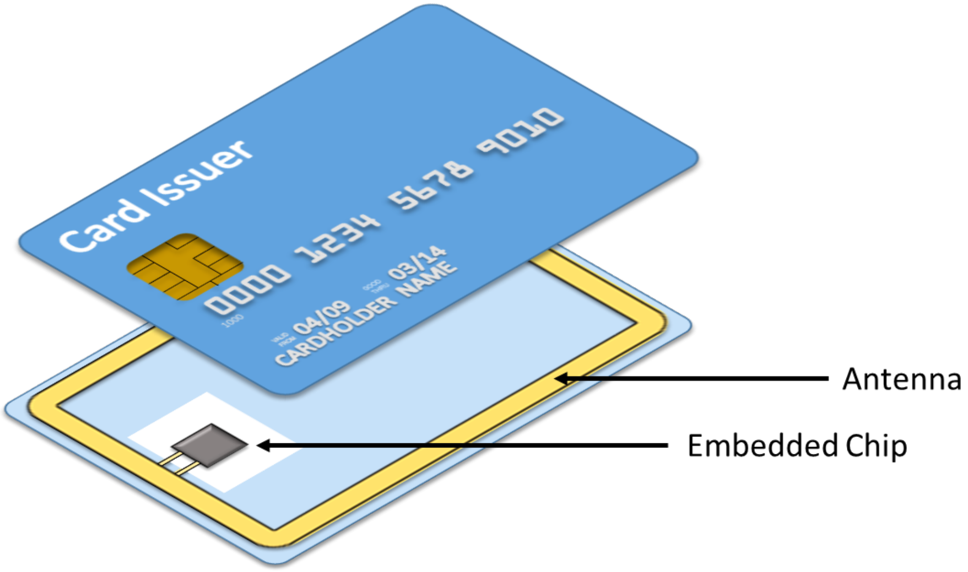
Card Issuer
0000 1234 5678 9010
1000
04/09
03/14
CARDHOLDER NAME

EMV Card

Referring to the above figure, note the window with the gold contact plates which is located on the left side of the card, just above the first four digits of the card number. The microchip is embedded in the card, just behind the gold contact plates. When the card is inserted in the chip card reader, a connection is made through these contacts which powers the microchip and enables it to communicate directly with the terminal.

EMV cards can interface with payment terminals as contact only, contactless only, or as dual-interface (both contact and contactless). An EMV card with contactless capability contains an embedded antenna, enabling it to interact with a payment terminal via radio waves. A contactless card requires no battery. When the card is placed in close proximity to the contactless card reader (within a few inches, typically), the RF field generated by the proximity coupling device flows through an inductor. The signal is rectified and converted to a DC voltage which applies power to the microchip.

Refer to the following figure for an illustration of a contactless EMV card. Note the antenna and embedded microchip.



EMV Card Anatomy

The following section describes the EMV Transaction Sequence, from card insertion through authentication and confirmation. This includes discussion on the interaction between the EMV card, terminal, and Point of Sale system (POS). Subsequent sections provide detailed information on EMV Host Interface Messages and how these messages are used in the EMV Transaction Flow, which includes in-depth information pertaining to purchase transactions, refunds, chip card insertion and contactless transactions. Also refer to the EMV with P2PE Enabled section for information on encryption during EMV transactions.

8.2 EMV Transaction Sequence

This section describes the steps involved in processing an EMV transaction, including a general description of the interaction between the card, terminal, and POS. The sequence of steps in a typical EMV transaction is as follows:

1. Card Detection
2. Language Selection for EMV Transactions
3. Application Selection
4. Read Application Data
5. Data Authentication

6. Cardholder Verification
7. Terminal Risk Management
8. Terminal Action Analysis
9. First Card Action Analysis
10. Online Transaction Authorization
11. Second Card Action Analysis
12. Transaction Completed

This section also provides a general overview of the decision making process involved with card authentication and transaction authorization. A specific protocol governing these processes is in place. Depending on the type of transaction and card configuration, different cryptograms are embedded in the transaction messages with time tags to further protect against fraud. The card itself is an integral part of the authorization process. For greater detail on the EMV transaction, refer to the following subsections.

8.2.1 Card Detection

Transactions are initiated by a 01.x Online Message (standard flow) or 23.x Card Read Request (On-Demand). The next action is to insert the EMV card which must be detected by the terminal in order to proceed with the transaction. As of RBA 21.0.1, a card may be inserted prior to the 23.x request in On-Demand flow. For a non-contactless card, it must be inserted in the chip card reader. The chip card reader is located at the front of the terminal, just beneath the keypad. See the below figure for an example of EMV card detection using an Ingenico iSC350 terminals.

EMV Card Inserted

EMV Card Tapped (Contactless)

If the EMV card is swiped using the MSR instead of the chip card reader, then the user should be prompted to “INSERT CARD IN CHIP READER.” This logic should be implemented in the POS; it can check the first digit of the service code, which will be ‘2’ or ‘6’ for EMV cards. For a contactless card, tapping the card (holding the card close enough to the contactless car reader) will serve the same purpose. Once the card is inserted, the terminal may request the cardholder to select the language, or to select the desired application.

8.2.2 Language Selection for EMV Transactions

Language selection may be performed manually or automatically. If the auto-selection flag for language is enabled and the EMV card has a preferred language list programmed into the chip, then the terminal will read this list and automatically select a language which is supported based on the priority assigned to it on the card. (For MSR cards, the language code is retrieved from the card language code from card Track 2.) If the EMV card does not have a preferred language list, or if the preferred language list does not include any languages which are supported by the terminal, then the cardholder will be prompted by the terminal to select a language which is supported.

The cardholder also has the option of manually selecting a language by pressing the desired language icon on the touch screen before inserting the card. The user must select a language in order to proceed with the transaction. The supported language list is provided in the `EMVCONTACT.XML` file ICS parameters. Languages are defined in tag `T9F8431`.

Language Selection Icons

8.2.3 Application Selection

8.2.3.1 Overview

Each terminal contains a set of applications that are supported during EMV transactions. These applications are uniquely identified and addressed using an Application ID, or AID. Data on an EMV card cannot be accessed until an application is selected. The following table describes example AIDs:

Example Application IDs

Card Issuer	Product	AID
VISA	VISA credit or debit	A0000000031010
	VISA Electron	A0000000032010
	VISA Plus	A0000000038010
MasterCard	MasterCard credit or debit	A0000000041010
	Maestro	A0000000043060
	Cirrus	A0000000046000
American Express	American Express	A00000002501
Interac (Canada)	Interac	A0000002771010
Discover	Discover	A0000001523010

8.2.3.2 *Application Selection Process Flow*

The EMV application selection flow is based on the `config.dfs` parameter `0019_0003` setting and also the setting of each AID's Confirmation Required tag `T87:b8`. Regardless of parameter `0019_0003` setting, if only one AID exists and the Confirmation Required flag is set to 0, this AID is auto-selected without cardholder confirmation. If set to 1, the application is auto-selected and the cardholder is prompted for confirmation of this selection. Depending on the setting of parameter `0019_0003` there are four possible flows for the application selection process:

- 0 = AID is selected by the cardholder via menu followed by cardholder confirmation.
  - View flow diagram Application Selection for 0019_0003 = 0
- 1 = Cardholder is prompted to confirm AID automatically starting with cards highest-priority AID.
  - View flow diagram Application Selection for 0019_0003 = 1
- 2 = AID is selected by the cardholder via menu without cardholder confirmation.
  - View flow diagram Application Selection for 0019_0003 = 2
- 3 = Cardholder is prompted to confirm AID automatically starting with cards highest-priority AID but the application auto-selects the lowest-priority application without cardholder confirmation if all other applications are declined.
  - View flow diagram Application Selection for 0019_0003 = 3

Auto-selection is the preferred method for two cases:

- Single-AID cards that do not require confirmation
- Dual-AID (US common or global) debit cards when either US common or global debit is filtered from the candidate list, which leaves only one AID available for processing payment.

Outside of these scenarios, merchants should give cardholders the option to select the AID on cards that contain two or more AIDs from different funding sources or different payment types (such as credit and debit).
The Telium kernel is certified with cardholder confirmation, which is shown in the LoA. EMVCo classifies this configuration as major, requiring cardholder confirmation in Telium terminal applications.
While parameter `0019_0003` allows three settings that require no confirmation, `0019_0003` = 1 alone conforms to EMVCo confirmation requirements.

For the list of available AIDs used in EMV transactions, refer to Configuring EMV Application IDs.

Refer to Interac EMV Requirements for more information about Canadian application configuration.

If an EMV card is swiped instead of inserted, the terminal displays the message, *Insert card in chip reader*, and waits for the card insertion if the service code is set to chip.

The first digit of the service code for EMV cards is 2 or 6 and their BIN range indicates that EMV is supported.

8.2.3.3 *Interac EMV Requirements*

Interac EMV cards are issued in Canada and have features and processing requirements that differ between how they are handled in the US and Canada.

For Canadian-issued EMV cards, the Canadian Application Selection Flag (ASF) enables a card issuer to encode multiple applications on one card in compliance with Interac Direct Payment (IDP) specifications, the EMV specifications for Canada. It permits the card issuer to determine the primary application to use at a payment terminal or ATM.

The Domestic VISA Debit Application Selection Flag can be used to enable or disable VISA Debit for Canada.

Application selection for Interac Application Selection and Domestic VISA Debit can be enabled or disabled in the EMV.DAT file, using these parameters:

- 0019_0007 - Interac Application Selection Flag
- 0019_0008 - Domestic VISA Debit Application Selection flag

8.2.3.3.1 AID Selection Process

A card can be encoded with a VISA AID (A0000000031010) as the primary AID for transactions, and an Interac AID as the primary AID for ATM transactions.

In Canada, when the card is inserted into:

- A payment terminal, both AIDs are included in the list of applications available for the transaction
- An ATM, only the Interac AID is included in the list of applications available for the transaction

8.2.3.3.2 Handling Canada-issued co-badged debit and credit cards in the US

In the US, VISA requires that co-badged VISA/Interac cards must be handled as VISA.

This requires that US terminals (those with the terminal country configuration, 0007_0044, set to 0 for the US) be configured as follows:

- If the merchant does **not** support the Interac AID, the ASF process must be disabled.
  - Set 0019_0007 = 0 - ASF is disabled
  - Remove the Interac AID from EMVCONTACT.XML
  - Result: Interac-only cards are not supported
- If the merchant **supports** the Interac AID, the ASF process must be enabled.
  - Set 0019_0007 = 1 - ASF is enabled
  - Include the Interac AID in EMVCONTACT.XML
  - Result: Interac-only cards are supported
- **In either case**, with a co-branded VISA and Interac card, RBA automatically selects the VISA AID.

8.2.3.4 Configuring EMV Application IDs

The default application configuration supports a basic set of application IDs (AIDs). AIDs can be added or removed, and parameters can be modified, by changing `emvaid.dat`, `emvbrand.dat`, `emvcontact.xml` and/or `emvcless.xml`. Settings for each AID are configured in each of these files.

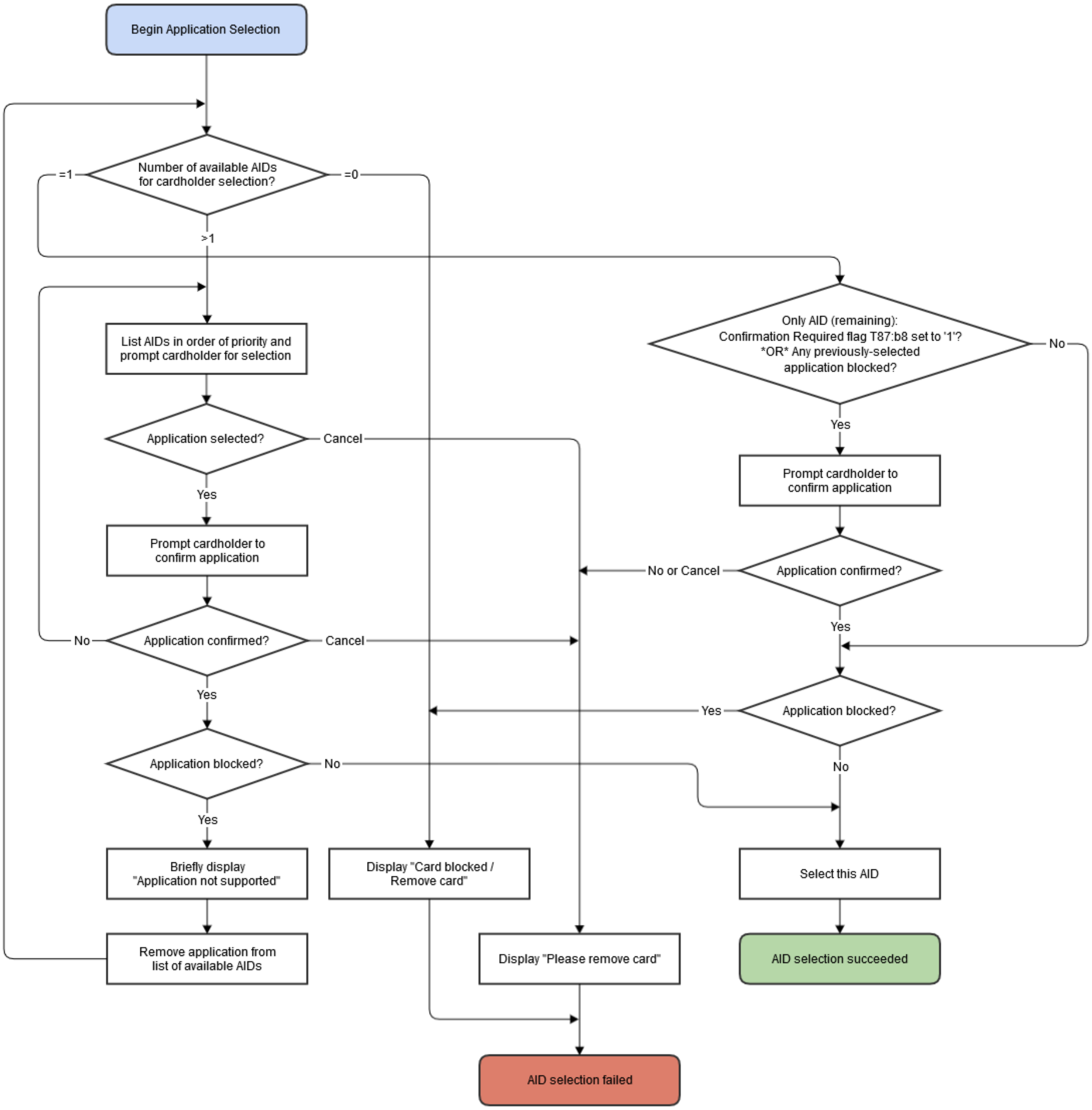
In order to support a card with a given AID, the AID must be configured in emvcontact.xml (to be supported for contact EMV cards) and/or emvcless.xml (for contactless). Otherwise, the EMV Kernel will not recognize the AID.

Configuration of the `emvcontact.xml` and `emvcless.xml` files is covered in EMV Configuration Parameters.

Once an AID is configured correctly in the .xml file(s), processing details for that AID can be further configured in `emvaid.dat.`

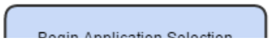
See EMV AID Parameters and EMV Brand Parameters for more information.

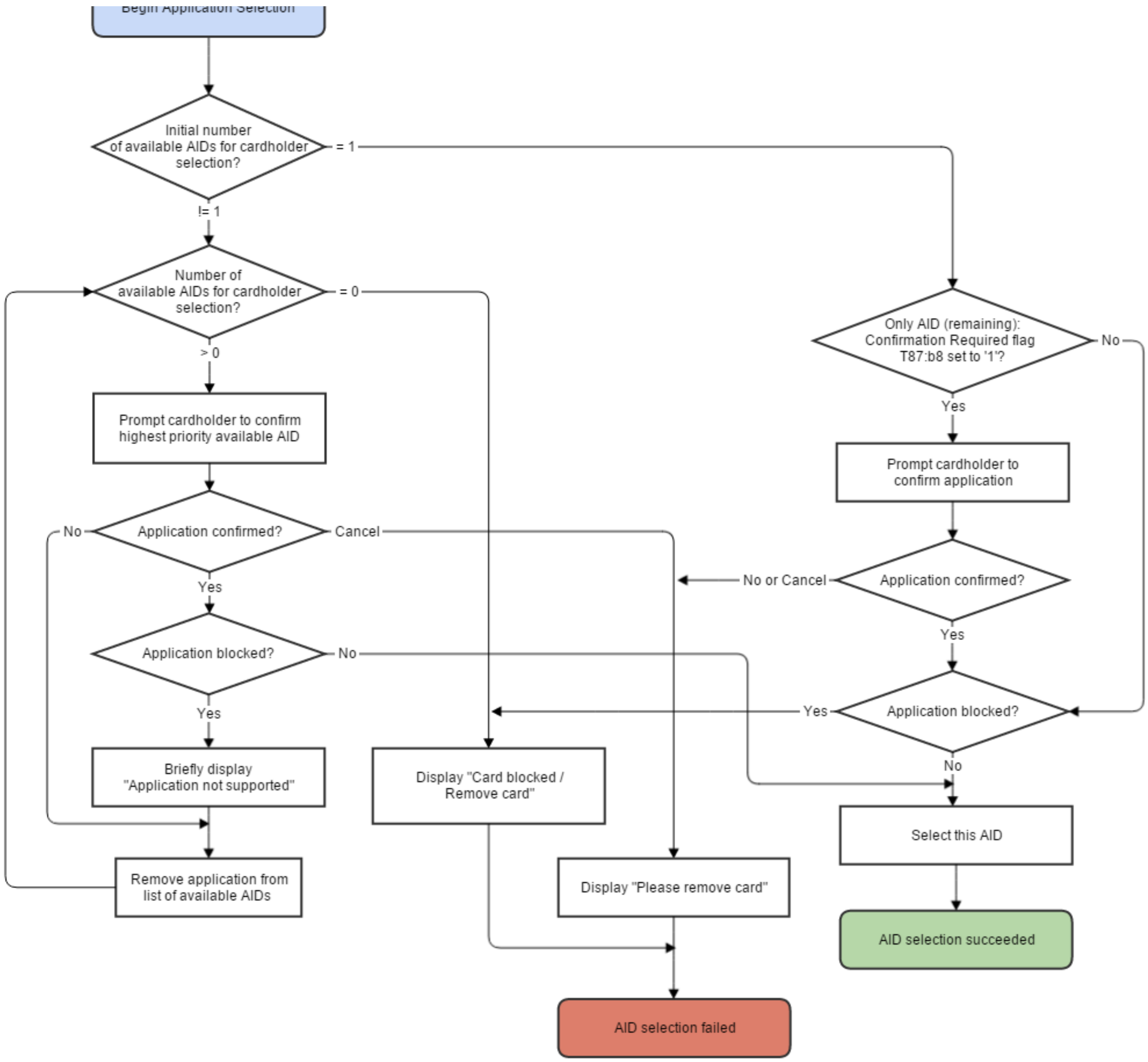
8.2.3.5 Application Selection for '0019_0003' = '0'


Begin Application Selection
Number of available AIDs for cardholder selection?
=1
=0
>1
Only AID (remaining): Confirmation Required flag T87:b8 set to '1'? *OR* Any previously-selected application blocked?
No
Yes
List AIDs in order of priority and prompt cardholder for selection
Application selected?
Cancel
Yes
Prompt cardholder to confirm application
Prompt cardholder to confirm application
No or Cancel
Application confirmed?
Yes
No
Application confirmed?
Cancel
Yes
Yes
Application blocked?
Application blocked?
No
No
Yes
Briefly display "Application not supported"
Display "Card blocked / Remove card"
Select this AID
Remove application from list of available AIDs
Display "Please remove card"
AID selection succeeded
AID selection failed

Application Selection for '0019_0003' = '0'

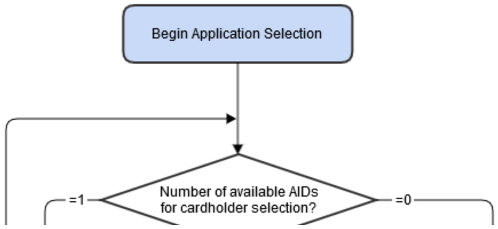
8.2.3.6 Application Selection for '0019_0003' = '1'

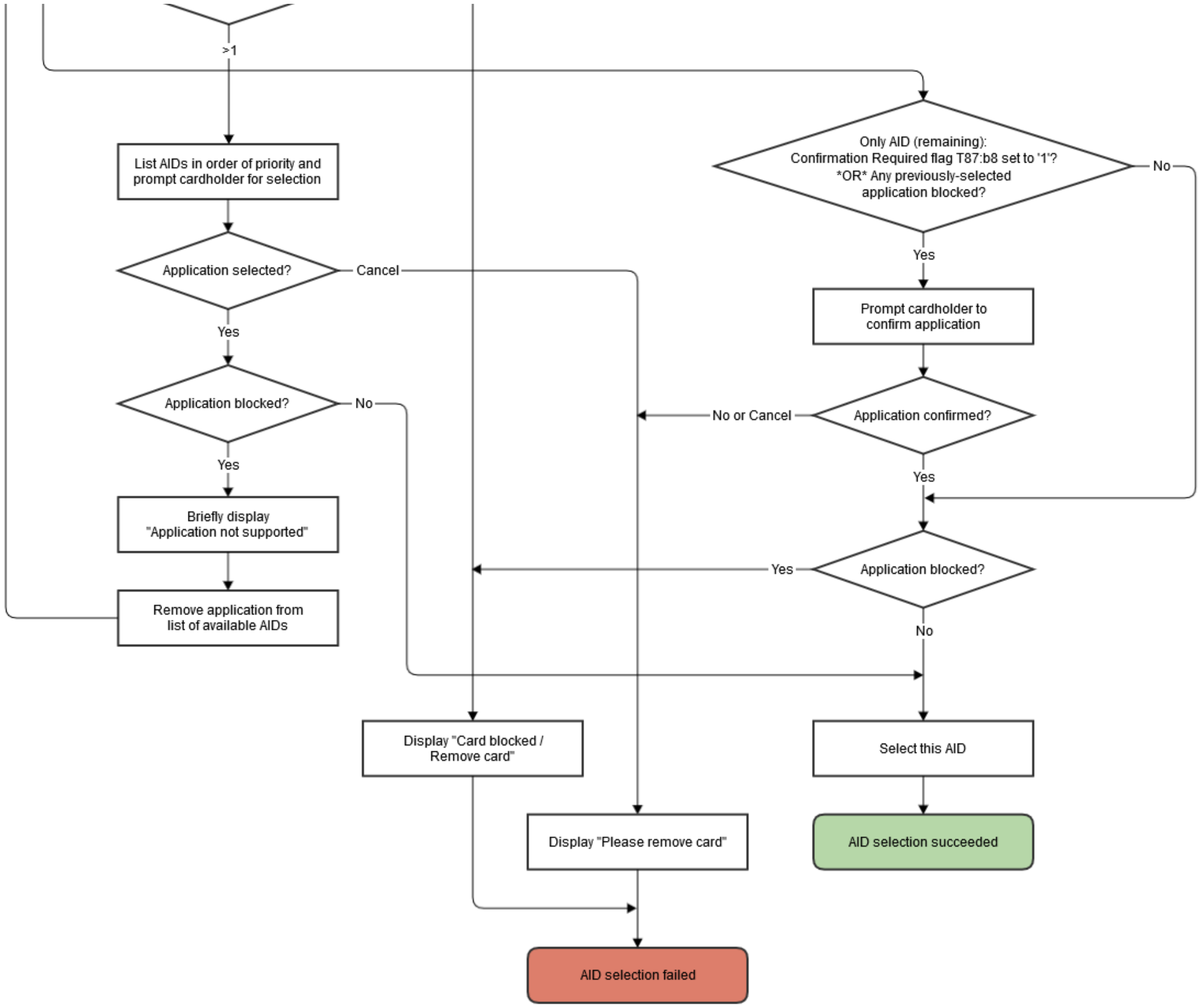



Begin Application Selection
Initial number of available AIDs for cardholder selection?
= 1
!= 1
Number of available AIDs for cardholder selection?
= 0
> 0
Prompt cardholder to confirm highest priority available AID
Application confirmed?
No
Cancel
Yes
Application blocked?
No
Yes
Briefly display "Application not supported"
Remove application from list of available AIDs
Display "Card blocked / Remove card"
Display "Please remove card"
AID selection failed
Only AID (remaining): Confirmation Required flag T87:b8 set to '1'?
No
Yes
Prompt cardholder to confirm application
Application confirmed?
No or Cancel
Yes
Application blocked?
Yes
No
Select this AID
AID selection succeeded

Application Selection for '0019_0003' = '1'

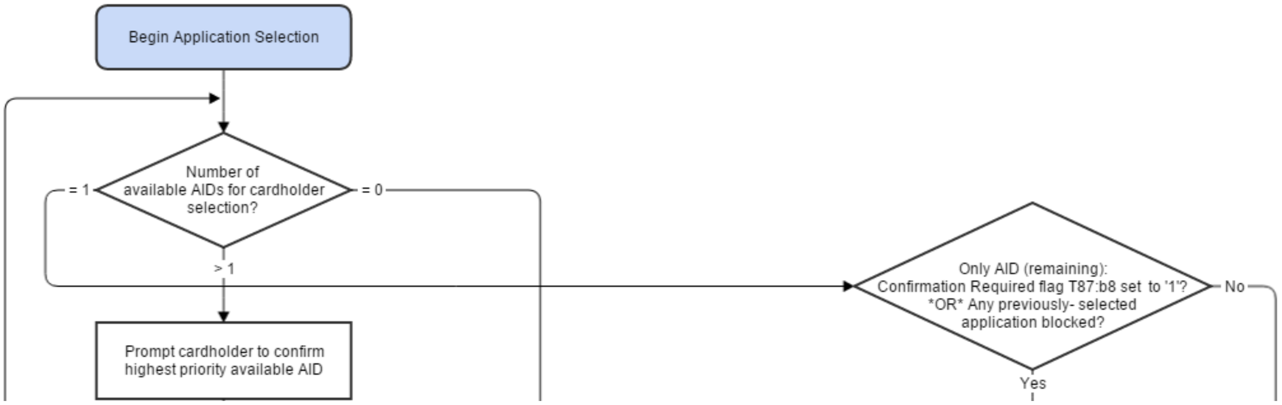
8.2.3.7 Application Selection for '0019_0003' = '2'

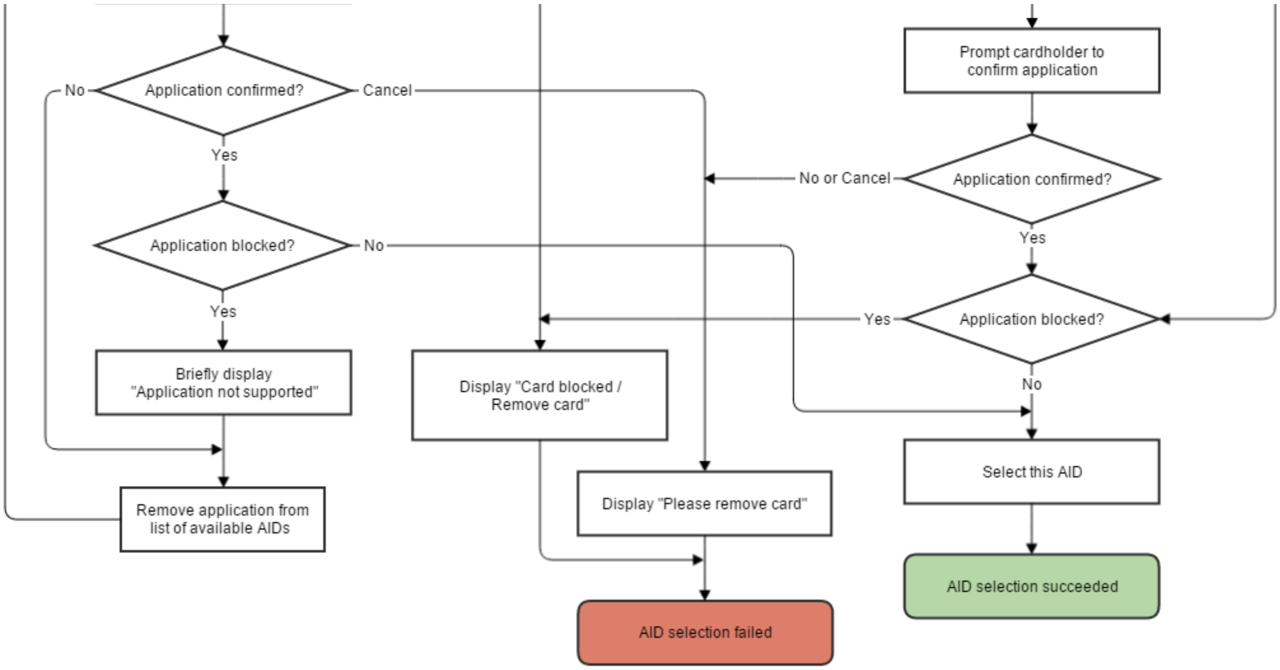

Begin Application Selection
=1
Number of available AIDs for cardholder selection?
=0


>1
List AIDs in order of priority and prompt cardholder for selection
Application selected?
Cancel
Yes
Application blocked?
No
Yes
Briefly display "Application not supported"
Remove application from list of available AIDs
Only AID (remaining): Confirmation Required flag T87:b8 set to '1'? *OR* Any previously-selected application blocked?
No
Yes
Prompt cardholder to confirm application
No or Cancel
Application confirmed?
Yes
Yes
Application blocked?
No
Display "Card blocked / Remove card"
Display "Please remove card"
Select this AID
AID selection succeeded
AID selection failed

Application Selection for '0019_0003' = '2'

8.2.3.8 Application Selection for '0019_0003' = '3'


Begin Application Selection
= 1
Number of available AIDs for cardholder selection?
= 0
> 1
Prompt cardholder to confirm highest priority available AID
Only AID (remaining): Confirmation Required flag T87:b8 set to '1'? *OR* Any previously- selected application blocked?
No
Yes


Application confirmed?
No
Cancel
Yes
Application blocked?
No
Yes
Briefly display "Application not supported"
Remove application from list of available AIDs
Display "Card blocked / Remove card"
Display "Please remove card"
AID selection failed
Prompt cardholder to confirm application
Application confirmed?
No or Cancel
Yes
Application blocked?
Yes
No
Select this AID
AID selection succeeded

Application Selection for '0019_0003' = '3'

8.2.4 Read Application Data

Once the application is selected, the next step is to read the application data. An Application File Locator (AFL) is read from the card and is used by the terminal to read data records from the card which are required for the transaction (e.g., card verification method list, Track 2 equivalent data). The terminal retrieves the information and then sends it to the POS using an EMV '33.02.x' Track 2 Equivalent Data Message.

8.2.5 Data Authentication

Data authentication is the process of performing a cryptographic validation of the EMV card using public key cryptography. Authentication may be performed using static Data Authentication (SDA), Dynamic Data Authentication (DDA), or by using a combination of Dynamic Data Authentication with the application cryptogram generated by the card itself. While both SDA and DDA methods protect against modification, the DDA method provides protection from cloning.

8.2.6 Cardholder Verification

A prioritized list of Cardholder Verification Methods (CVMs) is stored on the EMV card. This is due to the variance in supported CVMs from terminal to terminal. The terminal reads this list from the card during the cardholder verification process and selects the supported CVM with the highest priority. Refer to the following sections for more information on the cardholder verification processes:

- PIN Entry for EMV Transactions
- Signature Management
- EMV Reversal
- EMV Fallback

8.2.6.1 PIN Entry for EMV Transactions

PIN verification may be performed offline or online. The difference is in how the PIN verification is implemented. In the offline mode, PIN verification is performed by the card, whereas in the online mode PIN verification is performed by card issuer. In either case, if the PIN entry is cancelled by the user, then the transaction will be cancelled as well. Refer to the following sections for more information on offline PIN entry and online PIN verification:

- Offline PIN Verification
- Online PIN Verification

8.2.6.1.1 Offline PIN Verification

Once the PIN entry has been completed, the terminal proceeds according to the below table.

Action Taken Following PIN Entry

PIN Entry Status	Action Taken
Correct PIN entry	Terminal continues with the transaction.
Incorrect PIN entry	If the PIN is not blocked, the user is prompted to enter PIN again.
	If this is the last attempt to enter the PIN, a warning message is displayed to inform the cardholder.
	If the PIN is blocked, the terminal can either continue or abort the transaction according to the card CVM list.

8.2.6.1.2 Online PIN Verification

Online PIN entry requires that a PIN session key is loaded into the secret area. The PIN key index and encryption method are retrieved from the `PIN.DAT` parameter file. This key will be used to encrypt the PIN block value. The online PIN entry must be ignored in the case when the PIN session key is not found in the secret area.

8.2.6.2 Signature Management

The signature management function is already supported for MSR transactions. It is required under the following conditions:

- The `CARDS.DAT` parameter file has been configured to require pre-sign for the selected payment type.
- The `CARDS.DAT` parameter file has been configured to require a signature when the transaction amount exceeds a set threshold defined in this file.

For EMV transactions, a signature is required under the following conditions:

- A signature is required in the Cardholder Verification Method (CVM) result for the CVM used in the transaction.
- The transaction is a full EMV transaction without PIN entry.
- The transaction is an EMV contactless transaction and the amount exceeds the contactless CVM threshold. (There is a tag for each kernel type that indicates if a signature is required).

In all three cases, the process is the same. The RBA application loads the form of signature and awaits cardholder entry.

8.2.6.3 EMV Reversal

The EMV card must remain in the terminal until the transaction has been completed, as the card is required to send an application cryptogram to the Host immediately following the transaction. This application cryptogram is stored by the Host and authenticates the presence of the card during the transaction. Once the Host has approved the transaction and sent this decision back to the terminal, there are two conditions which may cause EMV reversal:

1. The transaction has been approved by the Host system, but the transaction is declined by the card.
2. The transaction has been approved by the Host system, but the card was removed prematurely.

The POS must detect these two cases and implement the reversal since it is responsible for Host communications.

8.2.6.4 EMV Fallback

Fallback is the process of changing the EMV transaction from reading the embedded chip on the card to reading the magnetic stripe on the back of the card. This may be implemented when the payment terminal is unable to communicate with the embedded chip. There are several scenarios where this can occur:

1. The card chip malfunctioned at the start of the transaction.
2. Other card chip related problems occurred during the transaction.
3. There are problems which are unrelated to the card, such as a MasterCard stipulation that requires fallback if the PAN within Track 2 differs from the actual PAN.

For the first scenario above, the RBA sends an EMV '33.05.x' Authorization Confirmation Message with tag `T1010` (Error Response Code) set to CDIV (Card Data Invalid) or CNSUP (Card Not Supported). The POS decides whether or not fallback is allowed.

For the second and third scenarios, the RBA sends an EMV '33.02.x' Track 2 Equivalent Data Message. EMV tags encoded in the message inform the POS of the potential problems. The POS then decides whether or not fallback is necessary.

If the fallback is required according to the above conditions, the POS should allow the swipe of a chip card for once (since the POS is doing the check of the service code to verify if the swiped card is an EMV chip card or not).

8.2.6.5 EMV Fallback for Unattended Terminals

8.2.6.5.1 Overview

EMV fallback can now be handled on cards inserted in unattended terminals without the need to reinsert the card. When a card is inserted, the application detects the chip. If no chip is detected, or if an error with the chip occurs, then the terminal can fallback to magnetic stripe read. If a chip is detected and identified as an EMV chip with no errors, then the terminal will initiate an EMV transaction.

In the event of an error associated with the card chip or no matching Application ID, the application can be configured to fall back to reading the magnetic stripe on the back of the card and continue with the transaction. The cardholder is notified that fallback is being implemented, and they are prompted to remove their card. By reading the magnetic stripe as the card is removed, this eliminates the need for reinserting the card. For more information on fallback, refer to the EMV Fallback section in this guide. An advantage of reading the magnetic stripe on the card as it is removed is that this ensures that data is read from the same card that exhibited the chip malfunction or failed to match an application ID.

8.2.6.5.2 Implementation

Parameters must be set to tell the application the conditions to detect the following:

- Detecting chip power on failure:
  - This criteria can be enabled (set to '1') or disabled via parameter '0019_0014'.
- Detecting no matching Application ID:
  - This criteria can be enabled (set to '1') or disabled via parameter '0019_0015'.

To enable RBA to implement fallback when one of these conditions is met, variable 420 must be set to 1. This variable functions as follows:

- 0 = Proceed with normal process
- 1 = Continue to fallback

By enabling fallback using this variable, the POS does not have to to manage fallback every time one of these conditions is met. Variable 420 resets at the start of each transaction.

8.2.6.5.3 Detecting and Enabling Fallback

The POS must notify the application of the conditions that trigger fallback. Fallback must be enabled via variable 420 as follows:

Detecting Chip Power On Failure and Allowing Fallback

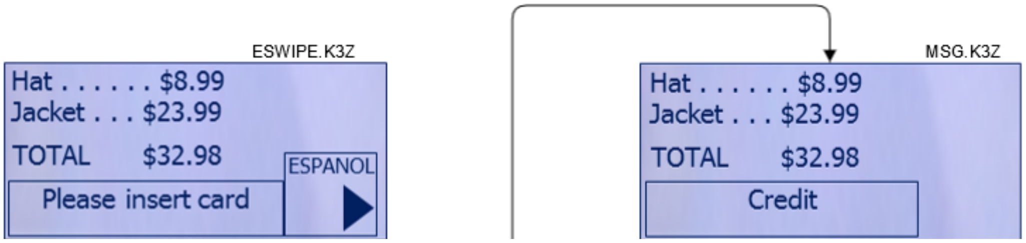
1. The POS Sets Parameter '0019_0014' to '1' using the 60.x message.
2. The POS sends a 01.x Online Message or 00.x Offline Message to the terminal.
3. The POS sends a '28.900004201' message setting variable 420 to '1' to enable fallback.

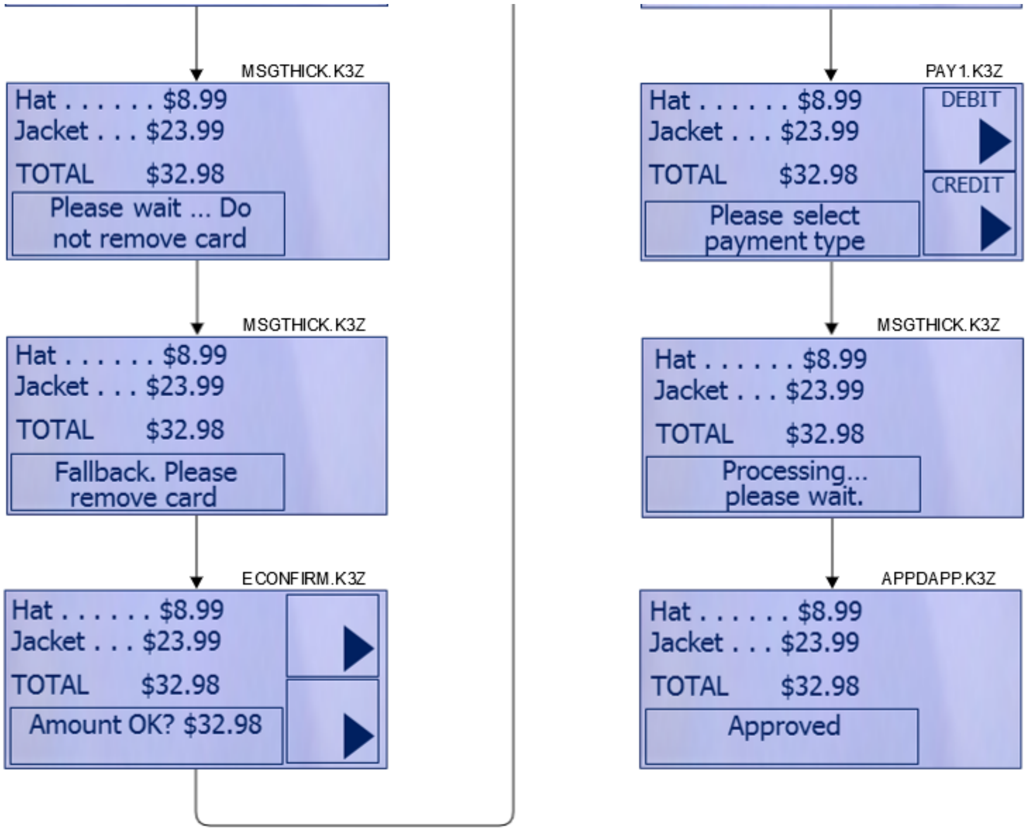
Detecting No Matching Application ID and Allowing Fallback

1. The POS Sets Parameter '0019_0015' to '1' using the 60.x message.
2. The POS sends a 01.x: Online Message or 00.x: Offline Message to the terminal.
3. The POS sends a '28.900004201' message setting variable 420 to '1' to enable fallback.

8.2.6.5.4 EMV Transaction Flow with Fallback

The following flow diagram illustrates an example transaction flow for a fallback scenario resulting from no matching Application ID. The RBA compares the list of Application IDs supported by the chip to those supported by the terminal. If there is no match, then the RBA will test configuration parameter '0019_0015' to determine if this criteria for fallback is enabled. The application then checks variable 420 to determine if fallback is enabled for the current transaction. When fallback occurs, the cardholder is prompted to remove their card. The terminal will emit a beep every 2 seconds until the card has been removed. As the card is removed, the magnetic stripe is read and if variable 420 is set to '1' then the application continues with the transaction by falling back to the MSR data processing.





iUN Example EMV Transaction Flow with Fallback for No AID Match

8.2.6.5.5 EMV Transaction Fallback for On-Demand Applications

When fallback occurs for an on-demand transaction, the RBA will include the magstripe data in tags sent to the POS via the EMV '33.05.x' Authorization Confirmation Response Message since the 23.x: Card Read Request (On-Demand) was already sent upon inserting the card. The following table lists the tags returned in the '33.05.x' message.

Tags Returned in the '33.05.x' Message for Fallback in On-Demand Mode

Tag	Value	Description
D1003	'F'	Fallback
D1017	MSR Track 1	Sample Track 1: B6510000000000133^CARD/IMAGE 13 ^17122011000048800000
D1018	MSR Track 2	Sample Track 2: 6510000000000133=17122011000048800000
D1019	MSR Track 3	

Tag	Value	Description

D101A	Fallback Status	FLBKOK = Magstripe read successful.
		FLBKERR = Magstripe read error.

Tags D1017, D1018, and D1019 will only be returned when a value is present.

8.2.6.5.6 Customizing the Fallback Prompt Message

The fallback prompt message can be changed by updating prompt ID 375 in the PROMPT.xml file.

8.2.7 Terminal Risk Management

Terminal risk management is performed to evaluate whether a transaction should be authorized offline or online. The amount of the transaction is compared to an offline limit. If this limit is exceeded, then online authorization is required.

8.2.8 Terminal Action Analysis

Once the terminal risk management has been completed, a decision must be made as to whether or not to proceed with the transaction, and if so, whether the transaction should be authorized offline or online. This decision is based on Issuer Action Codes (IACs) on the card, and Terminal Action Codes (TACs) configured for each AID in the `EMVCONTACT.XML` and `EMVCLESS.XML` configuration files.

8.2.9 First Card Action Analysis

The first card action analysis is performed by the EMV card, which makes the decision as to how to process the transaction (offline or online). The card requests a list of tags from the terminal which it will use in the decision making process. The terminal sends this tag data and then requests a cryptogram from the card based on its decision as to how to proceed with the authorization. Refer to the below table for the card response cryptogram method.

Card Response Cryptogram Method

Terminal Decision	Cryptogram Requested from Card
Offline Decline	Application Authentication Cryptogram (AAC)
Offline Approval	Transaction Certificate (TC)
Online Authorization	Authorization Request Cryptogram (ARQC)

8.2.10 Online Transaction Authorization

Online transaction authorization occurs when an Authorization Request Cryptogram (ARQC) is requested. This cryptogram is generated by the EMV card, providing what could be considered a digital signature of the transaction. This card issuer responds with an Authorization Response Cryptogram (ARPC) and authorization response (“Approved” or “Declined”). The ARPC is in turn verified by the card before validating the transaction.

8.2.11 Second Card Action Analysis

Following the online transaction authorization, the card will analyze the results and the terminal will request an approval status from the card ("Approved" or "Declined"). The card will communicate this decision to the terminal which in turn will complete the transaction.

8.2.12 Transaction Completed

Several options are available to reset terminals following EMV transactions and/or clear line display at the end of an EMV transaction:

- '10.1' and '15.8' reset messages from the POS to the terminal will only clear the line display.
- Any reset message (00.x, 10.x, 01.x) from the POS to the terminal will:
    - Interrupt and/or cancel the current EMV transaction
    - Reset the transaction, and
    - Prompt the cardholder to remove their card if inserted.
- iUN terminals beep as an audible cue when prompting the cardholder to remove their card.

> The 10.x message is not used at the end of completed EMV transactions. For EMV transactions, cancellation will cause RBA to send a '33.05.x' with `D1010` (Error Response) set to 'CAN'. However, 10.x may be sent during EMV transactions and typically indicates cardholder declined transaction amount(s).

8.2.12.1 Persisting Line Display

If a merchant wishes **NOT** to clear the line display until the card is removed, they have a few options:

- Wait for a 09.x remove message or an 11.x status change message before sending '10.1' or '15.8' reset messages.
- Don't send a hard reset message, e.g. '10.1'.

  > This second option is especially useful if an EMV transaction was declined and/or only a partial payment where another payment method is required. For any of these cases, the terminal may reset the transaction to await further payment without clearing the line items via 00.x or '10.0' reset messages.

8.3 EMV Host Interface Messages

8.3.1 Overview of EMV Host Interface Messages

When a transaction is being processed as an EMV transaction, communication messages between the terminal and the POS are uniquely identified by a 33.x message identifier. The message type is selected using a Subcommand Identifier embedded in the message as explained in the following table.

EMV Host Interface Message Subcommand Identifiers

Offset	Length	Format	Description
0	1	Constant	STX-0x02.
1	3	Constant	Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 00 = EMV '33.00.x' Transaction Initiation Message • 01 = EMV 33.01.x Status Message • 02 = EMV '33.02.x' Track 2 Equivalent Data Message • 03 = EMV '33.03.x' Authorization Request Message • 04 = EMV '33.04.x' Authorization Response Message • 05 = EMV '33.05.x' Authorization Confirmation Response Message • 07 = EMV '33.07.x' Terminal Capabilities Message • 08 = EMV '33.08.x' Set Variables Message • 09 = EMV '33.09.x' Set Tag Data Message • 10 = EMV '33.10.x' Get Tag Data Message
7	2	Alphanum	iConnectEFT Constant = P33_00_REQ_STATUS Status code. • 00 = OK - no error • 01 = FAIL - generic EMV message failure • 03 = INVALID_MSG_HEADER - invalid EMV message header format • 04 = UPDATE_LIST_FAIL - invalid status list field in 33.00 or 33.01 message • 05 = SUSPEND_LIST_FAIL - invalid suspend list field in 33.00 or 33.01 message • 06 = UPDATE_TIMER_FAIL - invalid timer field in 33.00 or 33.01 message • 07 = TAG_LIST_FAIL - missing or invalid tag field or invalid tags requested in EMV message • 08 = INVALID_TARGET - invalid configuration requested in 33.08 message (for example, not EMV or contactless) • 09 = INVALID_LOAD_SRC - configuration file requested in 33.08 message found but in invalid directory (not in secure application directory or /HOST) • 10 = INVALID_FILENAME - invalid configuration file name requested in 33.08 message • 11 = FILE_NOT_FOUND - configuration file requested in 33.08 message not found • 12 = INVALID_CMD_TYPE - missing or invalid 33.09/33.10 message command type (e.g. NOT an 'R', 'J', 'C', etc.)

Offset	Length	Format	Description
9	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_CURRENT_PACKET_NBR Current packet number. • 0 - 9 Starts at 0 for each new message, and increments for each packet of the message. The possible range is from 0 to 9, although in practice only one or two packets will be part of a request. The packet number wraps back to 0 after 9. EMV response messages always return 0.
10	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. EMV response messages always return 0. • 1 = Indicates that the packet is the first of more to come. • 2 = Indicates that more packets are to come. • 3 = Indicates the last packet.
Search for FS character to locate the next field.			
11	...	...	...
M	1	Constant	ETX-0x03.
M+1	1	Binary	LRC check character.

Also refer to EMV Tag Data Format for a description and examples of the data format of tags included in the 33.x messages.

8.3.2 EMV '33.00.x' Transaction Initiation Message

8.3.2.1 Overview

The EMV '33.00.x' Transaction Initiation Request message is sent from the POS to the terminal when initiating an On-Demand EMV transaction. This occurs once the RBA notifies the POS that an EMV card has been detected. The information provided in this message is used by the RBA to configure the transaction flow. The POS may optionally include one or more EMV/non-EMV tag values which will be used during the transaction. The format for the request and response messages are the same; the response message will contain the status codes but none of the optional fields.

Upon receiving and processing the EMV '33.00.x' Transaction Initiation Request message from the POS, the terminal will return an EMV '33.00.x' Transaction Initiation Response message. The transaction amount is now set via tag `T9F02`. This can be implemented using the EMV '33.00.x' Transaction Initiation Request message. The transaction amount can also be set or changed at a later point using the EMV '33.09.x' Set Tag Data message.

8.3.2.2 EMV '33.00.x' transaction Initiation Request Message Format

The following table describes the format for the EMV '33.00.x' Transaction Initiation Request Message.

EMV '33.00.x' Transaction Initiation Request Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_00_EMV_TRANSACTION_INITIATION Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 00. = Transaction initiation.
7	2	Alphanum	iConnectEFT Constant = P33_00_REQ_STATUS Status code.
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_PACKET_TYPE Packet type. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.
12	Variable	Alphanum	iConnectEFT Constant = P33_00_REQ_UPDATE_STEP_LIST Status update step list (optional, may be empty). This list includes the steps where the RBA should provide a status response message. This list may contain one or more steps, concatenated to one another. Refer to the Transaction Step List for a complete list of steps.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.

Offset	Length	Format	Description
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_00_REQ_SUSPEND_STEP_LIST Suspend status step list (optional, may be empty). This list includes the steps where the RBA should should suspend the transaction flow and await the resume message from the POS. This list may contain one or more steps, concatenated to one another. Refer to the Transaction Step List for a complete list of steps.
Search for FS character to locate the next field.			
N	1	Constant	FS – 0x1C.
N + 1	5	Constant	iConnectEFT Constant = P33_00_REQ_RESEND_TIMER Status message re-send timer (optional). This timer is used by the RBA to determine when to resend the status message during the suspend states if provided in the previous field. The timer value is in milliseconds.
Search for FS character to locate the next field.			
N + 6	1	Constant	FS – 0x1C.
EMV Tag and Data Fields			
N + 7	Variable	Alphanum	iConnectEFT Constant = P33_00_REQ_EMV_TAG EMV '33.00.x' Transaction Initiation data tags (optional). For the format of this field, refer to EMV Tag Data Format. Refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
Search for FS character to locate the next field.			
O	1	Constant	FS – 0x1C.
O + 1	1	Constant	ETX – 0x03.
O + 2	1	Binary	LRC check character.

8.3.2.3 *EMV '33.00.x' Transaction Initiation Response Message Format*

The following table describes the format for the EMV '33.00.x' Transaction Initiation Response Message.

EMV '33.00.x' Transaction Initiation Response Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.

Offset	Length	Format	Description
1	3	Constant	iConnectEFT Constant = M33_00_EMV_TRANSACTION_INITIATION Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 00. = Transaction initiation.
7	2	Alphanum	iConnectEFT Constant = P33_00_RES_STATUS Status code.
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always 0. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_PACKET_TYPE Packet type. Always 0. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.
12	1	Constant	ETX – 0x03.
13	1	Binary	LRC check character.

8.3.3 EMV 33.01.x Status Message

8.3.3.1 Overview

The EMV 33.01.x Status Request message enables the POS to request a status response from the terminal. This message can also be used to:

- Change the Status Update Step List
- Suspend the Status Step List
- Resend Status Message Timer values

Typically, the POS uses the EMV 33.00.x Transaction Initiation Message to:

- Request status messages for specific EMV transaction steps
- Suspend the transaction during specific steps in the EMV flow. Refer to the Transaction Step List.

8.3.3.1.1 Guidelines

- The terminal replies using the EMV 33.01.x Status Response message when it reaches the steps requested using the 33.00.x message.
- It notifies the POS when the application suspends the transaction flow to await action from the POS (Flag 26 in the status response message).
- If the EMV 33.01.x message is sent outside an EMV transaction, the message shows only dashes, except in the Card Inserted field. This field contains an I if a card is inserted and an R if there is no card. This flag does not indicate whether the card is a smart card. It only indicates whether a card is present. Deactivating this EMV function conserves battery life.

> **Warning**
> If an EMV card is removed while an EMV transaction is suspended, a 09.x Card Removed message is sent to the POS following the 33.05 Authorization Confirmation message; **however, if the card is removed during suspend step U, a 09.x Card Removed message is deferred until after the transaction is resumed.**

> Instead of using the EMV 33.00.x or 33.01.x messages to set status and the suspend list for contactless transaction, use `cless.dat` parameters 0008_0012, 0008_0013, and 0008_0014. See Contactless Reader Configuration (cless.dat) for more information.

8.3.3.1.2 EMV 33.01.x Status Request Message Format

The following table describes the EMV 33.01.x Status Request message format.

EMV 33.01.x Status Request Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_01_EMV_STATUS Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 01. = Status.
7	2	Alphanum	iConnectEFT Constant = P33_01_REQ_STATUS Status code.
EMV Request Header			

Offset	Length	Format	Description
9	1	Alphanum	iConnectEFT Constant = P33_01_REQ_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_01_REQ_EMVH_PACKET_TYPE Packet type. See EMV Host Interface Messages.
Search for FS character to locate the next field or End of Message.			
11	1	Constant	FS – 0x1C.
12	Variable	Alphanum	iConnectEFT Constant = P33_01_REQ_UPDATE_STEP_LIST Status update list (optional, may be empty). This field enables the POS to update this list if necessary. The lists consists of the transaction steps where the application should send an EMV 33.01.x Status Response message to the POS. This list may contain one or more steps, concatenated to one another. Refer to the Transaction Step List.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_01_REQ_SUSPEND_STEP_LIST Suspend status step list (optional; can be empty). This field enables the POS to update this list if necessary. The list contains the transaction steps where the application should suspend the flow and await the resume message from the POS. This list contains one or more of the steps listed in the Transaction steps concatenated together. Refer to the Transaction Step List.
Search for FS character to locate the next field.			
N	1	Constant	FS – 0x1C.
N + 1	5	Alphanum	iConnectEFT Constant = P33_01_REQ_RESEND_TIMER Status message re-send timer (optional). This field enables the POS to update the timer value, if necessary. This timer is used by the application to determine when to re-send the status message during the suspend states if provided in the previous field. The value for this timer is in milliseconds.
Search for FS character to locate the next field.			

Offset	Length	Format	Description
N + 6	1	Constant	FS – 0x1C.
N + 7	1	Constant	EXT–0x03.
N + 8	1	Binary	LRC check character.

8.3.3.1.3 EMV 33.01.x Status Response Message Format

The following table describes the EMV 33.01.x Status Response message format.

EMV 33.01.x Status Response Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_01_EMV_STATUS Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 01. = Status.
7	2	Alphanum	iConnectEFT Constant = P33_01_RES_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_01_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always 0. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_01_RES_EMVH_PACKET_TYPE Packet type. Always 0. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.
12	2	Alphanum	iConnectEFT Constant = P33_01_RES_TRANSACTION_CODE Transaction code. • 00 = Purchase. • 01 = Refund.

Offset	Length	Format	Description
14	1	Alphanum	iConnectEFT Constant = P33_01_RES_F1_CHIP_CARD Flag 1 - Indicates whether the chip card is inserted. • – = Chip card is not inserted. • I = Chip card is inserted. • R = Chip card is removed.
15	1	Alphanum	iConnectEFT Constant = P33_01_RES_F2_EMV_STARTED Flag 2 - Indicates whether the EMV process is started. • – = EMV process is not started. • S = EMV process is started.
16	1	Alphanum	iConnectEFT Constant = P33_01_RES_F3_EMV_COMPLETED Flag 3 - Indicates whether the EMV process is completed. • – = Not completed. • C = Completed (e.g., case of refund). • A = Completed with approval (e.g., case of purchase or refund). • D = Completed with decline (e.g., case of purchase/refund). • E = Error or incompletion reason. • F = Fallback to MSR (only for combination EMV/MSR readers, such as the iUN terminal) Icon See associated error flags for reason of the error, or reference the error response tag 0x1010 which will be sent in a 33.05 confirmation message.
17	1	Alphanum	iConnectEFT Constant = P33_01_RES_F4_LANGUAGE_SELECTED Flag 4 - Indicates whether the language is selected. • – = Not selected. • M = Manually selected. • A = Automatically selected.

Offset	Length	Format	Description
18	1	Alphanum	iConnectEFT Constant = P33_01_RES_F5_APP_SELECTED Flag 5 - Indicates whether the application is selected. • – = Not selected. • M = Manually selected. • A = Automatically selected.
19	1	Alphanum	iConnectEFT Constant = P33_01_RES_F6_APP_CONFIRMED Flag 6 - Indicates whether the application is confirmed. • – = Not confirmed. • A = Confirmation accepted. • R = Confirmation rejected.
20	1	Alphanum	iConnectEFT Constant = P33_01_RES_F7_REWARD_REQ_RECEIVED Flag 7 - Indicates whether a rewards 05. request is received. • – = Rewards request is not received. • R = Rewards request is received. • S = Rewards response sent.
21	1	Alphanum	iConnectEFT Constant = P33_01_RES_F8_PAYMENT_TYPE_RECEIVED Flag 8 - Indicates whether a payment type 04. request is received. • – = Payment type request is not received. • R = Payment type request is received. • S = Payment type response sent.
22	1	Alphanum	iConnectEFT Constant = P33_01_RES_F9_AMOUNT_CONFIRMED Flag 9 - Indicates whether a purchase or refund amount is manually confirmed by the user. • – = Amount not confirmed. • A = Amount confirmation accepted. • R = confirmation rejected.
23	1	Alphanum	iConnectEFT Constant = P33_01_RES_F10_LAST_PIN_TRY Flag 10 - Indicates whether this is the last PIN try for the card. • – = This is not the last PIN try. • L = This is the last PIN try.

Offset	Length	Format	Description
24	1	Alphanum	iConnectEFT Constant = P33_01_RES_F11_OFFLINE_PIN_ENTERED Flag 11 - Indicates whether an offline PIN is entered. • – = Offline PIN is not entered. • P = Offline PIN is entered. • B = PIN bypassed.
25	1	Alphanum	iConnectEFT Constant = P33_01_RES_F12_ACCOUNT_TYPE_SELECTED Flag 12 - Indicates whether an account type is selected. • – = Account type is not selected. • C = Checking account type is selected. • S = Savings account type is selected.
26	1	Alphanum	iConnectEFT Constant = P33_01_RES_F13_AUTH_REQ_SENT Flag 13 - Indicates whether the authorization request is sent. • – = Authorization request is not sent. • S = Authorization request is sent. • F = Authorization request failed to send.
27	1	Alphanum	iConnectEFT Constant = P33_01_RES_F14_AUTH_RES_RECEIVED Flag 14 - Indicates whether the authorization response is received. • – = Authorization response is not received. • R = Authorization response is received. • T = Internal terminal timeout on authorization response. • H = Register indication of no Host available; down or timeout.
28	1	Alphanum	iConnectEFT Constant = P33_01_RES_F15_CONFIRMATION_RES_SENT Flag 15 - Indicates whether the confirmation response is sent. • – = Confirmation response is not sent. • S = Confirmation response is sent. • F = Confirmation response failed to send.

Offset	Length	Format	Description
29	1	Alphanum	iConnectEFT Constant = P33_01_RES_F16_TRANSACTION_CANCELLED Flag 16 - Indicates whether the transaction is cancelled. • – = Transaction is not cancelled. • C = Transaction is cancelled.
30	1	Alphanum	iConnectEFT Constant = P33_01_RES_F17_CARD_CANNOT_READ Flag 17 - Indicates whether card data unreadable or is of an invalid format. • – = Card data is not invalid or is not detected. • I = Card data is invalid but fallback is allowed. • N = Card data invalid, fallback data not allowed due to being an Interac debit transaction.
31	1	Alphanum	iConnectEFT Constant = P33_01_RES_F18_CARD_OR_APP_BLOCKED Flag 18 - Indicates whether a card or application block is detected. • – = Card or application block is not detected. • A = Application is blocked. • B = Card is blocked.
32	1	Alphanum	iConnectEFT Constant = P33_01_RES_F19_ERROR_DETECTED Flag 19 - Indicates whether an error or incompletion is detected. • – = No fatal error is detected. • F = Fatal error is detected. • K = Track 2 data consistency failed. • O = User interface timeout. • X = EMV cards application is expired. • C = Cashback error (e.g. on-demand cashback amount > current transactions maximum cashback amount). • B = Cashback requested before PIN entry but PIN bypassed.
33	1	Alphanum	iConnectEFT Constant = P33_01_RES_F20_PREMATURE_CARD_REMOVAL Flag 20 - Indicates whether completion occurred from premature card removal. • – = No premature card removal was detected. • R = Premature card removal was detected.

Offset	Length	Format	Description
34	1	Alphanum	iConnectEFT Constant = P33_01_RES_F21_CARD_NOT_SUPPORTED Flag 21 - Indicates whether the card is not supported. • – = No status is available. • N = Card is not supported (e.g., Application ID is not found).
35	1	Alphanum	iConnectEFT Constant = P33_01_RES_F22_MAC_VERIFICATION Flag 22 - Indicates status of MAC verification required on Interac debit authorization response. • – = No MAC verification performed in transaction • P = MAC verification passed. • F = MAC verification failed.
36	1	Alphanum	iConnectEFT Constant = P33_01_RES_F23_POST_CONFIRM_START_TO_WAIT Flag 23 - Indicates whether the post confirmation wait has been started. • – = Post confirmation wait has not been started. • S = Post confirmation wait has started. Icon Post confirmation wait can only start if it is configured through the unit data message.
37	1	Alphanum	iConnectEFT Constant = P33_01_RES_F24_SIGNATURE_REQUEST Flag 24 - Indicates whether a signature request has started or ended. • – = No signature request has been detected and started. • S = Signature request has been detected and started. • E = Signature request has completed. • R = Paper signature requested.

Offset	Length	Format	Description
38	1	Alphanum	iConnectEFT Constant = P33_01_RES_F25_TRANSACTION_PREPARATION_SENT Flag 25 - Indicates whether the transaction preparation response has been sent to the register. • – = Transaction preparation response not sent. • S = Transaction preparation response sent. • F = Transaction preparation response failed to send.
39	1	Alphanum	iConnectEFT Constant = P33_01_RES_F26_EMV_FLOW_SUSPENDED Flag 26 - Indicates whether the EMV flow is suspended. • – = Suspend operation did not occur. • 1 = EMV flow is suspended. • 0 = EMV flow is resumed.
40	1	Alphanum	iConnectEFT Constant = P33_01_RES_F27_ONLINE_PIN_REQUESTED Flag 27 - Indicates whether Online PIN entry is requested. • – = Initialized state at start of transaction. • R = PIN entry request. Flag set to R at start of first online PIN entry. • C = PIN entry cancelled, having failed due to invalid PIN. It will remain set as R if manually cancelled by the Cancel button. • A = PIN block is accepted and valid. Updated when host returns `T8A` = 00 or other transaction approval. • B = PIN bypassed. Can be bypassed via Enter key or card removal. • E = PIN entry failed for any error (including PIN entry timeout). • I = PIN entered is invalid. Updated when host returns `T8A` = 55. PIN entry will restart, but flag will remain set to I until updated (rather than resetting to –). • D = PIN not verified. Updated when host returns `T8A` = 05 or other transaction declination.
41	1	Alphanum	iConnectEFT Constant = P33_01_RES_F28_CURRENT_EMV_STEP Flag 28 - Indicates the current EMV step. • – = EMV transaction not started. • For all other values refer to the Transaction Step List.
42	1	Alphanum	iConnectEFT Constant = P33_01_RES_F29_RESERVED Flag 29 - Reserved

Offset	Length	Format	Description
43	1	Alphanum	iConnectEFT Constant = P33_01_RES_F30_RESERVED Flag 30 - Reserved
44	1	Alphanum	iConnectEFT Constant = P33_01_RES_F31_EMV_CASHBACK Flag 31 - EMV cashback. • - = Initialized at start state (before cashback request is set). • R = Cashback requested (when configuration is set). • C = Cashback is accepted (after cashback value is set by POS). Flag 31 is used as follows to implement cashback in an EMV transaction: 1. The device sets Flag 26 to 1 and Flag 31 to R to suspend the EMV flow and indicates cashback is requested after PIN entry. 2. POS prompt for cashback. 3. POS sends an Amount Change message with the cashback amount. 4. A new Status message with Flag 26 set to 0 and Flag 31 set to C is returned to indicate completion of cashback request.
45	1	Alphanum	Flag 32 - P33_01_RES_F32_CONTACTLESS_STATUS. • - = Contactless transaction not yet started. • 1 = Contactless transaction started • 0 = Contactless transaction stopped.
46	1	Alphanum	Flag 33 - P33_01_RES_F33_CONTACTLESS_ERROR. • - = No error. • C = Collision detected. • R = Re-tap required.
Location where additional parameter may be added in the future. Search for FS character to locate the next field.			
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
M + 1	1	Constant	ETX – 0x03.
M + 2	1	Binary	LRC check character.

8.3.4 EMV '33.02.x' Track 2 Equivalent Data Message

The EMV '33.02.x' Track 2 Equivalent Data message is sent from the terminal to the POS. This data is similar to the Track 2 data which is stored on a magnetic stripe card. Included in this message is the information necessary for the POS to initiate the EMV transaction. During the initial stages of an EMV transaction, the POS may require tag information from the card and terminal. This message is used to convey this information. Tag data such as terminal serial number, Track 2 equivalent data, Primary Account Number (PAN), PAN sequence number, and issuer country code are included in this message. The below table provides more information on this message.

EMV '33.02.x'Track 2 Equivalent Data Message

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_02_EMV_TRANSACTION_PREPARATION_RESPONSE Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 02. = Track 2 Equivalent Data.
EMV Request Header.			
7	2	Alphanum	iConnectEFT Constant = P33_02_RES_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_02_RES_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_02_RES_EMVH_PACKET_TYPE Packet type. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
EMV tag and data fields.			
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_02_RES_EMV_TAG EMV '33.02.x' Track 2 Equivalent Data Message tag and data field. For the format of this field, refer to EMV Tag Data Format. Refer to Non-EMV Tag Definitions for more information on non-EMV tags. For the specific tag field IDs and data transmitted for this message, refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
Search for FS character to locate the next field.			

Offset	Length	Format	Description
N	1	Constant	FS – 0x1C.
Additional tag or data fields.			
Search for FS character to locate the next field.			
O	1	Constant	FS – 0x1C.
O + 1	Variable	Alphanum	Final transaction preparation response tag or data field.
Search for FS character to locate then next field or End of Message.			
P	1	Constant	ETX – 0x03.
P + 1	1	Binary	LRC check character.

8.3.5 EMV '33.03.x' Authorization Request Message

8.3.5.1 Overview

The EMV '33.03.x' Authorization Request message is sent from the terminal to the POS to provide the cryptographic information necessary to authorize the transaction. The authorization process is initiated by the terminal issuing a request to the POS. The POS then responds to the terminal with a final confirmation. Refer to the below table for a description of this message.

The following three tags are required to be included in the EMV '33.03.x' message to pass TSYS certification:

- 9F21 - Transaction Time.
- 9F39 - POS Entry Mode.
- 9F40 - Additional Terminal Capabilities.

8.3.5.2 Partial Online Authorization with Communication Failure

During an EMV transaction, the terminal sends a '33.03.x' message to the POS requesting authorization. If a communication error occurs while waiting for the EMV '33.04.x' Authorization Response Message message from the POS, the terminal can accept or reject the transaction by comparing tag T95 (Transaction Verification Result) to tag T9F918709 (Terminal Action Code default, or TAG_EMV_INT_TAC_DEFAULT) and the default Issuer Action Code (IAC-Default). Tag T9F918709 specifies the acquirer's conditions which result in a transaction being rejected if it may have been approved online, but the terminal is unable to process the transaction online. The types of communication failures handled include:

- Timeout while waiting for the EMV '33.04.x' Authorization Response message.
- Receiving an EMV '33.04.x' Authorization Response message with non-EMV tag D1004 having a value of '0' indicating that the Host is not available.

8.3.5.3 EMV '33.03.x' Authorization Request Message Format

The following table provides the format for the EMV '33.03.x' Authorization Request message.

EMV '33.03.x' Authorization Request Message

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_03_EMV_AUTHORIZATION_REQUEST Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 03. = Authorization Request.
EMV Request Header.			
7	2	Alphanum	iConnectEFT Constant = P33_03_REQ_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_03_REQ_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_03_REQ_EMVH_PACKET_TYPE Packet type. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
EMV tag and data fields.			
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_03_REQ_EMV_TAG EMV Authorization Request Message tag and data field. For the format of this field, refer to EMV Tag Data Format. Refer to Non-EMV Tag Definitions for more information on non-EMV tags. For the specific tag field IDs and data transmitted for this message, refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
Search for FS character to locate the next field.			
N	1	Constant	FS – 0x1C.
Additional tag or data fields.			
Search for FS character to locate the next field.			
O	1	Constant	FS – 0x1C.

Offset	Length	Format	Description
O + 1	Variable	Alphanum	Final transaction preparation response tag or data field.
Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message. The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] character separating the different tag data. The last tag data may be followed by '[FS][ETX][LRC]' or '[FS][FS][ETX][LRC]', meaning the second [FS] character is optional.			
P	1	Constant	ETX – 0x03.
P + 1	1	Binary	LRC check character.

8.3.6 EMV '33.04.x' Authorization Response Message

The EMV '04.' Authorization Response message is sent from the POS to the terminal in response to the EMV Authorization Request message. This message includes cryptographic information which is read by the embedded microchip on the card. Refer to the below table for detailed information on this message.

The '33.04.x' message can be qualified by including the `D1011` tag. `D1011` is used to handle special cases like partial authorization and voice referral, it qualifies the Approval Tag `T8A`. A partial authorization works as such:

- For a transaction to be approved, the partial authorization requires tag `T8A` = '10'.
- Full approval comes afterwards, when tag `D1011` = '10' after receiving partial authorization.

EMV '33.04.x' Authorization Response Message

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_04_EMV_AUTHORIZATION_RESPONSE Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 04. = Authorization Response.
EMV Response Header.			

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_04_RES_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_04_RES_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_04_RES_EMVH_PACKET_TYPE Packet type. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
EMV tag and data fields.			
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_04_RES_EMV_TAG EMV Authorization Response Message tag and data field. For the format of this field, refer to EMV Tag Data Format. Refer to Non-EMV Tag Definitions for more information on non-EMV tags. For the specific tag field IDs and data transmitted for this message, refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
Search for FS character to locate the next field.			
N	1	Constant	FS – 0x1C.
Additional tag or data fields.			
Search for FS character to locate the next field.			
O	1	Constant	FS – 0x1C.
O + 1	Variable	Alphanum	Final transaction preparation response tag or data field.

Offset	Length	Format	Description
Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message. The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] character separating the different tag data. The last tag data may be followed by '[FS][ETX][LRC]' or '[FS][FS][ETX][LRC]', meaning the second [FS] character is optional.			
P	1	Constant	ETX – 0x03.
P + 1	1	Binary	LRC check character.

8.3.6.1 EMV '33.04.x' Authorization Response Error Reply Message

If any problems were detected in the authorization response message sent by the POS, then a generic error response message is returned to the POS by the terminal in order to assist the POS with identifying the reason that the authorization response is unable to be processed. Once the terminal sends this message, no further action is taken. The terminal assumes that valid messages must be sent to it in order to proceed with the transaction, with the understanding that it is the responsibility of the sender to correct the message and retransmit. Refer to the below table for a description of the EMV Authorization Response Error Reply message.

> This message should only be seen during the early development integration phase. After this phase, it is assumed the appropriate testing has been done to guarantee that only correctly constructed messages are transmitted.

EMV '33.04.x' Authorization Response Error Reply Message

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 04. = Authorization Response Error Reply.
7	2	Alphanum	iConnectEFT Constant = P33_04_RBA_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_04_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always 0. See EMV Host Interface Messages.

Offset	Length	Format	Description
10	1	Alphanum	iConnectEFT Constant = P33_04_RES_EMVH_PACKET_TYPE Packet type. Always 0. See EMV Host Interface Messages.
11	1	Constant	ETX – 0x03.
12	1	Binary	LRC check character.

8.3.6.2 Authorization Response Codes

The following table lists the Authorization Response Code values for EMV tag 8A.

Authorization Response Code Values

Authorization Response Code	Value
Online Approval	00
Online Decline	05
Offline Approved	Y1
Offline Declined	Z1
Unable to go Online, Offline Approved	Y3
Unable to go Online, Offline Declined	Z3
Referral Requested by Issuer	01
Capture Card	04

8.3.7 EMV '33.05.x' Authorization Confirmation Response Message

The EMV '33.05.x' Confirmation Response message is sent from the terminal to the POS, and contains the results from applying the authorization data to the embedded microchip on the EMV card. This includes the necessary tags for transaction completion and receipt printing. The data fields provided in this response are utilized by the POS for EMV transaction reversal purposes in the event authorization was declined by the card. Refer to the below table for detailed information on this message.

EMV '33.05.x' Authorization Confirmation Response Message

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.

Offset	Length	Format	Description
1	3	Constant	iConnectEFT Constant = M33_05_EMV_AUTHORIZATION_CONFIRMATION Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 05. = Authorization Confirmation.
7	2	Alphanum	iConnectEFT Constant = P33_05_RES_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_05_RES_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_05_RES_EMVH_PACKET_TYPE Packet type. See EMV Host Interface Messages.
Reserved for possible future confirmation header.			
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
EMV tag and data fields.			
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_05_RES_EMV_TAG EMV '33.05.x' Authorization Confirmation Response Message tag and data field. For the format of this field, refer to EMV Tag Data Format. Refer to Non-EMV Tag Definitions for more information on non-EMV tags. For the specific tag field IDs and data transmitted for this message, refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
N	1	Constant	FS – 0x1C.
Additional tag or data fields.			
Search for FS character to locate the next field.			
O	1	Constant	FS – 0x1C.
O + 1	Variable	Alphanum	Final transaction preparation response tag or data field.

Offset	Length	Format	Description
Search for FS character to locate then next field or End of Message.			
P	1	Constant	ETX – 0x03.
P + 1	1	Binary	LRC check character.

8.3.8 EMV '33.07.x' Terminal Capabilities Message

8.3.8.1 Overview of the Terminal Capabilities Request

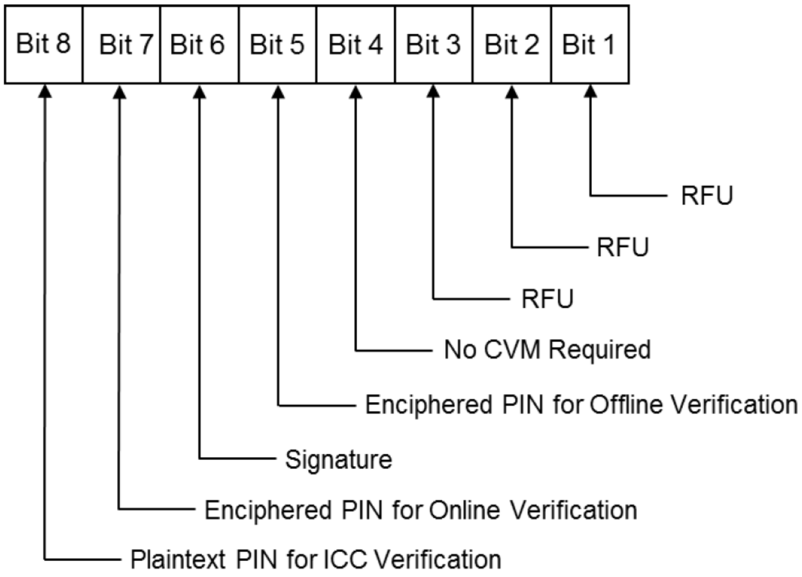
The EMV '33.07.x' Terminal Capabilities request is sent from the terminal to the POS to implement MasterCard Quick Payment Service (MasterCard QPS) and VISA Easy Payment Service (VEPS).

> To allow CVM modification by the POS, set 0019_0009 to 1.

Once the application for the payment has been selected, tag T84 (Application ID) will be populated. This tag along with tags T9F1E (Interface Device Serial Number) and D1016 (U.S. Common AID Flag) are sent from the terminal to the POS. Depending on the Application ID, transaction amount, and possibly additional criteria, the POS sends a reply indicating if PIN entry and signature card verification methods are to be bypassed for the transaction. If so, the POS returns the T9F33 Terminal Capabilities tag along with the AID in the EMV Terminal Capabilities Response Message, changing the card verification method settings. Once the transaction is completed, this tag is restored to its default value. Refer to the EMV '33.07.x' Terminal Capabilities Message Flow section for a description of the message flow.

8.3.8.2 Overview of the EMV Terminal Capabilities Response

The EMV '33.07.x' Terminal Capabilities response is sent in response to the EMV '33.07.x' Terminal Capabilities request. Depending on the Application ID, transaction amount, and possibly additional criteria, the POS sends this response message with the T9F33 Terminal Capabilities tag and the AID to the terminal in order to implement MasterCard Quick Payment Service (MasterCard QPS) and VISA Easy Payment Service (VEPS). Refer to the following illustration which describes EMV tag T9F33 byte 2 contents. Note the bit allocated for "No CVM Required."


Bit 8
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
RFU
RFU
RFU
No CVM Required
Enciphered PIN for Offline Verification
Signature
Enciphered PIN for Online Verification
Plaintext PIN for ICC Verification

EMV Tag T9F33 Byte 2 Contents

8.3.8.3 EMV '33.07.x' Terminal Capabilities Message Format

Refer to the following tables which describes the EMV '33.07.x' Terminal Capabilities Request and Response message formats.

EMV '33.07.x' Terminal Capabilities Request Message

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_07_EMV_TERMINAL_CAPABILITIES_REQ Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 07. = Terminal Capabilities Request.
EMV Request Header.			
7	2	Alphanum	iConnectEFT Constant = P33_07_RES_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_07_RES_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.

<table>
<tr><th>Offset</th><th>Length</th><th>Format</th><th>Description</th></tr>
<tr><td>10</td><td>1</td><td>Alphanum</td><td>iConnectEFT Constant = P33_07_RES_EMVH_PACKET_TYPE
Packet type. See EMV Host Interface Messages.</td></tr>
<tr><td colspan="4">Search for FS character to locate the next field.</td></tr>
<tr><td>M</td><td>1</td><td>Constant</td><td>FS – 0x1C.</td></tr>
<tr><td colspan="4">EMV tag and data fields.</td></tr>
<tr><td>M + 1</td><td>Variable</td><td>Alphanum</td><td>iConnectEFT Constant = P33_07_RES_EMV_TAG
EMV '33.07.x' Terminal Capabilities Request Message tag and data field.
For the format of this field, refer to EMV Tag Data Format.
Refer to Non-EMV Tag Definitions for more information on non-EMV tags.
For the specific tag field IDs and data transmitted for this message, refer to the following table:
<table>
<tr><th>Tag Field</th><th>Data Transmitted</th><th>Format</th></tr>
<tr><td>D1016</td><td>U.S. Common AID Flag</td><td>ASCII</td></tr>
<tr><td>T84</td><td>Application ID (AID) of application being used for payment</td><td>Hex ASCII</td></tr>
<tr><td>T9F1E</td><td>Interface Device (IFD) Serial Number</td><td>ASCII</td></tr>
</table></td></tr>
<tr><td colspan="4">Search for FS character to locate the next field.</td></tr>
<tr><td>N</td><td>1</td><td>Constant</td><td>FS – 0x1C.</td></tr>
<tr><td colspan="4">Additional tag or data fields.</td></tr>
<tr><td colspan="4">Search for FS character to locate the next field.</td></tr>
<tr><td>O</td><td>1</td><td>Constant</td><td>FS – 0x1C.</td></tr>
<tr><td>O + 1</td><td>Variable</td><td>Alphanum</td><td>Final transaction preparation response tag or data field.</td></tr>
</table>

Offset	Length	Format	Description
Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message. The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] character separating the different tag data. The last tag data may be followed by '[FS][ETX][LRC]' or '[FS][FS][ETX][LRC]', meaning the second [FS] character is optional.			
P	1	Constant	ETX – 0x03.
P + 1	1	Binary	LRC check character.

EMV '33.07.x' Terminal Capabilities Response Message

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_07_EMV_TERMINAL_CAPABILITIES_REQ Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 07. = Terminal Capabilities Response.
EMV Response Header.			
7	2	Alphanum	iConnectEFT Constant = P33_07_RES_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_07_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always 0. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_07_RES_EMVH_PACKET_TYPE Packet type. Always 0. See EMV Host Interface Messages.

<table>
<tr><th>Offset</th><th>Length</th><th>Format</th><th>Description</th></tr>
<tr><td colspan="4">Search for FS character to locate the next field.</td></tr>
<tr><td>M</td><td>1</td><td>Constant</td><td>FS – 0x1C.</td></tr>
<tr><td colspan="4">EMV tag and data fields.</td></tr>
<tr><td>M + 1</td><td>Variable</td><td>Alphanum</td><td>iConnectEFT Constant = P33_07_RES_EMV_TAG
EMV '33.07.x' Terminal Capabilities Response Message tag and data field.
For the format of this field, refer to EMV Tag Data Format.
Refer to Non-EMV Tag Definitions for more information on non-EMV tags. Also refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
For the specific tag field IDs and data transmitted in this message, refer to the following table:
<table>
<tr><th>Tag Field</th><th>Data Transmitted</th><th>Format</th></tr>
<tr><td>T84</td><td>Application ID</td><td>Hex ASCII</td></tr>
<tr><td>T9F33</td><td>Terminal Capabilities/ Configuration</td><td>Hex ASCII</td></tr>
</table>
</td></tr>
<tr><td colspan="4">Search for FS character to locate the next field.</td></tr>
<tr><td>N</td><td>1</td><td>Constant</td><td>FS – 0x1C.</td></tr>
<tr><td colspan="4">Additional tag or data fields.</td></tr>
<tr><td colspan="4">Search for FS character to locate the next field.</td></tr>
<tr><td>O</td><td>1</td><td>Constant</td><td>FS – 0x1C.</td></tr>
<tr><td>O + 1</td><td>Variable</td><td>Alphanum</td><td>Final transaction preparation response tag or data field.</td></tr>
<tr><td colspan="4">Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message.
The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] character separating the different tag data. The last tag data may be followed by '[FS][ETX][LRC]' or '[FS][FS][ETX][LRC]', meaning the second [FS] character is optional.</td></tr>
</table>

Offset	Length	Format	Description
P	1	Constant	ETX – 0x03.
P + 1	1	Binary	LRC check character.

8.3.8.4 Usage Example

The terminal has been configured to allow CVM modification by the POS by setting parameter 0019_0009 to 1, but by default, still requires signature per the middle byte of T9F33 (F0). The terminal sends the following message:

```
33.07.0000[FS]T84:07:hA0000000031010[FS]T9F1E:08:a70092873[FS]D1016:01:h00[FS]
```

The POS returns the following message to declare no CVM required by setting the second byte in tag T9F33 to 08:

```
33.07.0000[FS]T84:07:hA0000000031010[FS]T9F33:0003:hE008C8[FS]
```

The second byte in above T9F33 shows ('08') confirming the "No CVM Required" flag has been set. For this transaction, no card verification is required.

8.3.8.5 EMV '33.07.x' Terminal Capabilities Message Flow

The following is a partial representation of an EMV transaction flow illustrating the usage of the EMV '33.07.x' Terminal Capabilities message. A Terminal Capabilities Request message is sent from the terminal to the POS shortly after the card is inserted and prior to the EMV '33.02.x' Track 2 Equivalent Data Message. The terminal must wait for a Terminal Capabilities Response message from the POS before proceeding. The response message will include a modified `T9F33` Terminal Capabilities tag required to implement MasterCard Quick Payment Service (MasterCard QPS) and VISA Easy Payment Service (VEPS).

EMV Terminal Capabilities Message Flow

Sequence	Message	POS	Terminal
The EMV card is inserted in the terminal. The terminal sends an EMV Terminal Capabilities Request message to the POS.	EMV '33.07.x' Terminal Capabilities Request Message	←	
The terminal waits for the EMV Terminal Capabilities Response message before continuing with the transaction.			
The POS returns an EMV Terminal Capabilities Response message.	EMV '33.07.x' Terminal Capabilities Response Message	→	

Sequence	Message	POS	Terminal
Once the EMV Terminal Capabilities Response message is received, the transaction is resumed. "Please wait" is displayed on the screen while the card is read. The cardholder is prompted for language and application selection if the terminal is configured for manual selection. "Please wait" is again displayed, followed by the purchase amount. Track 2 equivalent data is then sent to the POS.	EMV '33.02.x' Track 2 Equivalent Data Message	←	

8.3.9 EMV '33.08.x' Set Variables Message

The POS can use the EMV '33.08.x' Set Variables message to effect permanent changes to EMV configuration settings, beyond the scope of the current transaction. This message effectively updates the settings from the `EMVCONTACT.XML` and `EMVCLESS.XML` configuration files. These new settings will serve as the defaults for subsequent transactions unless they are temporarily modified for a particular transaction (using the '33.00.x' or '33.09.x' message) or updated using a subsequent '33.08.x' message.

The following read-only variables have been added:

- Variable `600` returns the path and file name of the current EMV contact configuration.
- Variable `601` returns the path and file name of the current EMV contactless configuration.

The following table lists the EMV flags which have been added to support this feature:

EMV Flags Supporting EMV '33.08.x'

DFS Data Index	Parameter	Description
0019_0010	EMVCONTACT2.XML	This is the Contact EMV configuration file which is loaded when the terminal is booted. This parameter can be overridden using the '33.08.x' message. The name and path of the last file loaded can be retrieved using variable `600`. If left blank, the `EMVCONTACT.XML` file will be loaded. The source folder for this file is determined by parameter '0091_0031'.

DFS Data Index	Parameter	Description
0019_0011	EMVCLESS2.XML	This is the Contactless EMV configuration file which is loaded when the terminal is booted. This parameter can be overridden using the '33.08.x' message. The name and path of the last file loaded can be retrieved using variable 601. If left blank, the EMVCLESS.XML file will be loaded. The source folder for this file is determined by parameter '0091_0031'.

Using the '600' variable in the '33.08.x' message will call the most recent .XML file used for EMV contact transaction settings. Similarly, using the '601' variable will load the most recent .XML file used for contactless defaults. The new defaults will be used for future transactions, unless modified temporarily for a particular transaction (using a '33.00.x' or '33.09.x' message). A subsequent '33.08.x' message can replace these defaults. Note that changes made using the '33.08.x' message will persist only until the next reboot. The 60.x Configuration Write message can be used to effect permanent changes.

A single '33.08.x' message can update either the EMVCONTACT or EMVCLESS parameters, but not both at once. Using this message effectively updates the entire XML file by specifying a complete replacement file. After the new settings have taken place, the RBA replies with a '33.08.x' Response message.

8.3.9.1 EMV '33.08.x' Set Variables Request Message Format

The following table describes the format for the EMV '33.08.x' Set Variables Request Message.

EMV '33.08.x' Set Variables Request Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_08_EMV_SET_VARIABLES Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 08. = Set EMV Variables.
7	2	Alphanum	iConnectEFT Constant = P33_08_REQ_STATUS Status code.
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_08_REQ_EMVH_CURRENT_PACKET_NBR Current packet number. Always 0. See EMV Host Interface Messages.

Offset	Length	Format	Description
10	1	Alphanum	iConnectEFT Constant = P33_08_REQ_EMVH_PACKET_TYPE Packet type. Always 0. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.
12	1	Alphanum	iConnectEFT Constant = P33_08_REQ_TARGET_EMV_INTERFACE Target EMV Interface. • E = EMV. • C = Contactless.
Search for FS character to locate the next field.			
13	1	Constant	FS – 0x1C.
EMV Tag and Data Fields			
14	1	Alpha	iConnectEFT Constant =P33_08_REQ_FILE_LOCATION Location of the file on the terminal. • F = Parameter type file located under SYSTEM disc. • H = Parameter type file located under HOST.
15	1	Constant	: (colon) separator character.
16	Variable	Alphanum	iConnectEFT Constant =P33_08_REQ_FILE_NAME Name of the uploaded file.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
M + 1	1	Constant	ETX – 0x03.
M + 2	1	Binary	LRC check character.

8.3.9.2 EMV '33.08.x' Set Variables Response Message Format

The following table describes the format for the EMV '33.08.x' Set Variables Response Message.

EMV '33.08.x' Set Variables Response Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.

Offset	Length	Format	Description
1	3	Constant	iConnectEFT Constant = M33_08_EMV_SET_VARIABLES Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 08. = Set EMV Variables.
7	2	Alphanum	iConnectEFT Constant = P33_08_RES_STATUS Status code.
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_08_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always 0. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_08_RES_EMVH_PACKET_TYPE Packet type. Always 0. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.
12	1	Constant	ETX – 0x03.
13	1	Binary	LRC check character.

8.3.10 EMV '33.09.x' Set Tag Data Message

8.3.10.1 Overview

The EMV '33.09.x' Set Tag Data message enables the POS to change EMV or Non-EMV tags during an EMV transaction. As an example, terminal capabilities or transaction amounts may requires changes. This change only affects the current transaction values.

After the POS issues an EMV '33.00.x' Transaction Initiation message to suspend a transaction, the terminal returns a '33.01.x' EMV Status Response Message to report the current status and transaction step. The POS should then use the '33.09.x' request message to update variables and issue commands with the Command Type field, specifying tags to be changed. When completed, the terminal replies with a '33.09.x' response message indicating success status of the request. The POS should then use the '33.10.x' get EMV Tag Data message to confirm the new tag setting by requesting the tag information.

Only the values of the current transaction are affected by the EMV '33.09.x' Set Tag Data message.

8.3.10.2 EMV '33.09.x' Set Tag Data Request Message Format

The following table describes the format for the EMV '33.09.x' Set Tag Data Request Message.

The EMV '33.09.x' may only set tags with a tag ID of 2 digits or 4 digits, e.g., T5A, D1005.

EMV '33.09.x' Set Tag Data Request Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_09_EMV_SET_DATA Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 09. = Set EMV Tag Data.
7	2	Alphanum	iConnectEFT Constant = P33_09_REQ_STATUS Status code.
			EMV Request Header
9	1	Alphanum	iConnectEFT Constant = P33_09_REQ_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_09_REQ_EMVH_PACKET_TYPE Packet type. See EMV Host Interface Messages.
			Search for FS character to locate the next field.
11	1	Constant	FS – 0x1C.

Offset	Length	Format	Description
12	Variable	Alphanum	iConnectEFT Constant = P33_09_REQ_COMMAND_TYPE Command type. • T = Tag data only (next field). • J = Resume and skip built-in screen. • R = Resume and run built-in screen. • G = Clear suspend list and resume. • C = Cancel transaction. • A = Request AAC for non-full EMV transaction. • F = Fallback. The list of fallback options will include the following command type: ◦ S = Chip (smart card). ◦ C = Contactless. ◦ M = MSR. For suspended transactions, resuming will either resume where left off or skip steps based on the current step, as determined by the Command type specified.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
EMV Tag and Data Fields			
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_09_REQ_EMV_TAG EMV '33.09.x' Set Tag Data tag and data field. For the format of this field, refer to EMV Tag Data Format. Refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message. The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] character separating the different tag data. The last tag data may be followed by '[FS][ETX][LRC]' or '[FS][FS][ETX][LRC]', meaning the second [FS] character is optional.			
N	1	Constant	FS – 0x1C.
N + 1	1	Constant	ETX – 0x03.
N + 2	1	Binary	LRC check character.

8.3.10.3 EMV '33.09.x' Set Tag Data Response Message Format

The following table describes the format for the EMV '33.09.x' Set Tag Data Response Message.

EMV '33.09.x' Set Tag Data Response Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_09_EMV_SET_DATA Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 09. = Set EMV Tag Data.
7	2	Alphanum	iConnectEFT Constant = P33_09_RES_STATUS Status code.
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_09_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always 0. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_09_RES_EMVH_PACKET_TYPE Packet type. Always 0. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.
12	1	Constant	ETX – 0x03.
13	1	Binary	LRC check character.

8.3.10.4 EMV '33.09.x' Set Tag Data Message Usage Example

In the following example, the EMV '33.09.x' Set Tag Data Request message is used to overwrite the currency code as specified in tag `T5F2A`.

1. The POS sends the following request to change currency to U.S. Dollars:
   '33.09.0000[FS]R[FS]T5F2A:02:h0840[FS]T5F36:01:h02'

Per ISO 4127 (International Standard for currency), the currency code for U.S. Dollars is '08 40'.

2. The terminal responds with:

   '33.09.0000[FS]'

   This indicates that the request was executed successfully.
3. To confirm the tag values are set to the values request, the POS follows up with a '33.10.x' Get EMV Tag Data Request:

   '33.10.0000[FS]T5f2a:T57:T95:T5f36[FS]'
4. The terminal returns the following message confirming the currency change to U.S. Dollars:
   '33.10.0000[FS]T5f2a:
   02:h0840[FS]T57:13:h6510000000000174D17122011000050600000F[FS]T95:05:h0000000000[FS]T5f
   36:01:h02[FS]'

8.3.11 EMV '33.10.x' Get Tag Data Message

8.3.11.1 Overview

The EMV '33.10.x' Get Tag Data message ('33.10.x') enables the POS to request the values of EMV or non-EMV tags during an EMV transaction.

The POS will typically use the EMV '33.00.x' Transaction Initiation Message ('33.00.x') to suspend the transaction at one or more points. After issuing the '33.00.x' message, the terminal returns an EMV '33.01.x' Status Response Message to report the current status and transaction step. The POS should then use the '33.10.x' request message to retrieve tag data and issue commands instructing the RBA on how to proceed using the Command Type field. When completed, the terminal replies with a '33.10.x' response message reporting the requested data.

8.3.11.2 EMV '33.10.x' Get Tag Data Request Message Format

The following table describes the format for the EMV '33.10.x' Get Tag Data Request Message.

EMV '33.10.x' Get Tag Data Request Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_10_EMV_GET_DATA Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 10. = Get EMV Tag Data.

Offset	Length	Format	Description
EMV Request Header			
7	2	Alphanum	iConnectEFT Constant = P33_10_REQ_STATUS Status code.
9	1	Alphanum	iConnectEFT Constant = P33_10_REQ_EMVH_CURRENT_PACKET_NBR Current packet number. See EMV Host Interface Messages.
10	1	Alphanum	iConnectEFT Constant = P33_10_REQ_EMVH_PACKET_TYPE Packet type. See EMV Host Interface Messages.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.
12	Variable	Alphanum	iConnectEFT Constant = P33_10_REQ_COMMAND_TYPE Command type. • T = Tag data only (next field). • J = Resume and skip built-in screen. • R = Resume and run built-in screen. • G = Clear suspend list and resume. • C = Cancel transaction. • A = Request AAC for non-full EMV transaction. • F = Fallback. The list of fallback options will include the following command type: ◦ S = Chip (smart card). ◦ C = Contactless. ◦ M = MSR. For suspended transactions, resuming will either resume where left off or skip steps based on the current step, as determined by the Command type specified.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
EMV Tag and Data Fields			

Offset	Length	Format	Description
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_10_REQ_EMV_TAG_LIST EMV '33.10.x' Get Tag Data Request Message tag and data field. For the format of this field, refer to EMV Tag Data Format. Refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message. The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] **or** : (colon) character separating the tags requested. The last tag data may be followed by '[FS][ETX][LRC]' or '[FS][FS][ETX][LRC]', meaning the second [FS] character is optional.			
N	1	Constant	FS – 0x1C.
N + 1	1	Constant	ETX – 0x03.
N + 2	1	Binary	LRC check character.

8.3.11.3 EMV '33.10.x' Get Tag Data Response Message Format

The following table describes the format for the EMV '33.10.x' Get Tag Data Response Message.

EMV '33.10.x' Get Tag Data Response Message Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_10_EMV_GET_DATA Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 10. = Get EMV Tag Data.
7	2	Alphanum	iConnectEFT Constant = P33_10_RES_STATUS Status code.
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_10_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always 0. See EMV Host Interface Messages.

<table>
<tr><th>Offset</th><th>Length</th><th>Format</th><th>Description</th></tr>
<tr><td>10</td><td>1</td><td>Alphanum</td><td>iConnectEFT Constant = P33_10_RES_EMVH_PACKET_TYPE
Packet type. Always 0. See EMV Host Interface Messages.</td></tr>
<tr><td colspan="4">Search for FS character to locate the next field.</td></tr>
<tr><td>11</td><td>1</td><td>Constant</td><td>FS – 0x1C.</td></tr>
<tr><td>12</td><td>Variable</td><td>Alphanum</td><td>iConnectEFT Constant = P33_10_RES_EMV_TAG
Returned tag data. This field may include multiple tags. Only available tags will be included in this field.
For the format of this field, refer to EMV Tag Data Format.
Refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.</td></tr>
<tr><td colspan="4">Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message.
The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] character separating the different tag data. The last tag data may be followed by '[FS][ETX][LRC]' or '[FS][FS][ETX][LRC]', meaning the second [FS] character is optional.</td></tr>
<tr><td>M</td><td>1</td><td>Constant</td><td>FS – 0x1C.</td></tr>
<tr><td>M + 1</td><td>1</td><td>Constant</td><td>ETX – 0x03.</td></tr>
<tr><td>M + 2</td><td>1</td><td>Binary</td><td>LRC check character.</td></tr>
</table>

8.3.11.4 '33.10.x' Set EMV Data Message Usage Example

In the following example, the EMV '33.10.x' Get Tag Data Request message is used to verify the currency code change effected in tag `T5F2A`.

1. In the following example, the EMV '33.09.x' Set Tag Data Request message is used to overwrite the currency code as specified in tag `T5F2A`.
   a. The POS sends the following request to change currency to U.S. Dollars:
   '33.09.0000[FS]R[FS]T5F2A:02:h0840[FS]T5F36:01:h02'

   > Per ISO 4127 (International Standard for currency), the currency code for U.S. Dollars is '08 40'.

   b. The terminal responds with:

   '33.09.0000[FS]'

This indicates that the request was executed successfully.

c. To confirm the tag values are set to the values requested, the POS follows up with a '33.10.x' Get EMV Tag Data Request:

'33.10.0000[FS]T5f2a:T57:T95:T5f36[FS]'

d. The terminal returns the following message confirming the currency change to U.S. Dollars: '33.10.0000[FS]T5f2a: 02:h0840[FS]T57:13:h6510000000000174D17122011000050600000F[FS]T95:05:h0000000000[FS]T5f36:01:h02[FS]'

8.3.12 EMV '33.11.x' External AID Selection Notification

During the Application Selection Step, the terminal sends the EMV '33.11.x' External AID Selection Notification to convey the terminal candidate list to the POS. In both standard flow and on-demand modes, the terminal waits for the POS to send an EMV '33.12.x' External AID Selection Request to continue with the Application Selection process.

8.3.12.1 Requirements

- To allow External AID Selection by the POS, set parameter 0019_0020 to 1.
- Enable on-demand mode by setting 0007_0015 to 1.
- (Optional) Set duration to display results 0007_0001 to 0 to prevent selection screen timeout.

8.3.12.2 EMV External AID Selection Notification

Each entry of the candidate list starts with the AID tag field T4F. All the other tags for this specific entry are separated by FS (field separator character).

Each AID tag T4F begins a new candidate list entry. Refer to the message notification structure for the tags that compose an entry in the candidate list. Not all tags are present for a specific candidate entry.

Tags Sent in EMV '33.11.x' Notification per AID

EMV Tag	Description
T4F	AID
T87:b4-b1	Application priority
T87:b8	Cardholder confirmation required for application
T50	Application label
T9F12	Application preferred name
T9F11	Code table index (required for preferred name)
T5F2D	Language preference (useful for prompts)

EMV Tag	Description
T42	Issuer Identification Number (IIN)

8.3.12.2.1 EMV '33.11.x' Notification Examples:

The terminal reports a candidate list with a single AID available:

> 33.11.0000[FS]**T4F**:08:hA000000003101003[FS]T87:01:h01[FS]T50:0B:aVISA CREDIT
> [FS]T9F12:0F:h4372656469746f2064652056495341[FS]T9F11:01:h01[FS]T5F2D:
> 04:aesen[FS]T42:03:h476173[FS]

The terminal reports a candidate list with series of AIDs available:

> 33.11.0000[FS]**T4F**:08:hA000000003101003[FS]T87:01:h01[FS]T50:0B:aVISA CREDIT[FS]
> T9F12:0F:h4372656469746f2064652056495341[FS]T9F11:01:h01[FS]T5F2D:
> 04:aesen[FS]T42:03:h476173[FS]
> **T4F**:08:hA000000003101002[FS]T87:01:h04[FS]T50:0A:aVISA
> DEBIT[FS]T9F12:0E:h44656269746f2064652056495341
> [FS]T9F11:01:h01[FS]T5F2D:04:aesen[FS]T42:03:h478291[FS]**T4F**:
> 08:hA000000003101007[FS]T87:01:h85[FS]
> T50:0C:aVISA DEBIT
> 2[FS]T9F12:11:h44656269746f20646520747520706170617061[FS]T9F11:01:h01[FS]T5F2D:04:aesen
> [FS]T42:03:h478291[FS]

8.3.12.3 EMV '33.11.x' External AID Selection Message Format

The following tables describe the EMV '33.11.x' External AID Selection Request formats.

EMV '33.11.x' External AID Selection Notification Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_11_EMV_EXTERNAL_AID_SELECT_NOTIFICATION Message identifier. 33.
4	3	Constant	Subcommand identifier. • 11. = External AID Selection Request.
EMV Request Header.			
7	2	Alphanum	iConnectEFT Constant = P33_11_REQ_STATUS Status code.

Offset	Length	Format	Description
9	1	Alphanum	iConnectEFT Constant = P33_11_REQ_EMVH_CURRENT_PACKET_NBR Current packet number.
10	1	Alphanum	iConnectEFT Constant = P33_11_REQ_EMVH_PACKET_TYPE Packet type.
Search for FS character to locate the next field.			
M	1	Constant	FS – 0x1C.
EMV tag and data fields.			
M + 1	Variable	Alphanum	iConnectEFT Constant = P33_11_REQ_EMV_TAG EMV '33.11.x' External AID Selection Request Message tag and data field. For the format of this field, refer to EMV Tag Data Format. For the specific tag field IDs and data transmitted for this message, refer to EMV and Non-EMV Tags Transmitted in Host Interface Messages.
Search for FS character to locate the next field. Search for tag T4F to locate the next entry.			
N	1	Constant	FS – 0x1C.
Additional tag or data fields.			
Search for FS character to locate the next field.			
Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message. Icon The [GR] character is used to separate groups. Each group of tag fields consists one entry in the candidate list. The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] character separating the different tag data. The last tag data may be followed by '[FS][ETX][LRC]' or '[FS][FS][ETX][LRC]', meaning the second [FS] character is optional.			
P	1	Constant	ETX – 0x03.

Offset	Length	Format	Description
P + 1	1	Binary	LRC check character.

8.3.13 EMV '33.12.x' External AID Selection Request and Response

After receiving an EMV '33.11.x' External AID Selection Notification, the POS can suspend the On-Demand transaction at Step C (Application Selection) to display custom forms and prompts.

The POS then sends an EMV '33.12.x' request to:

- Prompt the cardholder to select AID (with or without cardholder confirmation)
- Choose one AID from the candidate list supplied by terminal
- Filter (remove) AIDs from the candidate list
- Inspect the candidate list and accept it

8.3.13.1 Requirements

- To allow External AID Selection by the POS, set parameter 0019_0020 to 1.
- Enable on-demand mode by setting 0007_0015 to 1.
- (Optional) Set duration to display results 0007_0001 to 0 to prevent selection screen timeout.

8.3.13.2 Example for 33.12 Requests:

The POS sends the EMV '33.12.x' request in response to the EMV '33.11.x' notification. Depending on the POS-preferred method for selecting the AID:

The POS sends an '33.12.x' message indicating no modifications to the candidate list. The terminal continues as normal with the application selection process:

```
33.12.0000[FS]0[FS]
```

The POS sends this request with a single AID which represents the Selected AID. The terminal continues with the application selection process with the selected AID received in this request:

```
33.12.0000[FS]1[FS]T4F:08:hA000000003101002[FS]
```

If an AID not in the candidate list is requested, the terminal sends a 33.05 message reporting CNSUP (Card not supported) and ends the transaction.

The POS sends this request with a filtered AID list including two AIDs (any subset of the original candidate list, but at least containing one AID) sent in the notification message. Terminal continues the application selection process with the modified candidate list.

```
33.12.0000[FS]0[FS]T4F:08:hA000000003101002[FS]T4F:07:hA000000003101007[FS]
```

8.3.13.3 EMV '33.12.x' Message in EMV On-Demand Transaction Flow with Suspend-Resume

The following shows how to use the EMV '33.11.x' and EMV '33.12.x' messages when suspending EMV transaction flow.

1. The POS:

   a. Sends an EMV '33.00.x' Transaction Initiation Message indicating suspend at Step C.
   b. Waits for a successful reply.
2. The terminal sends:
   a. An EMV '33.11.x' notification message with the application candidate list.
   b. An EMV 33.01.x Status Message to indicate flow is suspended at Step C.
3. The POS:
   a. Sends a 34.x Save and Restore State Messages to save the current state.
   b. Waits for a successful reply.
4. The terminal sends a 34.x response.
5. The POS sends On-Demand messages such as 24.x Form Entry Request (On-Demand) to call custom forms and prompts.
6. The POS:
   a. Sends a 34.R to restore the saved state.
   b. Sends an EMV '33.09.x' Set Tag Data Message specifying R or J (Resume or Skip). Application selection flow does not change based on this flag.
   c. Sends an EMV '33.12.x' request with:
      - No change,
      - Selected AID, or
      - The filtered candidate list.
   d. Waits for a successful EMV '33.12.x' response.
7. The terminal sends an EMV '33.12.x' response.

8.3.13.4 EMV '33.12.x' External AID Selection Message Formats

The following tables describe the EMV '33.12.x' External AID Selection Request and Response message formats.

EMV '33.12.x' External AID Selection Request Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_12_EMV_EXTERNAL_AID_SELECT Message identifier. 33.
4	3	Constant	Subcommand identifier. • 12. = External AID Selection
EMV Request Header.			
7	2	Alphanum	iConnectEFT Constant = P33_12_REQ_STATUS Status code.

Offset	Length	Format	Description
q	1	Alphanum	iConnectEFT Constant = P33_12_REQ_EMVH_CURRENT_PACKET_NBR Current packet number.
10	1	Alphanum	iConnectEFT Constant = P33_12_REQ_EMVH_PACKET_TYPE Packet type.
11	1	Constant	FS – 0x1C.
12	1	Numeric	EMV AID selected flag. • 1 = One selected AID. • 0 = list of AIDs or no AID changes.
13	1	Constant	FS – 0x1C.
14	Variable	Alphanum	iConnectEFT Constant = P33_12_REQ_EMV_TAG Optional. EMV '33.11.x' External AID Selection Response Message AID tag and data field. For the format of this field, refer to EMV Tag Data Format.
M	1	Constant	FS – 0x1C. Included after each AID tag and data field.
Search for FS character to locate the next AID tag, if any.			
Search for FS character to locate the next field.			
Search for FS character to locate then next field or End of Message or optional extra [FS] followed by End of Message. Icon The [GR] character is used to separate groups. Each group of tag fields consists one entry in the candidate list. The [FS] character is used to separate fields. There could be one tag field or several tag fields within the message, each with a [FS] character separating the different tag data. The last tag data may be followed by '[FS][ETX][LRC]'.			
N	1	Constant	ETX – 0x03.
N + 1	1	Binary	LRC check character.

EMV '33.12.x' External AID Selection Response Format

Offset	Length	Format	Description
0	1	Constant	STX – 0x02.
1	3	Constant	iConnectEFT Constant = M33_12_EMV_EXTERNAL_AID_SELECT Message identifier. 33.
4	3	Constant	Subcommand identifier. - 12. = External AID Selection
EMV Response Header.			
7	2	Alphanum	iConnectEFT Constant = P33_12_RES_STATUS Status code. Note that this differs from the standard status codes for EMV Host Interface Messages. - 0 = Successfully updated candidate list. - 1 = Generic failure response. For example, the POS sends the EMV '33.12.x' request while the terminal is not awaiting an EMV '33.12.x' request. - 2 = Invalid EMV '33.12.x' request format. - 3 = The AID tag in the request is not in hexadecimal format (h) or missing. - 4 = Selected AID flag mismatch. For example, EMV AID selected flag = 1 but EMV AID list does not consist of exactly one AID. The list of potential errors represented by status codes 1-4 is not exhaustive. Requests with incorrect format will be ignored until a correctly formatted message is received.
q	1	Alphanum	iConnectEFT Constant = P33_12_RES_EMVH_CURRENT_PACKET_NBR Current packet number.
10	1	Alphanum	iConnectEFT Constant = P33_12_RES_EMVH_PACKET_TYPE Packet type.
11	1	Constant	FS – 0x1C.
12	1	Constant	ETX – 0x03.

Offset	Length	Format	Description
13	1	Binary	LRC check character.

8.3.14 MAC Messages (Canada Only)

A Message Authorization Code (MAC) is a code generated from an algorithm which is used to authenticate a message. The MAC calculation request message is sent from the POS to the terminal, as a request for the terminal to calculate the MAC value of a given buffer. The MAC session key is embedded in this request message.
The terminal first loads the session key in the correct location, and then calculates the MAC value. It then sends a message back to the POS with this value for authentication.

Data must be base64 encoded before being sent in a 80.x message.

Refer to the following sections for more detail on MAC message formats:

- 80.x MAC Calculation Message Format
- 81.x MAC Verification Message Format

8.3.14.1 MAC Data

8.3.14.1.1 Overview

MAC data might include both clear data and encrypted or masked sensitive data with E2EE enabled. Clear data in the MAC data field needs to be base64-encoded. If...

- E2EE is disabled **or** MAC data does not include any encrypted or masked sensitive data, the entire MAC data field is encoded.
- MAC data includes both clear data and encrypted or masked sensitive data, each clear data section needs to be encoded separately. The data can be sent before or after encrypted or masked sensitive data, or between two encrypted or masked sensitive data sections.

8.3.14.1.2 Sensitive Data in MAC Data Field

When data encryption is enabled (via Voltage for example), the data to MAC buffer should not contain any sensitive data; however, the buffer to MAC might contain data such as the card's Track 2 or PAN, which must be used in the clear for MAC calculation operation.

For most encryption types, the application cannot decrypt the encrypted data, so it uses data from the last performed transaction to get clear sensitive data.

A list of separators are used to locate sensitive data. These separators are chosen out of the base64 characters list. The following table describes the separators:

MAC Separators

Separator	Sensitive Data
:	Card PAN data
!	Track-2 data

Separator	Sensitive Data
#	Card service code
$	Card expiry date

Example of MAC data format:

[base64 encoded data][!][encrypted track2][!][base64 encoded data]
[base64 encoded data][:][masked PAN][:][base64 encoded data]

> For Track 2 and PAN, the POS must use the corresponding encrypted data or masked data already received from the terminal during the transaction. The application compares the received data with masked or encrypted data before replacing the clear data in the buffer and then calculates the MAC. If the data comparison fails, the application returns an error.

8.3.14.1.3 MAC data for On-Guard/KME

When KME or On-Guard encryption modes are enabled, the MAC data field is now processed in the same way as other encryption modes, i.e., using the base-64 format and separators. The same method stated in the previous section are used when KME or On-Guard are enabled using the same separators: ! for Track 2 and : for PAN. The data between separators is retrieved from the Ingecrypt data buffer. The following table explains the format of Ingecrypt data buffers:

KME Encryption Data Buffers

Data	Number of ASCII characters	Description
PAN first 6 digits	6	Clear value of the first six digits of the PAN
PAN last 4 digits	4	Clear value of the last four digits of the PAN
PAN length	2	Decimal length of the PAN
PAN mod 10 check flag	1	0 = PAN Failed mod 10 check 1 = PAN Passed mod 10 check
Expiry date	4	Clear value of the Track 2 expiry date (YYMM)
Service code	3	Clear value of the Track 2 service code
Language code	1	Track 2 card language indicator
Cardholder name length	2	Decimal length of the cardholder name (not for manual entry)
Cardholder name	n	Clear value of the cardholder name (not for manual entry)

Data	Number of ASCII characters	Description
Card data encrypted flag	1	0 = clear ASCII data 1 = encrypted data
Encrypted Format Type	1	A = KME Base 24 data format
KME card data length	2	Decimal length of KME encrypted card data
KME card data field	n	KME encrypted card data if successfully read
AES PAN length	2	Decimal length of AES encrypted PAN field
AES PAN field	n	AES encrypted PAN if successfully read
LS Card data length	2	Decimal length of AES/TDES encrypted card data field
LS Card data field	n	AES/TDES encrypted card data
Extended Lang. code	1	Track 2 card language indicator first digit

On-Guard Encryption Data Buffers

Data	Number of ASCII characters	Description
PAN first 6 digits	6	Clear value of the first six digits of the PAN
PAN last 4 digits	4	Clear value of the last for digits of the PAN
PAN length	2	Decimal length of the PAN
PAN mod 10 check flag	1	0 = PAN Failed mod 10 check 1 = PAN Passed mod 10 check
Expiry date	4	Clear value of the Track 2 expiry date (YYMM)
Service code	3	Clear value of the Track 2 service code
Language code	1	Track 2 card language indicator
Cardholder name length	2	Decimal length of the cardholder name (not for manual entry)

Data	Number of ASCII characters	Description
Cardholder name	n	Clear value of the cardholder name (not for manual entry)
Card data encrypted flag	1	0 = clear ASCII data 1 = encrypted data
Encrypted Format Type	1	8 = IngeCrypt (IC) data format
IC KSN	24	DUKPT Key Serial Number (KSN)
Resoerved	4	0114
IC card data length	2	Decimal length of IC encrypted card data
IC card data field	n	IC encrypted card data if reading OK
AES PAN length	2	Decimal length of AES encrypted PAN field
AES PAN field	n	AES encrypted PAN if successfully read
LS Card data length	2	Decimal length of AES/TDES encrypted card data field
LS Card data field	n	AES/TDES encrypted card data
Extended Lang. code	1	Track 2 card language indicator first digit

The encrypted data to be put between separators will be:

- For Track 2:
  - The KME card data field will be used for KME mode.
  - The IC card data field will be used for On-Guard mode to replace the Track 2 data: `![encrypted card data]!`
- For PAN, the AES PAN field will be used for both modes to replace the PAN data: `:[encrypted PAN data]:`

8.3.15 Transaction Step List

The following table lists and identifies the steps in an EMV transaction. These steps are used as reference points when requesting status or suspending flow during an EMV transaction.

Following EMV '33.09.x' Set Tag Data Message or EMV '33.10.x' Get Tag Data Message messages (e.g. 'R' resume or 'J') skip), functionality is added to reduce timing issues deadlocking the EMV transaction following on-demand save, restore, and resume. This requires the POS wait for '33.09' or '33.10' response OS messages.

Transaction Step List

Step	Step Name	Description
A	EMV Start.	The EMV transaction has started (used for POS information, suspend not required).
B	Select language service.	Language selection is performed via the terminal (used for POS information, suspend not required).
C	Select AID service.	Application ID selection is performed via the terminal (used for POS information, suspend only required for on-demand messages between EMV '33.11.x' External AID Selection Notification request and response messages).
D	Cardholder AID confirmation.	Cardholder confirms the application selection (used for POS information, suspend not required).
E	Application final selection.	This step can be used to set EMV proprietary tags during the transaction. Suspend is required to set data.
F	Get amount application selection.	This step is used to set the transaction total amount. The transaction should be suspended during this step in order to set the transaction total amount via 13.x message. Generally speaking, tags should not be updated mid-flow.
G	Set proprietary tags at application selection.	This step be used to set EMV proprietary tags during the transaction. Suspend is required to set data. The flow must be suspended in order to enable synchronization.
H	Read application data PAN ready (to stop for non-full EMV).	This step is used to stop a non-full EMV transaction. The RBA should be suspended in this case and the EMV '33.09.x' Set Tag Data message can be used with command "A" (Request AAC for non-full EMV transaction).
I	Set payment type.	Non-EMV step to set the payment type.
J	Get cash back amount.	This is a non-EMV step used to get the cashback amount.
K	Read application data change amount.	This step may be used to change the transaction total amount via 13.x message. Again, tags should generally not be individually manually updated mid-flow.
L	Amount confirmation.	Non-EMV step to confirm the amount.
M	Account selection.	Non-EMV step to select the account type (e.g., checking, saving).
N	Offline PIN entry.	This step is used for Offline entry; the cardholder must enter their PIN (used for POS information, suspend not required).
O	Online PIN entry.	This step is used for Online entry; the cardholder must enter their PIN (used for POS information, suspend not required).

Step	Step Name	Description
P	Last transaction data request.	This step is used to bypass the last EMV transaction data to the RBA. As an example, to pass the last transaction data in the same batch performed using the same card. The RBA should be suspended in this case and the EMV '33.09.x' Set Tag Data message can be used with command "T" (Tag data only) and corresponding tags (`T9C`, `T9F21`, `T9A`, `T9F8417`, `T9F04`, `T81`, `T9F8416`).
Q	Terminal action analysis (to stop for non-full EMV).	This step is used to stop a non-full EMV transaction. The RBA should be suspended in this case and the EMV '33.09.x' Set Tag Data message can be used with command "A" (Request AAC for non-full EMV transaction).
R	Online authorization response in progress.	This suspend step is used after the EMV '33.03.x' Authorization Request Message is sent and before waiting on the EMV '33.04.x' Authorization Response Message, allowing on-demand control. A good use case for this step is when users may want to send online on-demand PIN retries without having to generate a new cryptogram for each iteration.
S	EMV stop.	Transaction has ended (used for POS information, suspend not required).

Step	Step Name	Description
U	Completion Status.	End of Transaction control. When suspended, allows users to: • ignore a card decline (merchant stand in etc) and have the terminal display approval • display custom messages instead of or alongside approved/declined • provide custom display and/or beeps (via 51.x Beep On-Demand Message) to instruct cardholders to remove their card Resuming during this step (either with or without on-demand behavior) will return to displaying standard-flow transaction results. Skipping will proceed to the remove card state (if still inserted) and then end of transaction state. Do not use Reset messages when suspended at step 'U'. Instead, use the EMV '33.09.x' Set Tag Data Message message to resume or skip; after this, Reset messages can be used. In a specific scenario, a blank form will end up displaying until a reset message is sent: If the card is removed **after** the '33.09J' (skip) and Please Remove Card message is displayed following the use of custom display(s) at Step U, a blank form will be displayed until a reset message is received from the POS. This blank form could be modified using the 70.x Update Form Element Message if the form includes dynamic prompts.

EMV '33.09.x' Set Tag Data Message should only be sent during EMV transaction suspend steps. Otherwise, the message commands may be silently ignored. The same is true for EMV '33.10.x' Get Tag Data Message if sent with any commands other than 'T' (return tag data only).
Currently, both messages can resume EMV transactions at suspend steps, even for commands other than 'R' (Resume).

8.3.16 EMV Tag Data Format

Ingenico Proprietary EMV Tag Data Format follows the TLV Format (T=Tag; L=Length; V=Value). All the three fields: tag, length and value are in ASCII (the standard speaks about BER encoding).

The format of EMV tag data included in the 33.x messages is described in the following table.

EMV Tag Data Format as Included in 33.x Message

Length	Content
1	"T"

Length	Content
2 - 4	Hex ASCII tag ID
1	":"
2 - 4	Hex ASCII tag length
1	":"
1	Tag data format specifier ("a" or "h") where • "a" specifies ASCII. • "h" specifies hex ASCII.
1	ASCII or hex ASCII tag value
1	Field separator character (i.e., FS = 0x1C)

Non-EMV tag fields are prefixed with the descriptor "D". The rest of the syntax for the tag, however, follows the EMV tag format as shown in the preceding table titled "EMV Tag Data Format as Included in 33.x Message." Tag numbers for non-EMV tags start at 0x1000. Refer to the following examples:

Example 1: EMV tag in hex ASCII format with a tag ID of '0082' and length of 4 bytes:

T82:04:hxxxxxxxx<FS>

- T   - L -V   -

where "xx" represents a single byte.

Example 2: EMV tag in ASCII format with a tag ID of '82' and length of 4 bytes:

T82:04:axxxx<FS>

- T - L-V   -

where "x" represents a single byte.

Example 3: Non-EMV tag in hex ASCII format with a tag ID of '1000' and length of 4 bytes:

D1000:04:hxxxxxxxx<FS>

where "xx" represents a single byte.

Example 4: Non-EMV tag in ASCII format with a tag ID of '1001' and length of 1 byte:

D1001:01:ax<FS>

where "x" represents a single byte.

If the tag data format is "h" (specifying hex ASCII) then the tag data must be represented in an even number of hex ASCII digits. Wrapping of tag data across consecutive packets is permitted. In this event, the tag ID and length specified in the first packet are assumed for the data in subsequent packets.

8.3.17 EMV and Non-EMV Tags Transmitted in Host Interface Messages

The following table describes the EMV and non-EMV tags transmitted in the host interface messages:

- EMV 33.02.x Track 2 Equivalent Data Message
- EMV 33.03.x Authorization Request Message
- EMV 33.04.x Authorization Response Message
- EMV 33.05.x Authorization Confirmation Response Message
- EMV 33.11.x External AID Selection Notification

33.05.x Abbreviation Key

Abbreviation	Transaction Type
P	Purchase
R	Refund
VP	Void Purchase
VR	Void Refund
X	All of the above

EMV and Non-EMV Tags Transmitted in Host Interface Messages

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
T42	Issuer Identification Number	Hex-ASCII					X
T4F	Application Identifier (AID)	Hex-ASCII	X	X		X	X
T50	Application Label	ASCII	X	X		X	X
T57	Track 2 Equivalent Data	Hex-ASCII	X	X		X	
T5A	Primary Account Number	Hex-ASCII	X				
T71	Issuer script template 1				X		
T72	Issuer script template 2				X		

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
T82	Application Interchange Profile	Hex-ASCII		X		X	
T84	Dedicated File (DF) Name	Hex-ASCII	X	X		X	X
T87	Application Priority Indicator	Hex-ASCII					X
T8A	Authorization Response Code	ASCII			X	X	
T8E	Card Holder Verification Method (CVM) List	Hex-ASCII				X	
T91	Issuer Authentication Data	Hex-ASCII			X	X	
T95	Terminal Verification Results	Hex-ASCII		X		X	
T9A	Transaction Date	Hex-ASCII		X		X	
T9B	Transaction Status Information	Hex-ASCII		X		X	
T9C	Transaction Type	Hex-ASCII		X		X	
T5F20	Cardholder name	ASCII	X				
T5F24	Expiry Date	Hex-ASCII	X	X			
T5F28	Issuer Country Code, three-digit numeric	Hex-ASCII	X			X	

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
T5F2A	Transaction Currency Code	Hex-ASCII		X		X	
T5F2D	Preferred languages	ASCII	X				X
T5F30	Service Code	Hex-ASCII	X				
T5F34	Application PAN Sequence Number	Hex-ASCII	X	X		X	
T5F54	Bank Identifier Code (BIC)	Hex-ASCII	X			X	
T5F55	Issuer Country Code, two-digit alpha	ASCII	X				X
T5F56	Issuer Country Code, three-digit alpha	ASCII	X				
T9F02	Amount, Authorized (Numeric)	Hex-ASCII		X		X	
T9F03	Amount, Other (Numeric)	Hex-ASCII		X		X	
T9F06	Application ID Terminal	Hex-ASCII		X		X	
T9F07	Application Usage Control	Hex-ASCII	X			X	
T9F08	Application Version Number (ICC)	Hex-ASCII				X	

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
T9F09	Application Version Number (Terminal)	Hex-ASCII		X		X	
T9F0B	Cardholder Name Extended	ASCII	X				
T9F0D	Issuer Action Code Default	Hex-ASCII				X	
T9F0E	Issuer Action Code Denial	Hex-ASCII				X	
T9F0F	Issuer Action Code Online	Hex-ASCII				X	
T9F10	Issuer Application Data	Hex-ASCII		X		X	
T9F11	Issuer Code Table Index	Hex-ASCII	X	X		X	X
T9F12	Application Preferred Name	Hex-ASCII	X	X		X	X
T9F14	Lower Consecutive Offline Limit	Hex-ASCII		X			
T9F17	PIN Try Count	Hex-ASCII				X	
T9F1A	Terminal Country Code	Hex-ASCII	X	X		X	
T9F1B	Terminal Floor Limit	Hex-ASCII	X	X		X	
T9F1E	Interface Device (IFD) Serial Number	ASCII	X	X		X	
T9F1F	Track 1 Discretionary Data	ASCII	X	X		X	

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
T9F20	Track 2 Discretionary Data	Hex-ASCII	X	X		X	
T9F21	Transaction Time.	Hex-ASCII		X		X	
T9F26	Application Cryptogram (AC).	Hex-ASCII		X		X	
T9F27	Cryptogram Information Data (CID).	Hex-ASCII		X		X	
T9F33	Terminal Capabilities.	Hex-ASCII		X		X	
T9F34*	Cardholder Verification method (CVM) Results.	Hex-ASCII		X		X	
T9F35	Terminal Type.	Hex-ASCII		X		X	
T9F36	Application Transaction Counter (ATC).	Hex-ASCII		X		X	
T9F37	Unpredictable Number.	Hex-ASCII		X		X	
T9F39	POS Entry Mode.	Hex-ASCII		X		X	
T9F40	Additional Terminal Capabilities.	Hex-ASCII		X		X	
T9F41	Transaction Sequence Counter.	Hex-ASCII		X		X	
T9F42	Application Currency Code	Hex-ASCII	X			X	

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
T9F51	Application Currency Code/DRDOL	Hex-ASCII	X			X	
T9F53	Transaction Category Code (VISA only).	Hex-ASCII		X		X	
T9F5B	Transaction Category Code (VISA only).	Hex-ASCII				X	
T9F5D	Available Offline Spending Amount (AOSA).	Hex-ASCII	X	X		X	
T9F66	Terminal Transaction Qualifiers (TTQ)	Hex-ASCII	X	X		P, R	
T9F67	(Amex spec) - NATC (Track2) / MSD Offset.	Hex-ASCII	X	X		X	
T9F6C	Card Transaction Qualifiers (CTQ).	Hex-ASCII	X	X		X	
T9F6D	EMV Proprietary tag. See brand specifications for details per brand.	Hex-ASCII	X	X		X	
T9F6E	Third Party Data.	Hex-ASCII	X	X		X	

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
T9F71	Protected Data Envelope 2 & Mobile CVM Results	Hex-ASCII	X	X		X	
T9F7C	Customer Exclusive Data (CED) & Merchant Custom Data.	Hex-ASCII	X	X		X	
TDF03	Terminal Action Code Default.	Hex-ASCII				X	
TDF04	Terminal Action Code Denial.	Hex-ASCII				X	
TDF05	Terminal Action Code Online.	Hex-ASCII				X	
TDF11	Issuer Script Results.	Hex-ASCII				X	
D1000	Account Type (Interac only).	ASCII		X		X	
D1001	PIN Entry Required Flag.	ASCII		X		X	
D1002	Signature required Flag.	ASCII	X	X		X	
D1003	Confirmation response Code.	ASCII				X	
D1004	Host response available.	Hex-ASCII			X		
D1005	Transaction Type.	ASCII		X		X	

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
D100E	Selected Transaction Language.	ASCII		X		X	
D100F	PIN Entry Success Flag. (Required for Offline PIN entry only)	ASCII		X		X	
D1010	Error Response Code.	ASCII				X	
D1011	Special Case Authorization.	Hex-ASCII				X	
D1012	Contactless Transaction Outcome.	ASCII		X		X	
D1013	Contactless Profile Used.	ASCII				X	
D1014	Card Payment Type.	ASCII					
D1015	Suncor.	ASCII				X	
D1016	U.S. Common AID Flag.	ASCII					
D1017	MSR Track 1.	Hex-ASCII				X	
D1018	MSR Track 2.	Hex-ASCII				X	
D1019	MSR Track 3.	Hex-ASCII				X	
D101A	Fallback to MSR Status.	Hex-ASCII				X	
D101B	Contactless Online PIN CVM Flag.	ASCII	X	X		X	
D101C	Contactless NoCVM Flag.	ASCII	X	X		X	

Tag ID	Data Transmitted	Format	33.02.x	33.03.x	33.04.x	33.05.x	33.11.x
D101D	Cless Mobile CVM performed	ASCII	X	X		X	
D101E	Cless Mobile CVM results	ASCII	X	X		X	
D1020	Merchant Coupon Data	Hex-ASCII	X	X		P, R	
DFF1E	Encrypted Track 1 Data	ASCII	X	X		X	
DFF1F	Encrypted Track 2 Data	ASCII	X	X		X	
DFF20	ETB	ASCII	X	X		P	
DFF21	Encrypted Track 3 Data	ASCII	X	X		X	
D9000	Card payment Type	ASCII		X		P, R	
D9001	Card Entry Mode	ASCII		X		P, R	

*Paypass, ExpressPay, and Interac Flash are the only kernels that use tag T9F34.

This is a partial list. A complete list of tags cannot be provided here because it is dependent on the host and issuer data required by the card and application. More EMV tag definitions can be found at http://www.emvlab.org/emvtags/all/.

8.3.18 Non-EMV Tag Definitions

Non-EMV tags are used to exchange information not defined by EMV standards between the POS and the terminal. Non-EMV tags have the following characteristics:

- They are proprietary and defined by Ingenico.
- Tag fields start with the character *D*, rather than the standard *T* character used for EMV tags.
- These tags are primitive data objects in TLV format.

The data values included in these tags are useful when performing EMV transactions. Non-EMV tag numbers are intentionally non-compliant with EMV standards to avoid conflict with future EMV tags.

Non-EMV tags are included in the authorization and confirmation messages for retail application EMV transactions. They serve as prime indicators for the EMV transaction flow.

8.3.18.1 Examples

- Tag `D1003` contains the confirmation response code which is included in the confirmation response message to determine the final decision on a transaction. When the value for this tag is *E* (error or incomplete), tag `D1010` is provided in the confirmation response to indicate the type of error which as occurred (such as, Authorization Request Sent Failed, Card Data Invalid).
- Tag `D1005` is issued in both the authorization request and the confirmation response to indicate the type of transaction (purchase or refund) requesting authorization.

The following tabledescribes non-EMV tags:

Non-EMV Tag Definitions

Tag Number	Tag Name	Non-EMV Tag Definition and Values
D1000	Interac Account Type	Account type selected for an Interac debit transaction. • 0 = Checking. • 1 = Savings.
D1001	PIN Entry Required Flag	EMV CVM indication of whether PIN entry is required for transaction. • 0 = Not required. • 1 = Required. • U = Error, data is unavailable. Only available after cardholder verification has occurred on a full EMV transaction.
D1002	Signature Required Flag	EMV CVM indication of whether signature is required for transaction. • 0 = Not required. • 1 = Required. • U = Error, data is unavailable. Only available after card holder verification has occurred on a full EMV transaction.

Tag Number	Tag Name	Non-EMV Tag Definition and Values
D1003	Confirmation Response Code	Confirmation response code. • A = Approve (purchase or refund). • D = Decline (purchase or refund). • C = Completed (refund). • E = Error or incompletion (purchase or refund). • F = Fallback to MSR.
D1004	POS Response Available	Tag data to indicate if the POS cannot communicate with the Host. • 0 = Host response is not available. • 1 = Host response is available. Typically this tag with a value of ‘0’ will be sent by the POS in cases where the Host is not available or when it times out. In the case where the Host is available and a Host response returns then this tag is not expected, but if returned in the Host tag list the value of this tag must be '1'.
D1005	Transaction Type	Tag data indicating which transaction type is requesting authorization. • 00 = Purchase. • 01 = Refund. This tag is issued in both the authorization request and the confirmation response. This must be identical to either the transaction type specified in the transaction initiation command or to the refund situation specified through a negative amount or to the default situation of a purchase if neither of the above to transaction criteria have been issued.
D100E	User Language	Terminal language selected for EMV transactions. • EN = English. • FR = French. • ES = Spanish (not used in Canada). This ISO language identifier is 2 bytes in ASCII format.

Tag Number	Tag Name	Non-EMV Tag Definition and Values
D100F	PIN Entry Successful Flag	EMV CV indication of PIN entry success in the current transaction. Required for offline PIN input only. - 0 = Not successful. Could be entry was cancelled, bypassed, timed out, or otherwise unavailable. - 1 = Successful. - U = Error, data is unavailable. This value is only available after cardholder verification has occurred on a full EMV transaction.

<table>
<tr><th>Tag Number</th><th>Tag Name</th><th>Non-EMV Tag Definition and Values</th></tr>
<tr><td>D1010</td><td>Error Response Code</td><td>Error response code provided in the confirmation response.
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>APBLK</td><td>Application Blocked.</td></tr>
<tr><td>ARRT</td><td>Authorization Response Received Timeout – the terminal did not receive a '33.04' authorization response message from the POS.</td></tr>
<tr><td>ARSF</td><td>Authorization Request Sent Failed – the terminal could not send the authorization request message.</td></tr>
<tr><td>CABLK</td><td>Card Blocked.</td></tr>
<tr><td>CAN</td><td>Transaction Cancelled.</td></tr>
<tr><td>CDIV</td><td>Card Data Invalid.</td></tr>
<tr><td>CDIVN</td><td>Card Data Invalid but EMV fallback not permitted for Interac transaction.</td></tr>
<tr><td>CEXP</td><td>Card/Application is Expired.</td></tr>
<tr><td>CNSUP</td><td>Card Not Supported – there is no matching AID between the card and the terminal.</td></tr>
<tr><td>CRPRE</td><td>Card Removed Prematurely.</td></tr>
<tr><td>CRSF</td><td>Confirmation Response Sent Failed – the terminal could not send the '33.05' authorization confirmation response.
Due to the failure, the POS does not see this code in a '33.05' message. However, POS can use status message '33.01' flag 15 to determine whether the confirmation response failed to send, or if it was sent successfully.</td></tr>
<tr><td>FATAL</td><td>Fatal Error – a fatal error happened and no fallback is required. This code is used when a non-full-EMV transaction is performed for a non-Refund transaction, e.g., a purchase transaction performed as a non-full-EMV transaction. This code is also used for internal errors, when the EMV Library returns a non-processed status to the RBA, though this should never happen.</td></tr>
<tr><td>T2CF</td><td>Track 2 Consistency Check Failed.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Tag Number</th><th>Tag Name</th><th>Non-EMV Tag Definition and Values</th></tr>
<tr><td></td><td></td><td><table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>TPSF</td><td>Transaction Preparation Sent Failed.</td></tr>
<tr><td>UITMO</td><td>User Interface Timeout.</td></tr>
</table></td></tr>
</table>

Tag Number	Tag Name	Non-EMV Tag Definition and Values
D1011	Special Case Authorization	D1011 is used to handle special cases like partial authorization and voice referral, it qualifies the Approval Tag `T8A.` D1011 can force a transaction to decline or approve regardless of host decision **or** display transaction declined/approved regardless of EMV card decision: • 00 = Approved. • 01 = Voice Referral. • 05 = Declined. • 10 = Partial Authorization. • 55 = Bad PIN Value. POS may send D1011 = 00 = approved to approve most/all transactions regardless of risk or liability (e.g. fast food chain that prioritizes high-volume customer checkout vs. low-risk fraud). • Terminal displays "Approved" • Can send with or without host response and can even override card decision: ◦ If card decision also approves, ▪ `D1003` = A (approved) ▪ `T9F27` = 40 ◦ If card decision declines, ▪ `D1003` = D (declined) ▪ `T9F27` = 00 The terminal may optionally send `D1011` = '05' (declined) to decline transaction regardless of host response (e.g. to decline partial approvals). • Terminal displays "Declined". • Card decision will also decline if: ◦ `D1003` = D (declined) ◦ `T9F27` = 00 The terminalmight not send D1011 and instead allow T8A`,` host authorization response, and/or card decision to determine the final outcome of a transaction. If an invalid online PIN is entered, the POS includes tag D1011 with a value of '55' in the EMV 33.04.x Authorization Response Message.

Tag Number	Tag Name	Non-EMV Tag Definition and Values
D1012	Contactless Transaction Outcome	This tag provides the POS with the contactless transaction outcome. • 01 = Offline approved. • 02 = Online approved. • 03 = Online requested. • 04 = Normal decline. • 05 = Use chip interface. • 06 = Enter pass code in mobile and re-tap.
D1013	Contactless Profile Used	This tag provides the contactless profile used for the transaction. • E = EMV interface. • M = Magstripe (MSD) interface.
D1014	Card Payment Type	This tag provides the terminal with the payment type. this is generally used then the POS skips the SET_PAYMENT_TYPE transaction step. • A = Debit. • B = Credit.
D1015	Suncor	reserved for Suncor
D1016	US Common Debit flag	This tag indicates if transaction is related to U.S. common debit. • "1" - Related to U.S. Common Debit • "0" - Otherwise
D1017	MSR Track 1	Used to relay Track 1 data during on-demand fallback to MSR.
D1018	MSR Track 2	Used to relay Track 2 data during on-demand fallback to MSR.
D1019	MSR Track 3	Used to relay Track 3 data during on-demand fallback to MSR.
D101A	Fallback to MSR Status.	Indicates the status of fallback to MSR. • FLBKOK = Good card read at fallback. • FLBKERR = Bad card read at fallback.
D101B	Contactless online PIN CVM flag	This tag indicates an online PIN CVM. • “1” – required • “0” – not required

Tag Number	Tag Name	Non-EMV Tag Definition and Values
D101C	Contactless No CVM flag	Indicates if no CVM is required in contactless transactions for all card schemes supported. • "0" = CVM required. • "1" = No CVM required.
D101D	Mobile CVM performed	Indicates if Mobile CVM was performed. • '0' - Mobile CVM not performed. • '1' - Mobile CVM performed.
D101E	Mobile CVM Results	Non EMV D-Tag indicates Mobile CVM Results. • '0' - Unknown • '1'- Failed • '2' - Successful • '3' - Blocked
D9000	Card Payment Type	Card payment type. • A = Debit. • B = Credit.
D9001	Card Entry Mode	Card entry mode. • C = Chip entry. • D = Contactless EMV entry.

8.4 EMV Transaction Flow

This section provides more in-depth information on the flow of EMV transactions, showing the sequence and exchange of messages between the terminal and Point of Sale system (POS). For simplicity, only the key EMV transaction flows are listed.

When the application is online and idle it sets the next transaction as an EMV purchase if EMV is enabled. When a chip card is entered into the terminal, the purchase transaction immediately starts. The transaction types discussed in this section are:

- EMV Purchase Transaction Flow
- EMV Contactless Transaction Flow
- EMV Full Refund Transaction Flow
- EMV Partial Refund Normal Transaction Flow
- EMV Partial Refund On-Demand Transaction Flow
- EMV Contactless On-Demand Full Refund Transaction Flow
- Cancelling an EMV Transaction

- MSD Contactless Transaction Flow
- EMV Transaction Flows with Application Selection

> For EMV Account Verify/Balance Inquiry transactions, the flow is the same as a purchase, but with a $0 amount. Tag `T9C` is set to '30' for such transactions.

Also refer to the table of Non-EMV Tag Definitions.

> For EMV, different fonts will be displayed in the “Confirm Application” Prompt (e.g., Confirm Application DISCOVER) in `Prompt.XML`. This is because the application name "DISCOVER" is obtained from EMV card Tag `T50`, and the application simply displays this as is from the card.

8.4.1 EMV Purchase Transaction Flow

EMV purchase transactions are initiated through a 14.x Transaction Type message from the POS, which is followed by a 13.x Amount Message indicating the amount of the purchase or refund. If cashback is included it will be appended to the 13.x message as the last data field following a group separator. The 14.x message identifies the transaction as a purchase in this case. Once this message is received by the terminal, the chip card is inserted (or tapped in the case of a contactless card) and the terminal configures the transaction by selecting the language and application, or by prompting the user to select these when not configured for automatic selection.

Once the application and language selections have been made, the terminal will transmit the Track 2 equivalent (stored in the card microchip) to the POS using the EMV '33.02.x' Track 2 Equivalent Data message.

Depending on the type of transaction, the user may be prompted for a cash back amount and PIN entry. If the cardholder requires cash back then the appropriate screen is displayed, with the user prompted to select and confirm the cash back amount. A 04.x Payment Type Response message with the cash back amount is always returned to the POS in response to a 04.x Payment Type Request message.

Following the payment type message, a 13.x message with the final transaction amount must be sent to the terminal in order to complete the transaction. This message includes the cash back amount. The cardholder is then prompted to confirm the transaction amount including the cash back amount. For Interac debit transactions, the cardholder will be prompted to select the account type (checquing or savings).

Card authentication and transaction processing (approval or decline) follow with a confirmation message. An EMV '33.03.x' Authorization Request message is sent to the POS, and the terminal then waits for the EMV '33.04.x' Authorization Response message. Following the authorization response being received, the terminal determines if the response indicates approval or decline. An EMV '33.05.x' Authorization Confirmation Response message is then sent to the POS to inform the POS of the transaction state and to return transaction tag information needed for printing.

> For EMV Account Verify/Balance Inquiry transactions, the flow is the same as a purchase, but with a $0 amount. Tag `T9C` is set to '30' for such transactions.

Refer to the below table for an illustration of the EMV purchase transaction sequence.

EMV Purchase Transaction Flow

Sequence	Message	POS	Terminal
Transaction type and final amount are sent to the terminal.	14.x Set Transaction Type - 14.01 = Purchase. - 14.03 = Refund.	→	
	13.x Amount Message	→	
The EMV transaction is initiated as the card is inserted. "Please wait" is displayed on the screen while the card is read. The cardholder is prompted for language and application selection if the terminal is configured for manual selection. "Please wait" is again displayed, followed by the purchase amount. Track 2 equivalent data is then sent to the POS.	EMV '33.02.x' Track 2 Equivalent Data Message	←	
The POS requests the payment type.	04.x Set Payment Type Request	→	
The terminal responds with the payment type and cash back amount.	04.x Set Payment Type Response	←	
The transaction amount is sent to the terminal indicating the final amount and cash back amount. If cashback is included it will be appended to the 13.x message following a group separator.	13.x Amount Message	→	

Sequence	Message	POS	Terminal
The cardholder is prompted to confirm the purchase and cash back amounts, and enter their PIN (if required). An authorization request containing the necessary cryptographic information is sent to the POS. EMV transactions now utilize the Amount Verify flag. Depending on the setting of this flag, the merchant can suppress the "Amount OK?" screen and directly prompt the cardholder for amount verification. The transaction amount can be displayed on the PIN entry or signature screen. When the cardholder enters their PIN or signs, approval of the amount is implied.	EMV '33.03.x' Authorization Request Message	←	
The POS responds with an authorization response containing information to be read by the EMV card.	EMV '33.04.x' Authorization Response Message	→	
The transaction is approved or denied, based on the authorization response content. A confirmation response is sent to the POS.	EMV '33.05.x' Authorization Confirmation Response Message	←	
Return to idle state.			

8.4.2 EMV Contactless Transaction Flow

The POS typically sends two messages to the terminal at the start of a purchase or refund transaction:

- 14.x Transaction Type message
- 13.x Amount message

If cashback is included, it is appended to the 13.x message as the last data field following a group separator. The 14.x Transaction Type message identifies the transaction as a purchase (14.01) or refund (14.03). While the 14.x Transaction Type message is recommended, it is not required. If this message is not sent, the transaction defaults to a purchase transaction. The 13.x Amount Message is required, as this message specifies the amount of the purchase or refund.

To configure the terminal for contactless EMV transactions, the Contactless Reader Mode parameter (0008_0001) must be set to 9 which will enable contactless mode for EMV. Note that in on-demand mode, The POS must send the amount via the 13.x message before enabling the contactless reader via the 23.x message. Once the 13.x Amount message has been received and the contactless reader is enabled, the terminal display will be updated to include the tap option. The EMV transaction will then be initiated when the contactless card is tapped.

The next step in the process is to tap the EMV card. If contactless is enabled in the configuration file, then subsequent transactions are already configured for EMV contactless each time the application enters the online idle state. Accordingly, an EMV contactless transaction will be automatically initiated immediately when the card is tapped (read by the contactless card reader). If the card is tapped before the terminal receives the 13.x amount message from the POS, then the terminal will display an error message indicating that no amount has been entered.

When the EMV card is detected, there is no further cardholder input. An EMV 33.02.x Track 2-Equivalent Data message, sent by the terminal to the POS, conveys the necessary information to initiate the transaction. It is followed by an EMV 33.03.x Authorization Request message, which contains the cryptographic information used by the POS to authorize the transaction.

When the EMV 33.03.x Authorization Request message is received by the POS, it responds with an EMV 33.04.x Authorization Response message, which contains encrypted data used by the embedded microchip in the card to approve or decline the transaction. After the decision is made, an EMV 33.05.x Authorization Confirmation Response message is sent to the POS. This message includes the tag information used by the POS to print the transaction receipt.

EMV Contactless Transaction Flow

Sequence	Message	POS	Terminal
The transaction type and transaction amount are sent to the terminal. Although the 14.x Transaction Type message is recommended, it is not required. If it is not sent, the transaction will default to a purchase transaction. The contactless reader is not enabled until the 13.x Amount message is received. If a card is tapped before the contactless reader is enabled, the terminal displays an error message. Note that in on-demand mode, the POS must send the amount via the 13.x message before enabling the contactless reader via the 23.x message. If cash back is included, it is appended to the 13.x message, following a group separator. The transactions cannot proceed without an amount value set. The application accepts $0 amounts, but certain EMV cards might decline the transaction offline, preventing the transaction from authorizing online.	14.x Set Transaction Type • 14.01 = Purchase. • 14.03 = Refund.	→	
	13.x Amount Message	→	

Sequence	Message	POS	Terminal
After the 13.x Amount message is received, and the contactless reader is enabled, the EMV transaction can be initiated when the contactless card is tapped. Track 2-equivalent data is sent to the POS. An authorization request provides the POS with the information required to authorize the transaction.	EMV 33.02.x Track 2 Equivalent Data Message	←	
	EMV 33.03.x Authorization Request Message	←	
An authorization response is returned to the terminal with encrypted information read by the EMV card embedded microchip.	EMV 33.04.x Authorization Response Message	→	
The transaction is approved or denied based on the authorization response content. A confirmation response is sent to the POS.	EMV 33.05.x Authorization Confirmation Response Message	←	
Return to idle state.			

8.4.3 EMV Full Refund Transaction Flow

EMV refund transactions are processed for cards which fully support EMV transactions. In order to proceed with an EMV refund transaction, the card must be inserted and a '14.03' Transaction Type (refund) message (or '14.01' Transaction Type message followed by a 13.x Amount Message with a negative amount) must be received by the terminal. If the card is inserted before this message is received, then the transaction defaults to an EMV purchase transaction. If this message is received after the card is inserted, then the purchase transaction will be cancelled and a refund transaction will be initiated.

Once the refund transaction is initiated with the card detected, the cardholder may be prompted to select the language and application if not configured for auto-selection in the configuration file. If the auto-selection flags are enabled then the terminal will select the language and application based on highest priority.

With the language and application selected and the transaction amount sent, the terminal will send an EMV '33.02.x' Track 2 Equivalent Data message. As a future option, the POS may request this information via an EMV '33.02.x' Track 2 Equivalent Data Request message, but it is not recommended at this time.

A 04.x Set Payment Type Request message follows the EMV '33.02.x' Track 2 Equivalent Data Message message. The terminal responds with a 04.x: response message confirming the payment type request (e.g., Debit, Credit). In order to process the transaction as full EMV refund, the EMV Refund Option flag must be set to '1'. The EMV

Refund Option flag is set using the 60.x Configuration Write message to modify the cards.dat file before the start of the transaction.

A second 13.x Amount Message with the final transaction amount is then sent to the terminal. With the final transaction information received from the POS, the cardholder may be prompted to confirm the refund amount depending on the setting of the Verify Amount flag. The cardholder may additionally be prompted for an account selection (chequing or savings). The cardholder is then prompted for PIN entry if required.

The terminal sends an EMV '33.03.x' Authorization Request Message message to the POS, which will return the EMV '33.04.x' Authorization Response Message message. An EMV '33.05.x' Authorization Confirmation Response Message message is then returned to the POS confirming the refund transaction state and including the necessary tag information for printing. Included in the '33.05.x' message will be tag D1003 which will indicate the approval status ("A" for Approved and "D" for Declined) or D1004 = 0 if the Host response is not available.
The terminal will then display the transaction status as "Approved" or "Declined" based on this tag value. The cardholder is then prompted to remove their card. When the card is removed, the terminal sends a 09.x Card Status Message message to the POS indicating that the card has been removed.

Refer to the below table for an illustration of the EMV refund transaction sequence.

EMV Refund Transaction Flow

Sequence	Message	POS	Terminal
Transaction type (refund) and refund amount are sent to the terminal.	14.x Set Transaction Type • 14.03 = Refund.	→	
	13.x Amount Message	→	
The EMV transaction is initiated as the card is inserted. "Please wait" is displayed on the screen while the card is read. The cardholder is prompted for language and application selection if the terminal is configured for manual selection. "Please wait" is again displayed, followed by the refund amount. Track 2 equivalent data is then sent to the POS.	EMV '33.02.x' Track 2 Equivalent Data Message	←	

Sequence	Message	POS	Terminal
A Set Payment Type request message is sent to the terminal with the payment type. In order to process the transaction as full EMV refund, the EMV Refund Option flag must be set to '1'.	04.x Set Payment Type Request	→	
The terminal then responds with a 04.x: Set Payment Type Response message confirming the payment type.	04.x Set Payment Type Response	←	
A 13.x Amount message is sent to the terminal with the final transaction amount.	13.x Amount Message	→	
Depending on the setting of the Verify Amount flag, the cardholder may be prompted to confirm the refund amount. They may also be prompted to select the account type (if Interac), and enter their PIN (if required).			
An authorization request containing the necessary cryptographic information is sent to the POS.	EMV '33.03.x' Authorization Request Message	←	
The POS responds with an authorization response containing information to be read by the EMV card.	EMV '33.04.x' Authorization Response Message	→	
The transaction is approved or denied, based on the authorization response content. A confirmation response is sent to the POS.	EMV '33.05.x' Authorization Confirmation Response Message	←	
The cardholder is prompted to remove their card.			

Sequence	Message	POS	Terminal
When the card is removed a '09.020201R' message is sent to the POS indicating that the card has been removed.	09.x Card Status Message	←	

8.4.4 EMV Partial Refund Normal Transaction Flow

When EMV Partial Refund transactions are initiated, the POS sends a 14.x Set Transaction Type message and 13.x Amount Message. The '14.03' message identifies the transaction as a refund, and the amount of the refund is specified in the 13.x Amount message. If the card is inserted before this message is received, then the transaction defaults to an EMV purchase transaction. If this message is received after the card is inserted, then the purchase transaction will be cancelled and a refund transaction will be initiated.

Once the partial refund transaction is initiated with the card detected, the cardholder may be prompted to select the language and application if not configured for auto-selection in the configuration file. If the auto-selection flags are enabled then the terminal will select the language and application based on highest priority.

With the language and application selected and the transaction amount sent, the terminal will send an EMV '33.02.x' Track 2 Equivalent Data Message message. As a future option, the POS may request this information via an EMV '33.02.x' Track 2 Equivalent Data Request message, but it is not recommended at this time.

Depending on the setting of the Verify Amount flag in Card Configuration (cards.dat), the cardholder may be prompted to confirm the refund amount. <Yes>, <No>, and <Cancel> buttons are displayed.

The terminal returns an EMV '33.05.x' Authorization Confirmation Response Message message to the POS confirming the refund transaction state. For a partial refund, the Approval Status tag D1003 is be set to "C" (Completed) and no Application Cryptogram is generated. The cardholder is then prompted to remove their card. When the card is removed, the terminal sends a 09.x Card Status Message to the POS indicating that the card has been removed.

Refer to the below table for an illustration of the EMV partial refund transaction sequence.

EMV Partial Refund Transaction Flow

Sequence	Message	POS	Terminal
The POS sends transaction type message to terminal.	14.x Set Transaction Type - 14.03 = Refund.	→	
The POS sends the refund amount.	13.x Amount Message	→	

Sequence	Message	POS	Terminal
The EMV transaction is initiated as the card is inserted. "Please wait" is displayed on the screen while the card is read. The cardholder is prompted for language and application selection if the terminal is configured for manual selection. "Please wait" is again displayed, followed by the refund amount. Track 2 equivalent data is sent to the POS in tag T57 which is included in the '33.02.x' message.	EMV '33.02.x' Track 2 Equivalent Data Message	←	
For a partial refund, the Authorization Response Code (ARC) will be set as "Offline Declined" and the cryptogram type will be AAC (Application Authentication Cryptogram). A confirmation response which includes the ARC in tag T8A is sent to the POS.	EMV '33.05.x' Authorization Confirmation Response Message	←	
The cardholder is prompted to remove their card.			
Terminal sends a '09.020201R' message when the card is removed.	09.x Card Status Message	←	
The terminal returns to the idle state.			

8.4.5 EMV Partial Refund On-Demand Transaction Flow

EMV Partial Refund on-demand transactions are initiated with a 23.x Card Read Request (On-Demand) which sets the transaction as on-demand. The EMV card is inserted and an EMV '33.00.x' Transaction Initiation message is sent from the POS, followed by a 14.x Set Transaction Type message and 13.x Amount Message. The '14.03' message identifies the transaction as a refund, and the amount of the refund is specified in the 13.x Amount message. If the card is inserted before this message is received, then the transaction defaults to an EMV purchase transaction. If this message is received after the card is inserted, then the purchase transaction will be cancelled and a refund transaction will be initiated.

Once the partial refund transaction is initiated with the card detected, the cardholder may be prompted to select the language and application if not configured for auto-selection in the configuration file. If the auto-selection flags are enabled then the terminal will select the language and application based on highest priority.

With the language and application selected and the transaction amount sent, the terminal will send an EMV '33.02.x' Track 2 Equivalent Data Message message. As a future option, the POS may request this information via an EMV '33.02.x' Track 2 Equivalent Data Request message, but it is not recommended at this time.

A 04.x Set Payment Type Request message follows the EMV '33.02.x' Track 2 Equivalent Data Message message. The terminal responds with a 04.x: response message confirming the payment type request (e.g., Debit, Credit). In order to process the transaction as partial EMV refund, the EMV Refund Option flag must be set to '0'. The EMV Refund Option flag is set using the 60.x Configuration Write message to modify the cards.dat file before the start of the transaction.

Following the 04.x: Set Payment Type Request message, a second 13.x Amount Message with the final transaction amount is sent to the terminal so that it may proceed with the refund transaction. Depending on the setting of the Verify Amount flag, the cardholder may be prompted to confirm the refund amount. <Yes>, <No>, and <Cancel> buttons are displayed.

An EMV '33.05.x' Authorization Confirmation Response Message message is returned to the POS confirming the refund transaction state and including the necessary tag information for printing. For a partial refund, the Authorization Response Code (ARC) will be set as "Offline Declined" and the cryptogram type will be AAC (Application Authentication Cryptogram). The cardholder is then prompted to remove their car. When the card is removed, the terminal sends a 09.x Card Status Message to the POS indicating that the card has been removed.

Refer to the below table for an illustration of the EMV partial refund transaction sequence.

EMV Partial Refund Transaction Flow

Sequence	Message	POS	Terminal
The POS sends the 23.x message for on-demand flow.	23.x Card Read Request (On-Demand)	→	
The EMV card is inserted.			
The POS sends an EMV transaction initiation message to terminal.	EMV '33.00.x' Transaction Initiation Message	→	
The POS sends a transaction type message to terminal.	14.x Set Transaction Type • 14.03 = Refund.	→	
The POS sends the refund amount.	13.x Amount Message	→	

Sequence	Message	POS	Terminal
The EMV transaction is initiated as the card is inserted. "Please wait" is displayed on the screen while the card is read. The cardholder is prompted for language and application selection if the terminal is configured for manual selection. "Please wait" is again displayed, followed by the refund amount. Track 2 equivalent data is sent to the POS in tag T57 which is included in the '33.02.x' message.	EMV '33.02.x' Track 2 Equivalent Data Message	←	
A Set Payment Type request message is sent to the terminal with the payment type. In order to process the transaction as full EMV refund, the EMV Refund Option flag must be set to '1'.	04.x Set Payment Type Request	→	
The terminal then responds with a 04.x: Set Payment Type Response message confirming the payment type.	04.x Set Payment Type Response	←	
Following the 04.x Set Payment Type message, a second 13.x Amount message with the final transaction amount is sent to the terminal so that it may proceed with the refund transaction.	13.x Amount Message	→	
Depending on the setting of the Verify Amount flag, the cardholder may be prompted to confirm the refund amount.			

Sequence	Message	POS	Terminal
For a partial refund, the Authorization Response Code (ARC) will be set as "Offline Declined" and the cryptogram type will be AAC (Application Authentication Cryptogram). A confirmation response which includes the ARC in tag T8A is sent to the POS.	EMV '33.05.x' Authorization Confirmation Response Message	←	
The cardholder is prompted to remove their card.			
When the card is removed, the terminal sends a '09.020201R' message to the POS indicating that the card has been removed.	09.x Card Status Message	←	
The terminal returns to the idle state.			

8.4.6 EMV Contactless On-Demand Full Refund Transaction Flow

EMV contactless on-demand refund transactions are processed for cards which fully support EMV transactions. The transaction type is set as a refund using the 14.x Set Transaction Type message. This is followed by the 13.x Amount Message. Once the amount message is received, the contactless reader is enabled and the cardholder can tap their card. Tapping the card before the reader is enabled will produce a '0xFB' error.

When the card is tapped, a '23.0E' message is returned to the POS. The 'E' in the source field indicates EMV contactless as the source of the card read. The cardholder may then be prompted to select the language and application if not configured for auto-selection in the configuration file. If the auto-selection flags are enabled then the terminal will select the language and application based on highest priority.

With the language and application selected and the transaction amount sent, the terminal will send an EMV '33.02.x' Track 2 Equivalent Data Message message.

A 04.x Set Payment Type Request message follows the EMV '33.02.x' Track 2 Equivalent Data Message message. The terminal responds with a 04.x: response message confirming the payment type request (e.g., Debit, Credit). In order to process the transaction as full EMV refund, the EMV Refund Option flag must be set to '1'. The EMV Refund Option flag is set using the 60.x Configuration Write message to modify the cards.dat file before the start of the transaction.

A second 13.x Amount Message with the final transaction amount is then sent to the terminal. With the final transaction information received from the POS, the cardholder may be prompted to confirm the refund amount depending on the setting of the Verify Amount flag. The cardholder may additionally be prompted for an account selection (checking or savings). The cardholder is then prompted for PIN entry if required.

The terminal sends an EMV '33.03.x' Authorization Request Message to the POS, which will return the EMV '33.04.x' Authorization Response Message. An EMV '33.05.x' Authorization Confirmation Response Message is then returned to the POS confirming the refund transaction state and including the necessary tag information for printing. Included in the '33.05.x' message will be tag D1003 which will indicate the approval status ("A" for Approved and "D" for Declined) or D1004 = 0 if the Host response is not available. The terminal will then display the transaction status as "Approved" or "Declined" based on this tag value.

Refer to the below table for an illustration of the EMV refund transaction sequence.

EMV Contactless On-Demand Refund Transaction Flow

Sequence	Message	POS	Terminal
Transaction type (refund) and refund amount are sent to the terminal.	14.x Set Transaction Type • 14.03 = Refund.	→	
	13.x Amount Message	→	
A 23.x Card Read Request message is sent from the POS to enable the contactless card reader.	23.x Card Read Request (On-Demand)	→	
Once the 13.x Amount message is received, the contactless reader is enabled and the cardholder can tap their card. When the card is tapped, a '23.0E' message is returned to the POS. The 'E' in the source field indicates EMV contactless as the source of card read. Tapping a card before the reader is enabled will produce a '0xFB' error.	'23.0E' Response	←	

Sequence	Message	POS	Terminal
The EMV transaction is initiated as the card is tapped. "Please wait" is displayed on the screen while the card is read. The cardholder is prompted for language and application selection if the terminal is configured for manual selection. "Please wait" is again displayed, followed by the refund amount. Track 2 equivalent data is then sent to the POS.	EMV '33.02.x' Track 2 Equivalent Data Message	←	
A Set Payment Type request message is sent to the terminal with the payment type. In order to process the transaction as full EMV refund, the EMV Refund Option flag must be set to '1'.	04.x Set Payment Type Request	→	
The terminal then responds with a 04.x: Set Payment Type Response message confirming the payment type.	04.x Set Payment Type Response	←	
A 13.x Amount message is sent to the terminal with the final transaction amount.	13.x Amount Message	→	
Depending on the setting of the Verify Amount flag, the cardholder may be prompted to confirm the refund amount. They may also be prompted to select the account type (if Interac), and enter their PIN (if required).			
An authorization request containing the necessary cryptographic information is sent to the POS.	EMV '33.03.x' Authorization Request Message	←	

Sequence	Message	POS	Terminal
The POS responds with an authorization response containing information to be read by the EMV card.	EMV '33.04.x' Authorization Response Message	→	
The transaction is approved or denied, based on the authorization response content. A confirmation response is sent to the POS.	EMV '33.05.x' Authorization Confirmation Response Message	←	

8.4.7 Cancelling an EMV Transaction

The EMV '33.09.x' Set Tag Data Message can be used to cancel an EMV transaction. This is done by setting the command type in the message to "C" (cancel transaction). The 33.09.x message is sent after the EMV '33.02.x' Track 2 Equivalent Data Message is sent to the POS. The terminal then cancels the transaction and responds with an EMV EMV '33.05.x' Authorization Confirmation Response Message.

The following example transaction flows show how the EMV '33.09.x' Set Tag Data Message can be used to cancel an EMV purchase transaction and EMV refund transaction.

Cancelling an EMV Purchase Transaction Using the EMV '33.09.x' Set Tag Data Message

Sequence	Message	POS	Terminal
Transaction type and final amount are sent to the terminal.	14.x Set Transaction Type • 14.01 = Purchase. • 14.03 = Refund.	→	
	13.x Amount Message	→	

Sequence	Message	POS	Terminal
The EMV transaction is initiated as the card is inserted. "Please wait" is displayed on the screen while the card is read. The cardholder is prompted for language and application selection if the terminal is configured for manual selection. "Please wait" is again displayed, followed by the purchase amount. Track 2 equivalent data is then sent to the POS.	EMV '33.02.x' Track 2 Equivalent Data Message	←	
The POS requests the payment type.	04.x Set Payment Type Request	→	
The terminal responds with the payment type and cash back amount.	04.x Set Payment Type Response	←	
Cancelling the Transaction The POS sends a Set Tag Data message with the command type set to "C" to cancel the transaction.	EMV '33.09.x' Set Tag Data Message	→	
The terminal receives the cancellation message and return s an authorization confirmation response.	EMV '33.05.x' Authorization Confirmation Response Message	←	
The cardholder is prompted to remove their card.			
When the card is removed, the terminal sends a '09.020201R' message to the POS indicating that the card has been removed.	09.x Card Status Message	←	

Cancelling a Refund Transaction Using the EMV '33.09.x' Set Tag Data Message

Sequence	Message	POS	Terminal
The POS sends the 23.x message to initiate on-demand flow.	23.x Card Read Request (On-Demand)	→	
The EMV card is inserted.			
The POS sends the transaction initiation message to the terminal with suspend at step H so that tag data can be set.	EMV '33.00.x' Transaction Initiation Message	→	
The POS sends transaction type message to terminal.	14.x Set Transaction Type • 14.03 = Refund.	→	
The POS sends the refund amount.	13.x Amount Message	→	
The terminal responds with a Track 2 Equivalent Data message as well as a Status message indicating that the card has been inserted. "Please wait" is displayed on the screen while the card is read. The cardholder is prompted for language and application selection if the terminal is configured for manual selection. "Please wait" is again displayed, followed by the refund amount.	EMV '33.02.x' Track 2 Equivalent Data Message EMV 33.01.x Status Message	←	
Cancelling the Transaction The POS sends a Set Tag Data message with the command type set to "C" to cancel the transaction.	EMV '33.09.x' Set Tag Data Message	→	

Sequence	Message	POS	Terminal
The terminal receives the Set Tag Data message with the "cancel" command and cancels the transaction. A confirmation response is returned to the POS.	EMV '33.05.x' Authorization Confirmation Response Message	←	
The cardholder is prompted to remove their card.			
When the card is removed, the terminal sends a '09.020201R' message to the POS indicating that the card has been removed.	09.x Card Status Message	←	

8.4.8 MSD Contactless Transaction Flow

A transaction for a Magnetic Stripe Data (MSD) card is initiated when the POS sends a 14.x Transaction Type message to the terminal. A 13.x message must be sent to the terminal before the MSD card is tapped. If not, then the terminal will display an error message indicating that no amount has been entered.

If EMV contactless mode is enabled (Contactless Reader Mode parameter '0008_0001' = '9'), the terminal will automatically proceed with a contactless transaction when the MSD card is tapped. When the terminal detects that the card is not an EMV card, it will follow MSD transaction flow. Once the card is tapped, there is no further input from the cardholder. A 50.x Authorization Request message is sent to the POS with the information required to create the authorization message.

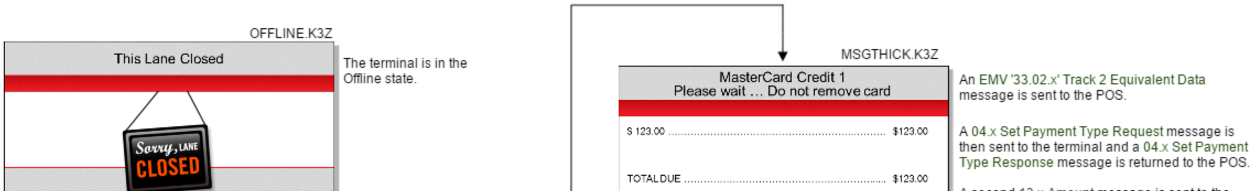
> When EMV is not enabled for contactless transactions (i.e., when '0019_0001' = 0), PayPass kernel 2 must be selected, by setting '0008_0015' = 2.

The terminal waits for the 0x Authorization Response message which contains the approval code, and displays the approval status as "Approved" or "Declined." There is no further interaction with the cardholder once the card is tapped. A 10.x Hard Reset message is then sent to the POS indicating that the terminal is returning to the online idle state. Refer to the below table for an illustration of the MSD Contactless Transaction sequence.

Refer to the General Message Flow section which describes the flow for this transaction type.

8.4.9 EMV Transaction Flows with Application Selection

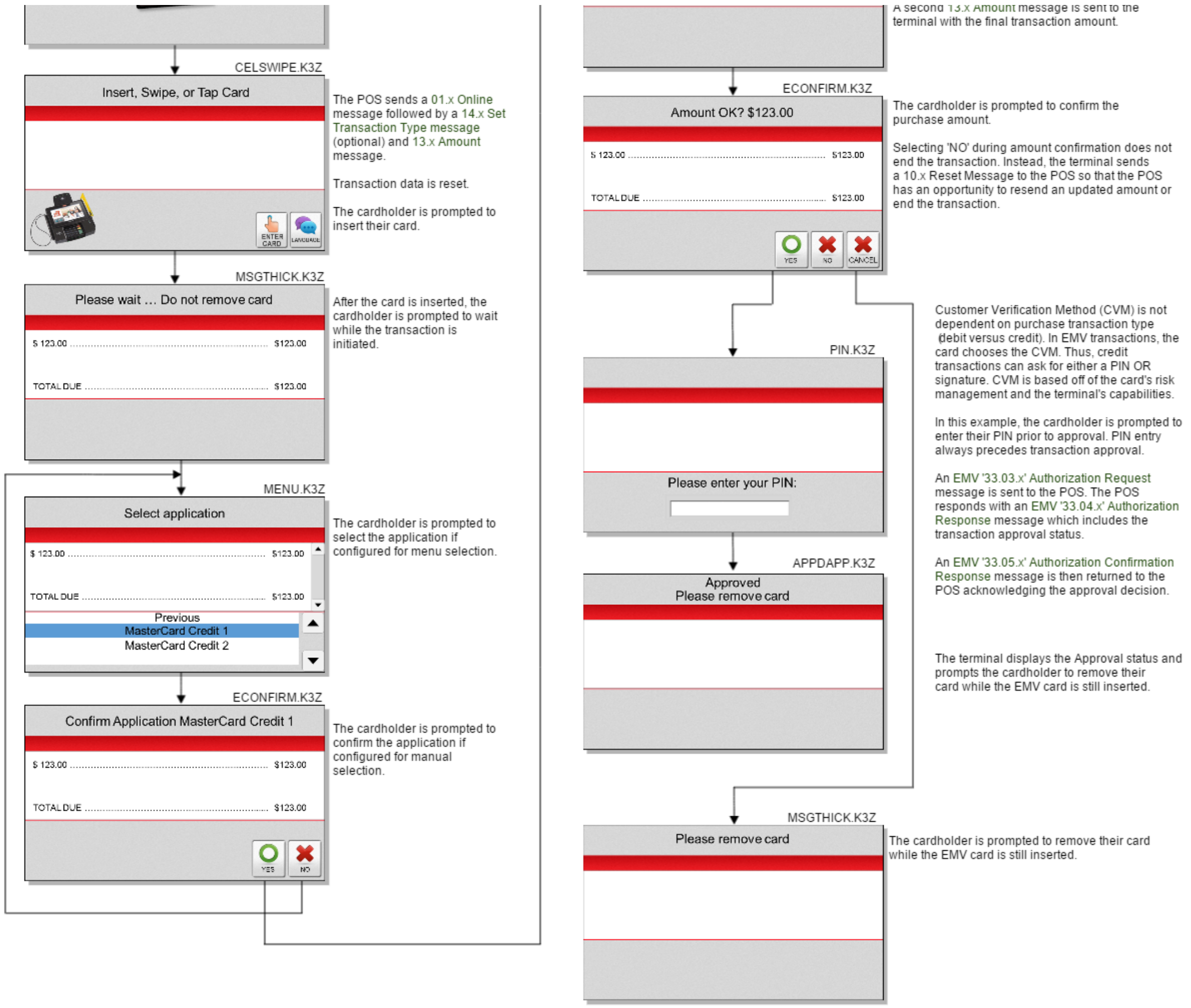
8.4.9.1 Credit Sale with PIN Entry and Application Selection Menu



The terminal is in the Offline state.

An EMV '33.02.x' Track 2 Equivalent Data message is sent to the POS.

A 04.x Set Payment Type Request message is then sent to the terminal and a 04.x Set Payment Type Response message is returned to the POS.



The POS sends a 01.x Online message followed by a 14.x Set Transaction Type message (optional) and 13.x Amount message.

Transaction data is reset.

The cardholder is prompted to insert their card.

After the card is inserted, the cardholder is prompted to wait while the transaction is initiated.

The cardholder is prompted to select the application if configured for menu selection.

The cardholder is prompted to confirm the application if configured for manual selection.

A second 13.x Amount message is sent to the terminal with the final transaction amount.

The cardholder is prompted to confirm the purchase amount.

Selecting 'NO' during amount confirmation does not end the transaction. Instead, the terminal sends a 10.x Reset Message to the POS so that the POS has an opportunity to resend an updated amount or end the transaction.

Customer Verification Method (CVM) is not dependent on purchase transaction type (debit versus credit). In EMV transactions, the card chooses the CVM. Thus, credit transactions can ask for either a PIN OR signature. CVM is based off of the card's risk management and the terminal's capabilities.

In this example, the cardholder is prompted to enter their PIN prior to approval. PIN entry always precedes transaction approval.

An EMV '33.03.x' Authorization Request message is sent to the POS. The POS responds with an EMV '33.04.x' Authorization Response message which includes the transaction approval status.

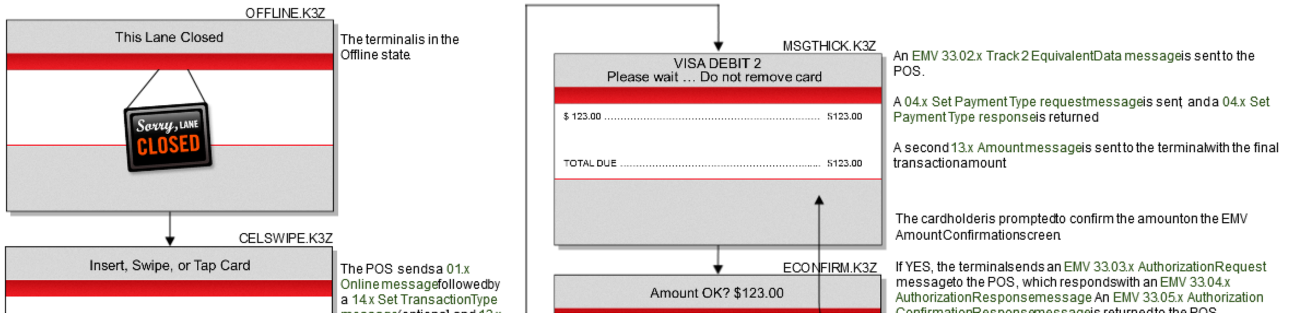
An EMV '33.05.x' Authorization Confirmation Response message is then returned to the POS acknowledging the approval decision.

The terminal displays the Approval status and prompts the cardholder to remove their card while the EMV card is still inserted.

The cardholder is prompted to remove their card while the EMV card is still inserted.

> For EMV Contactless, the application name will not be displayed above "Please wait ..."

8.4.9.2 *Debit Sale with Signature and Application Selection Menu*



The terminalis in the Offline state.

The POS sendsa 01.x Online messagefollowedby a 14.x Set TransactionType message(optional) and 13.x

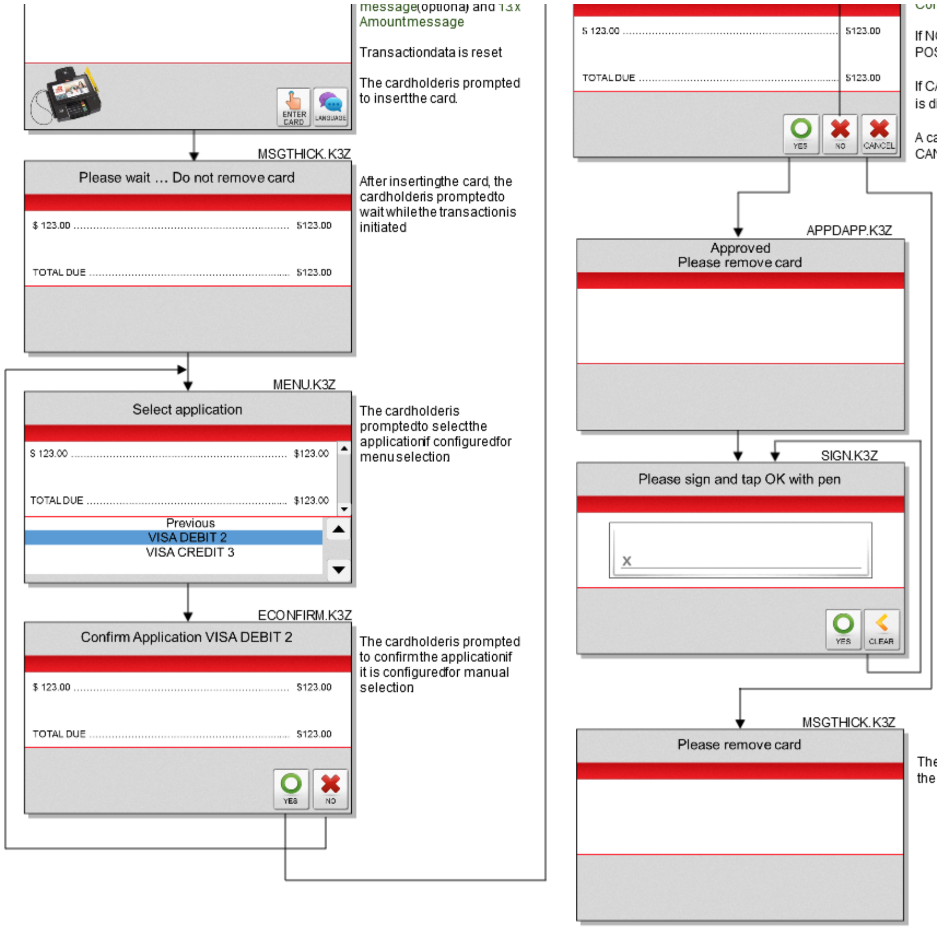
An EMV 33.02.x Track 2 EquivalentData messageis sentto the POS.

A 04.x Set Payment Type requestmessageis sent, and a 04.x Set Payment Type responseis returned

A second 13.x Amountmessageis sentto the terminalwith the final transactionamount

The cardholderis promptedto confirm the amounton the EMV AmountConfirmationscreen.

If YES, the terminalsends an EMV 33.03.x AuthorizationRequest messageto the POS, which respondswith an EMV 33.04.x AuthorizationResponsemessage An EMV 33.05.x Authorization ConfirmationResponsemessageis returnedto the POS.



message (optional) and 13.x Amount message

Transaction data is reset

The cardholder is prompted to insert the card.

After inserting the card, the cardholder is prompted to wait while the transaction is initiated

The cardholder is prompted to select the application if configured for menu selection

The cardholder is prompted to confirm the application if it is configured for manual selection

Confirmation Response message is returned to the POS.

If NO, the terminal sends a 10.x Reset Message to the POS, so the POS can resend an updated amount or end the transaction

If CANCEL, the Transaction Canceling Please Remove Card screen is displayed

A cardholder can also use the terminal keypad ENTER, CLEAR, and CANCEL for the expected result during the transaction

If YES is selected, an EMV 33.05.x Authorization Confirmation Response message is returned

Because the Customer Verification Method (CVM) is set to signature, the terminal displays the Approval status and prompts the cardholder to remove the card while the EMV card is still inserted

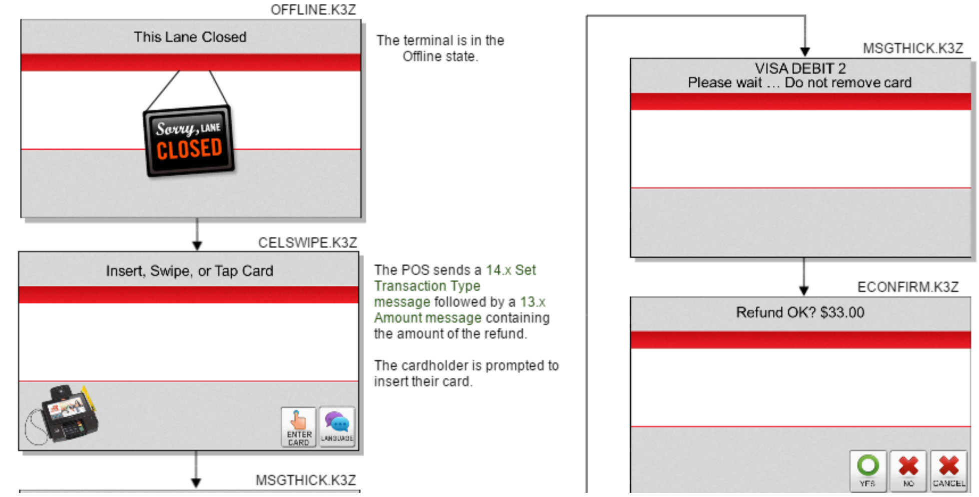
CVM is not dependent on transaction type. In EMV transactions the card chooses the CVM., so debit transactions can request a PIN or signature CVM is based on the card's settings and the terminals capabilities

In this example, the cardholder is prompted to enter a signature. Pressing the CLEAR button restarts the signature process

The cardholder is prompted to remove the card while the EMV card is still inserted in the reader.

For EMV contactless transactions, the application name is not displayed above *Please wait...*

8.4.9.3 Full Refund with Application Selection Menu



The terminal is in the Offline state.

The POS sends a 14.x Set Transaction Type message followed by a 13.x Amount message containing the amount of the refund.

The cardholder is prompted to insert their card.

A 04.x Set Payment Type Request message is sent to the terminal with the payment type configuration defining the transaction as partial (non-EMV) refund.

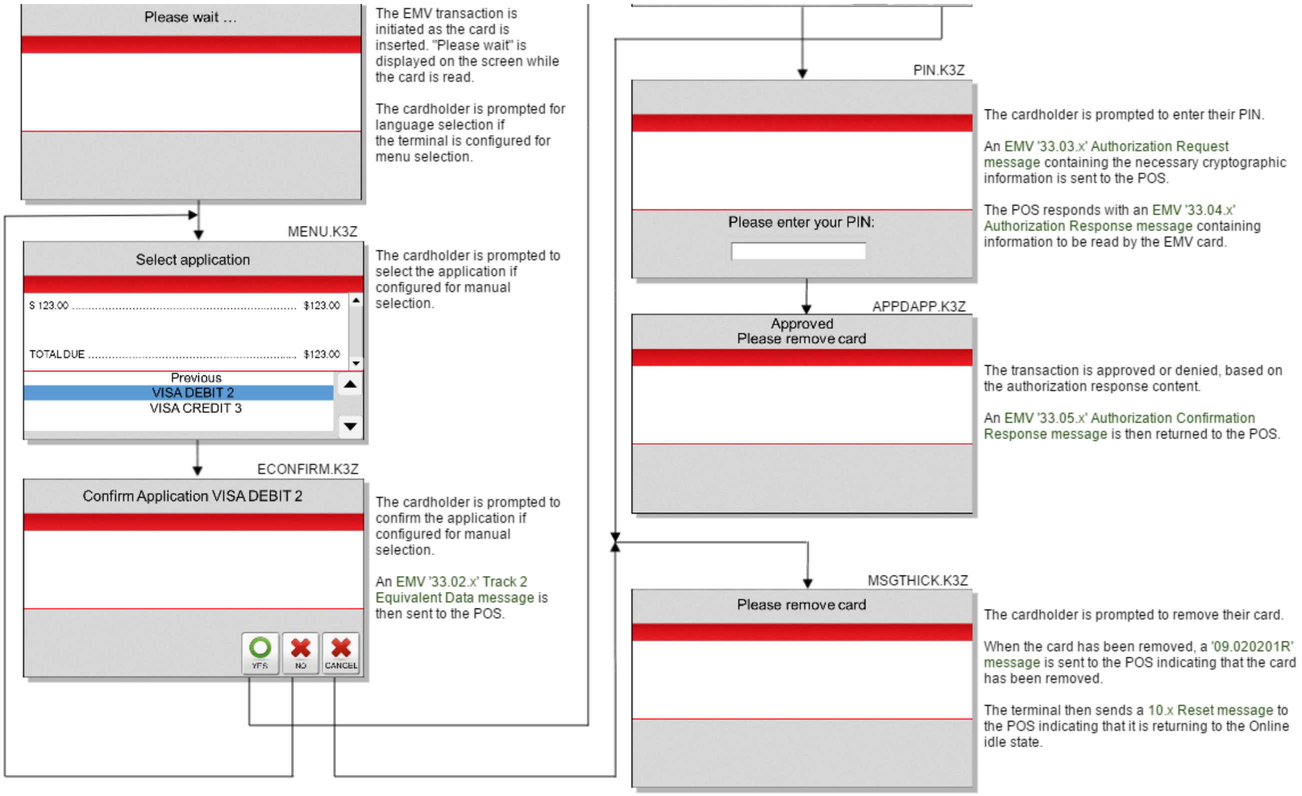
Also included in the Set Payment Type message is the configuration to enable or disable prompting the cardholder to confirm the refund amount.

The terminal then responds with a 04.x: Set Payment Type Response message confirming the configuration.

Following the 04.x Set Payment Type message, a 13.x Amount message with the final transaction amount is sent to the terminal so that it may proceed with the refund transaction.

The cardholder is prompted to confirm the refund amount.

Selecting 'NO' during amount confirmation does not end the transaction. Instead, the terminal sends a 10.x Reset Message to the POS so that the POS has an opportunity to resend an updated amount or end the transaction.



The EMV transaction is initiated as the card is inserted. "Please wait" is displayed on the screen while the card is read.

The cardholder is prompted for language selection if the terminal is configured for menu selection.

The cardholder is prompted to select the application if configured for manual selection.

The cardholder is prompted to confirm the application if configured for manual selection.

An EMV '33.02.x' Track 2 Equivalent Data message is then sent to the POS.

The cardholder is prompted to enter their PIN.

An EMV '33.03.x' Authorization Request message containing the necessary cryptographic information is sent to the POS.

The POS responds with an EMV '33.04.x' Authorization Response message containing information to be read by the EMV card.

The transaction is approved or denied, based on the authorization response content.

An EMV '33.05.x' Authorization Confirmation Response message is then returned to the POS.

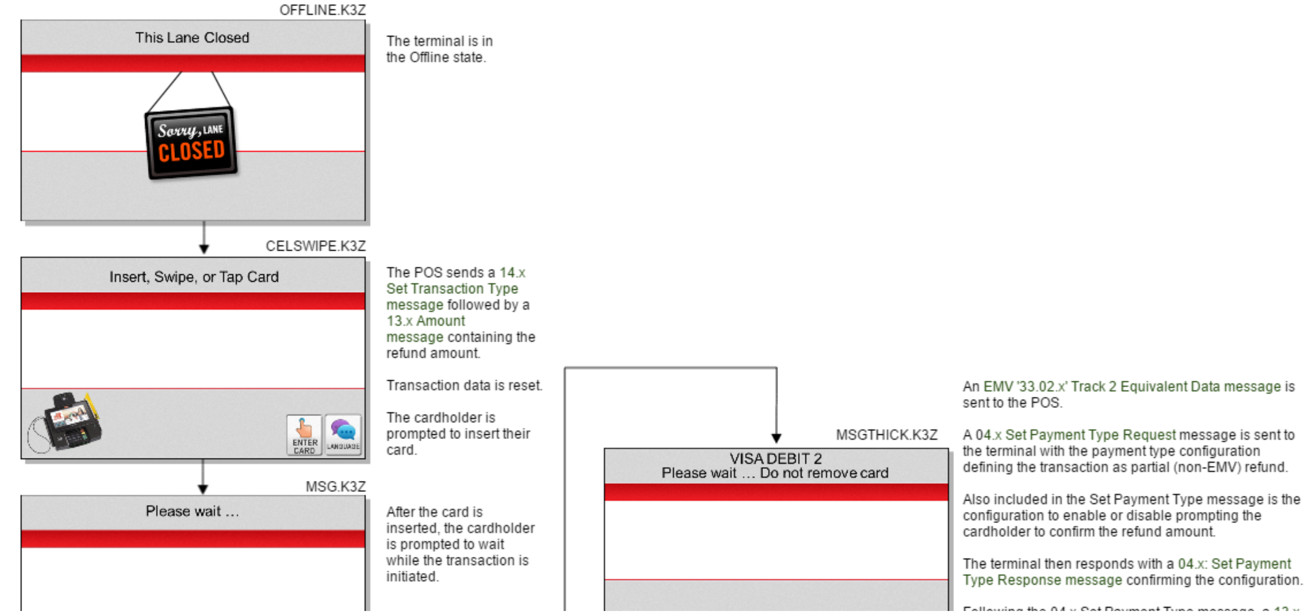
The cardholder is prompted to remove their card.

When the card has been removed, a '09.020201R' message is sent to the POS indicating that the card has been removed.

The terminal then sends a 10.x Reset message to the POS indicating that it is returning to the Online idle state.

For EMV Contactless, the application name will not be displayed above "Please wait ..."

8.4.9.4 Partial Refund Normal Flow with Application Selection Menu



The terminal is in the Offline state.

The POS sends a 14.x Set Transaction Type message followed by a 13.x Amount message containing the refund amount.

Transaction data is reset.

The cardholder is prompted to insert their card.

After the card is inserted, the cardholder is prompted to wait while the transaction is initiated.

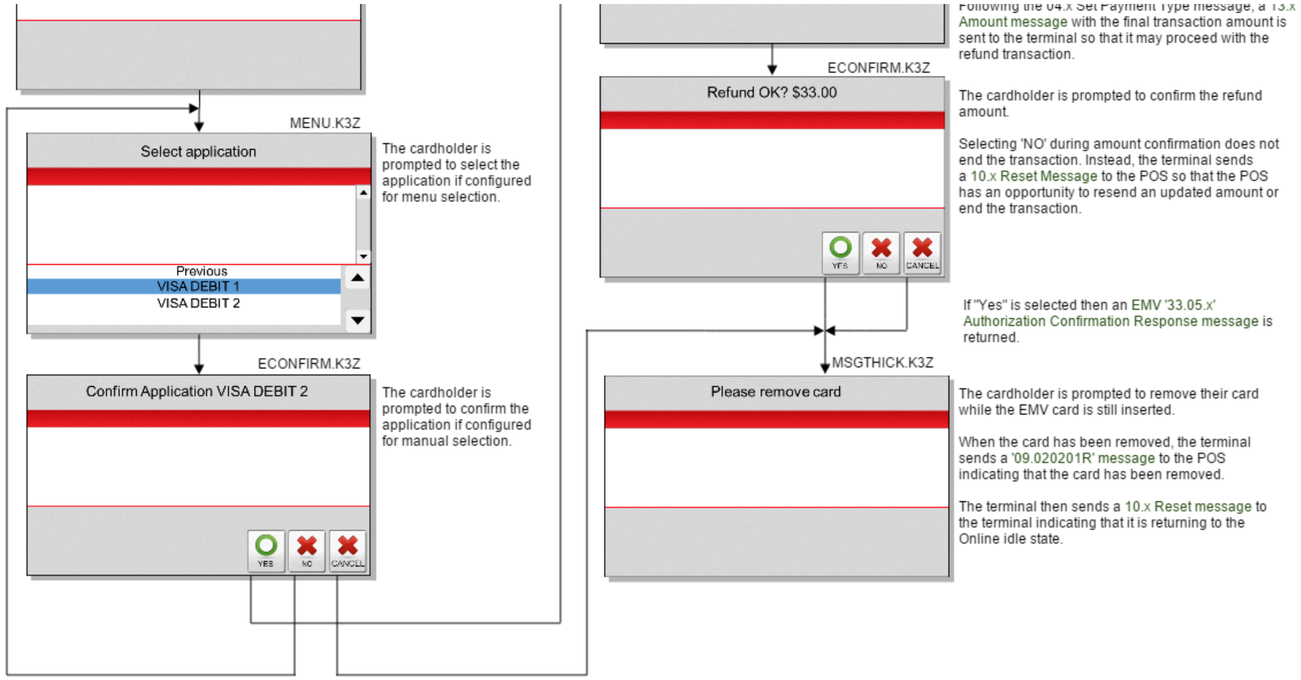
An EMV '33.02.x' Track 2 Equivalent Data message is sent to the POS.

A 04.x Set Payment Type Request message is sent to the terminal with the payment type configuration defining the transaction as partial (non-EMV) refund.

Also included in the Set Payment Type message is the configuration to enable or disable prompting the cardholder to confirm the refund amount.

The terminal then responds with a 04.x: Set Payment Type Response message confirming the configuration.

Following the 04.x Set Payment Type message, a 13.x



The cardholder is prompted to select the application if configured for menu selection.

The cardholder is prompted to confirm the application if configured for manual selection.

Following the 04.x Set Payment Type message, a 13.x Amount message with the final transaction amount is sent to the terminal so that it may proceed with the refund transaction.

The cardholder is prompted to confirm the refund amount.

Selecting 'NO' during amount confirmation does not end the transaction. Instead, the terminal sends a 10.x Reset Message to the POS so that the POS has an opportunity to resend an updated amount or end the transaction.

If "Yes" is selected then an EMV '33.05.x' Authorization Confirmation Response message is returned.

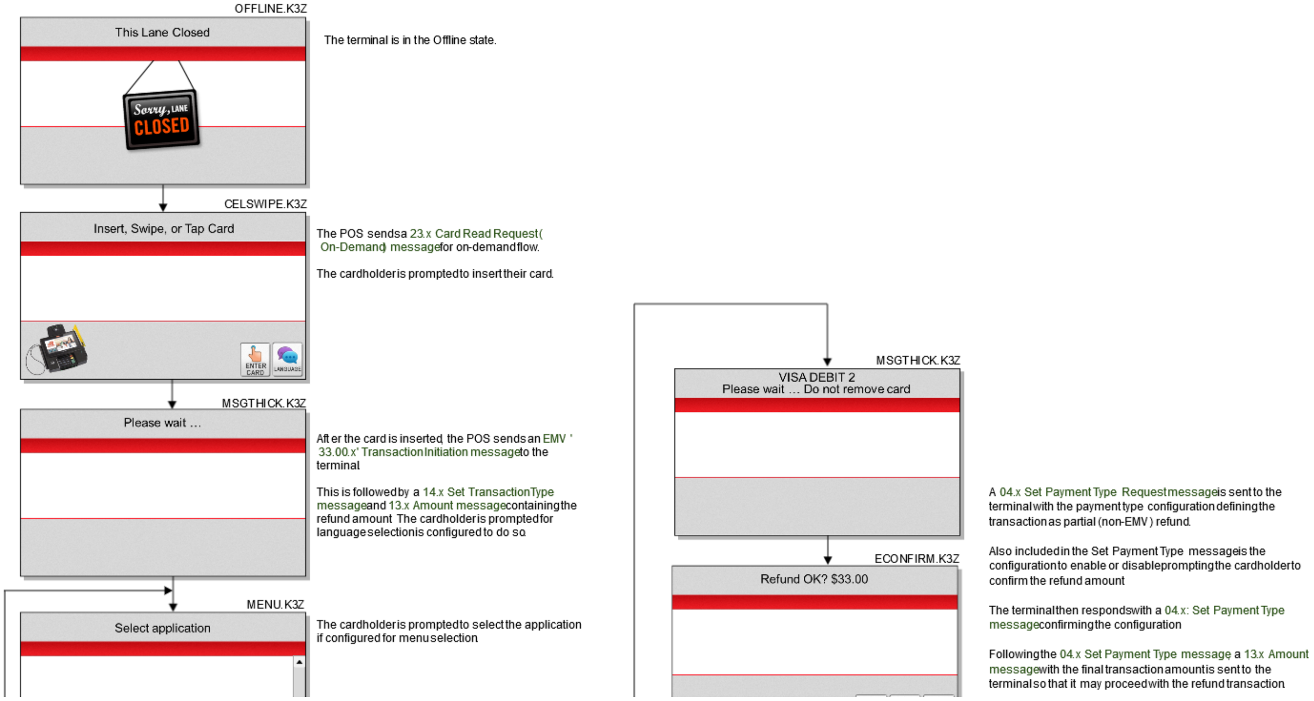
The cardholder is prompted to remove their card while the EMV card is still inserted.

When the card has been removed, the terminal sends a '09.020201R' message to the POS indicating that the card has been removed.

The terminal then sends a 10.x Reset message to the terminal indicating that it is returning to the Online idle state.

For EMV Contactless, the application name will not be displayed above "Please wait ..."

8.4.9.5 Partial Refund On-Demand Flow with Manual Application Selection



The terminal is in the Offline state.

The POS sendsa 23.x Card Read Request(On-Demand) messagefor on-demandflow.

The cardholderis promptedto insert their card.

Aft er the card is inserted, the POS sendsan EMV ' 33.00.x' TransactionInitiation messageto the terminal.

This is followedby a 14.x Set TransactionType messageand 13.x Amount messagecontainingthe refund amount. The cardholderis promptedfor languageselectionis configured to do so.

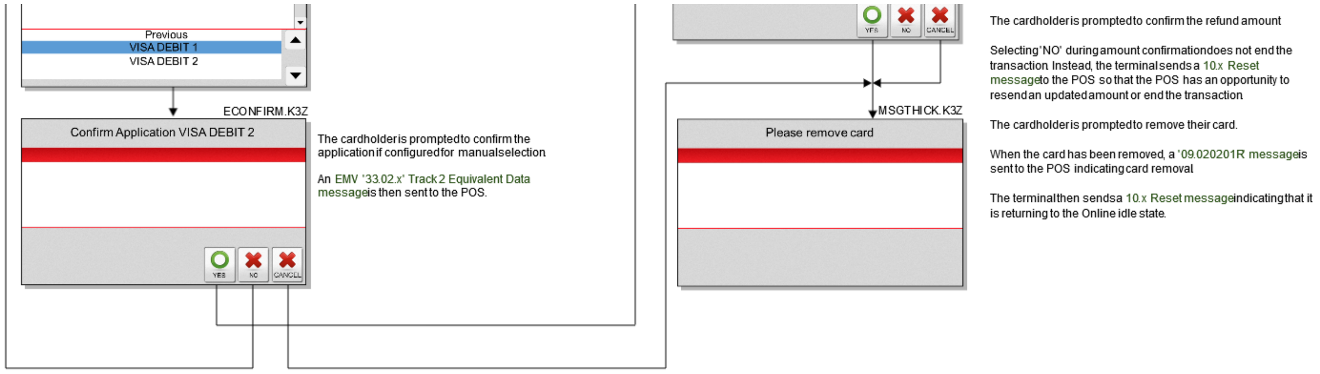
The cardholderis promptedto selectthe application if configured for menuselection.

A 04.x Set Payment Type Requestmessageis sentto the terminalwith the paymenttype configuration definingthe transactionas partial (non-EMV) refund.

Also includedin the Set Payment Type messageis the configurationto enable or disablepromptingthe cardholderto confirm the refund amount.

The terminalthen respondswith a 04.x: Set Payment Type messageconfirmingthe configuration.

Followingthe 04.x Set Payment Type message, a 13.x Amount messagewith the final transactionamountis sentto the terminalso that it may proceedwith the refund transaction.



The cardholder is prompted to confirm the application if configured for manual selection.

An EMV '33.02.x' Track 2 Equivalent Data message is then sent to the POS.

The cardholder is prompted to confirm the refund amount

Selecting 'NO' during amount confirmation does not end the transaction. Instead, the terminal sends a 10.x Reset message to the POS so that the POS has an opportunity to resend an updated amount or end the transaction.

The cardholder is prompted to remove their card.

When the card has been removed, a '09.020201R' message is sent to the POS indicating card removal.

The terminal then sends a 10.x Reset message indicating that it is returning to the Online idle state.

For EMV Contactless, the application name will not be displayed above "Please wait ..."

8.5 EMV On-Demand Flow

8.5.1 Configuration

The following table shows the parameters which must be configured in order for the terminal to process EMV transactions in the On-Demand flow:

Parameters to Configure for EMV On-Demand Flow

Parameter	Setting
0019_0001	This EMV flag must be set to '1' to enable the terminal to support EMV transactions.
0013_0014	This Compatibility flag must be set to '1' so that the Source field is included in the 23.x Card Read Request message.

Changing these parameters is accomplished by using the 60.x Configuration Write message to update the `config.dfs` file.

8.5.2 Initiate On-Demand

To initiate On-Demand, A '23.Please Slide Card' message is sent to the terminal from the POS. When the EMV card is inserted, the following message is returned from the terminal:

'23.0S[FS]'

where '0' indicates a good card read and 'S' indicates that a smart (EMV) card has been inserted. With the EMV card detected, the POS can proceed with sending an EMV '33.00.x' Transaction Initiation Message.

8.5.3 Initiate EMV Flow

To initiate the EMV flow, an EMV '33.00.x' Transaction Initiation Message is sent which will include the transaction amount. Refer to the following example:

'33.00.0000[FS][FS][FS][FS][FS]'

The cardholder will then be prompted to confirm the application (e.g., VISA, MasterCard). The cardholder may also be prompted to select the language. From here, transactions will continue per the normal EMV flow.

8.6 Enabling EMV Cash Back

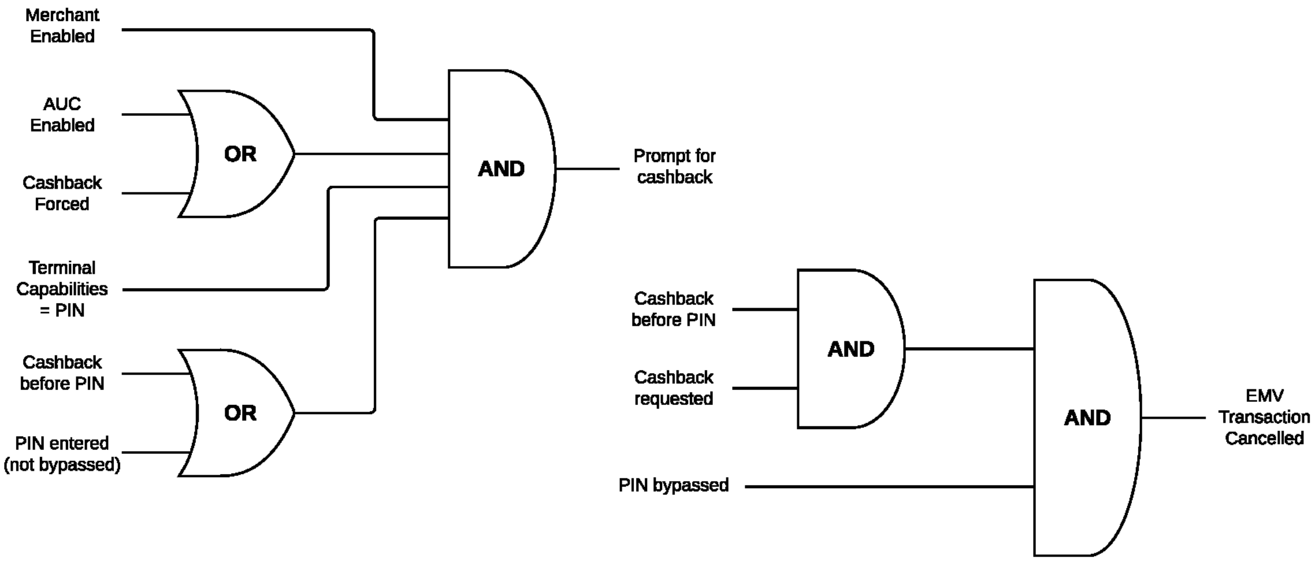
8.6.1 Overview

The following conditions must be met for the terminal to prompt a cardholder for cashback:

- Cashback must be enabled: 04.x Set Payment Type Request 0011_00xx::Cashback Limit configured (set to a value other than 0)
- AUC must be enabled (card AID has appropriate domestic/international bits enabled) OR cashback is forced (AID is configured to force cashback and ignore AUC cashback bits, 0021_00xx : : Force cashback)
- Terminal capabilities (T9F33:byte 2:bits 7,5) = PIN enabled

8.6.2 EMV Cashback Process

The following diagram illustrates the EMV cashback process:



RBA EMV Cashback Prompt Logic

> Selecting No on the Amount Confirmation screen when 0007_0006 = 2 prompts the cardholder to select cash back again.

8.6.3 Cashback/Total Amount Confirmation

The cashback/total amount confirmation for EMV transactions is displayed using one of the following forms:

- `$<Purchase amount> Please confirm`
    - This form is displayed to **confirm the total purchase amount only before PIN entry only if**:
        - `NOT(CB>PIN AND CBReqd) AND AmtConf`
- `$<Purchase amount> + $<Cashback amount> = $<Total amount> Please confirm`
    - This form is displayed to **confirm the cashback/total amounts before PIN entry only if**:
        - `NOT(CB>PIN AND CBReqd) AND (CBConf OR AmtConf)`
    - This form is displayed to **confirm the cashback/total amounts after PIN entry only if**:
        - `NOT(CB<PIN AND CBReqd) AND (CBConf OR AmtConf)`

where:

- `AmtConf` = 0011_00xx Verify Amount = 1 confirmation enabled
- `CBConf` = 0011_00xx Verify Cashback = 1 confirmation enabled
- `CB>PIN` = 0002_0010 = 0 cashback flow setting before PIN entry configured
- `CB<PIN` = 0002_0010 = 1 cashback flow setting after PIN entry configured
- `CBReqd` = cashback already prompted AND cashback > $0
- `P PC` = abbreviated $<Purchase amount> confirmation
- `P+CB=T` = abbreviated $<Purchase amount> + $<Cashback amount> = $<Total amount> confirmation

8.6.4 Sample EMV Cashback Messages

A 13.3000[GS]2000 message signifies:

- 3000 = $30.00 total purchase amount excluding $20.00 cashback
- 2000 = $20.00 cashback amount

When this 13.x message is sent, 29.x messages return the following values per variable:

- 29.00000303 returns 29.200003033000, `303` (purchase amount) = $30.00
- 29.00000305 returns 29.200003052000, `305` (cashback amount) = $20.00
- 29.00000306 returns 29.200003068000, `306` (maximum cashback) = $80.00
- 29.00000307 returns 29.200003075000 `307` (transaction total, `303` + `305`) = $50.00

A blank cashback amount field following a [GS] is actively parsed as $0 cashback.

A 13.3000[GS] message signifies:

- 3000 = $30.00 purchase amount with $0 cashback amount.

> The cashback amount, variable `305`s value, must be less than the value of maximum cashback, variable `306`.

8.6.5 Terminal and POS Notification

During on-demand EMV transactions, the POS informs the terminal of the cashback request via a 13.<purchase amount>[GS]<cashback amount> message.

The following table describes the process for notifying the POS of the cashback request:

Mode	Cashback Notification Process
Cashback before PIN Entry - RBA Flow	• 04.x Set Payment Type Request. • `29.305` (use 29.x Get Variable Request message to retrieve cashback amount variable `305`).
Cashback before PIN Entry - On-Demand	• The POS prompts for and handles the cashback amount and has the cashback input.
Cashback after PIN Entry - RBA Flow	• `29.305` (use 29.x: Get Variable Request message to retrieve cashback amount variable `305`). Cashback amount = T9F02 (Authorized Amount) - final 13.x Amount Message, where tag T9F02 is included in one of the following messages: ◦ EMV 33.03.x Authorization Request Message ◦ EMV 33.05.x Authorization Confirmation Response Message
Cashback after PIN Entry - On-Demand	• The POS prompts for and handles the cashback amount and has the cashback input.

8.6.6 Support for EMV Cashback Regardless of CVM or PIN Bypass

A merchant can configure the application to permit cash back to be enabled for an EMV transaction regardless of whether the cardholder meets the cardholder verification method (CVM) criteria or is bypassing PIN entry.

The following conditions must be met for a terminal to prompt the cardholder for cash back automatically:

- Cashback is enabled
- 04.x Set Payment Type Request 0011_00xx::Cashback Limit configured (set to value other than 0)
- The application usage control (AUC) must be enabled and one of the following conditions met:
  - The card AID has the appropriate T9F07:B2 domestic/international parameters enabled
  - Cashback is forced. The AID is configured to force cashback and ignore AUC cashback parameters, 0021_00xx::Force Cashback
  - Cashback is enabled for selected CVM and/or CVM action:
    - 0021_00xx::Offline PIN Cashback
    - 0021_00xx::Online PIN Cashback
    - 0021_00xx::Signature Cashback
    - 0021_00xx::No CVM Cashback
    - 0021_00xx::PIN-Bypass Cashback must also be enabled if the PIN is bypassed, even if a final selected CVM is enabled (such as signature)

8.6.6.1 Cashback/Total Amount Confirmation

The cashback/total amount confirmation for EMV transactions during standard flow is displayed using one of the following forms:

- $<Purchase amount> Please confirm - Confirms the total purchase amount if no cashback amount is selected
- $<Purchase amount> + $<Cashback amount> = $<Total amount> Please confirm - Confirms the total purchase amount if a cashback amount is selected

8.7 EMV Full and Partial Refunds

8.7.1 Overview

The RBA has been enhanced to support full and partial EMV refund transactions. EMV refunds are processed for cards which fully support EMV transactions. When a full refund transaction is processed, an EMV '33.03.x' Authorization Request Message is sent to the POS which returns an EMV EMV '33.04.x' Authorization Response Message. The Authorization Response Codes (e.g., Online Approved, Offline Declined) in tag T8A will be included in the EMV '33.05.x' Authorization Confirmation Response Message. For a partial refund, there is no authorization request/response exchange between the terminal and POS. The Authorization Response Code will be set as "Offline Declined" and the cryptogram type will be AAC (Application Authentication Cryptogram). Partial refund transactions, also referred to as non-EMV transactions, do not prompt the cardholder for cashback.

> Partial EMV refund transactions are also referred to as non-EMV transactions.

8.7.2 Important Tags Used in Refund Transactions

Important tags for Full EMV Refund include the following:

Tag	Contents
T57	Track 2 equivalent data.
T9C	Transaction type. • "20" indicates refund transaction. This tag is only used during EMV full refund transactions and is included in the EMV '33.03.x' Authorization Request and EMV '33.05.x' Authorization Confirmation Response messages.

Tag	Contents
D1003	Approval status. • "A" = Approved (used for EMV full refund, not for EMV partial refund). • "D" = Declined (used for EMV full refund, not for EMV partial refund). • "C" = Completed (used for EMV partial refund). • "E" = Error/incomplete.
D1005	"01" indicates a refund transaction.

When processing an EMV full refund transaction, the approval status will be included in tag D1003 as "A" (Approved) or "D" (Declined). Since EMV partial refund transactions are not host approved, tag D1003 will instead have a value of "C" indicated completion.

There are two types of EMV partial refund transactions; normal flow and on-demand flow. For normal flow refund transactions, the transaction is automatically initiated when an EMV card is inserted and detected. For on-demand flow refund transactions, the terminal sends a message to the POS indicating that a card was inserted. It then waits for the POS to return an EMV '33.00.x' Transaction Initiation Message before initiating the transaction.

Refer to the following **transaction flow** sections for more details, including messages exchanged between the terminal and POS:

- EMV Full Refund Transaction Flow
- EMV Partial Refund Normal Transaction Flow
- EMV Partial Refund On-Demand Transaction Flow

8.7.3 Configuring for Full or Partial Refund

The POS has the ability to select full or partial EMV transactions by setting a new flag in the `cards.dat` file. The EMV Refund Option flag works as follows:

- When set to '0' - Partial EMV refund transactions are processed.
- When set to '1' - Full EMV refund transactions are processed.

A 04.x Set Payment Type Request message now follows the 13.x Amount message in the refund transaction. This message includes the payment type configuration with the EMV Refund Option flag which defines the transaction as a partial or full EMV refund. Refer to Card Configuration (cards.dat) for more information on configuring this flag.

8.8 EMV with P2PE Enabled

8.8.1 EMV Tags Used with P2PE Enabled

When an EMV transaction is in process (contact or contactless), the EMV 33.03.x Authorization Request Message is used in place of the 50.x Authorization Request message. See EMV Tag Encryption for how to handle the tags that should be included in this message.

Additionally, the following Ingenico-specific tags are added:

- DFF1D

- DFF1E
- DFF1F
- DFF20
- DFF21

The tags provided depend on the card data and encryption type. As an example, tag `DFF20` is present only when Voltage encryption (TEP1, TEP2) is enabled.

For the EMV 33.02.x Track 2 Equivalent Data Message and EMV 33.05.x Authorization Confirmation Response Message, the Track 2 value will be replaced by the masked value.

8.8.2 Encrypting PAN-Related Data

All PAN-related data from an EMV card is to be encrypted when any of the following encryption methods are enabled:

- EPS (Element Payment Systems) P2PE encryption
- Generic TDES DUKPT encryption
- Magtek encryption for NKP4 build
- Mercury encryption
- Monetra encryption
- RSA-OAEP and TransArmor encryption
- S1 encryption
- Voltage TEP1 encryption
- Voltage TEP2 encryption

For more information on these encryption methods refer to P2PE Card Data Encryption .

8.8.3 EMV Tag Encryption

The following table shows how certain tags are handled during encryption for an EMV transaction.

8.8.3.1 EMV Tag Handling During Encryption

When E2EE encryption is enabled, the following functions apply to EMV tags:

E2EE EMV Tag Handling

Tag	Tag Name	With E2EE Encryption Enabled
T56	Track 1 data	Encrypted Track 1 data (if present). Tag optional. This optional field is defined by MasterCard only for its contactless cards.

Tag	Tag Name	With E2EE Encryption Enabled
T57	Track 2 equivalent data	Masked Track 2 data. Tag required.
T5A	PAN	Masked PAN data. Tag optional.
T5F24	Expiry date	Returned in the clear.
T5F30	Service code	Returned in the clear.
T9F6B	Track 2 Data for Contactless MasterCard	Encrypted Track 2 data for contactless MasterCard transactions. Tag required when performing a contactless MasterCard transaction.

Ingenico-Specific EMV Tags

Tag	Content
DFF1D	Masked PAN
DFF1E	Encrypted Track 1 Data
DFF1F	Encrypted Track 2 Data
DFF20	ETB (Voltage encryption only)
DFF21	Encrypted Track 3 Data

Tag Usage by Encryption Type

	DFF1D (PAN)	DFF1E (Track 1)	DFF1F (Track 2)	DFF20 (ETB)	DFF21 (Track 3)
On-Guard	Masked Pan	N/A	Encrypted Track 1, 2	N/A	N/A
Voltage (all)	Encrypted Pan	Encrypted Track 1	Encrypted Track 2	ETB	N/A
RSA-OAEP	Masked Pan	Masked Track 1	Masked Track 2	N/A	N/A
Trans Armor	Masked Pan	Masked Track 1	Masked Track 2	N/A	TransArmor-specific data (See Note)
S1	Masked Pan	Masked Track 1	Masked Track 2	N/A	Encrypted Track 3
TDES DUKPT	Masked Pan	Masked Track 1	Masked Track 2	N/A	Encrypted Track 3

	DFF1D (PAN)	DFF1E (Track 1)	DFF1F (Track 2)	DFF20 (ETB)	DFF21 (Track 3)
EPS	Masked Pan	Encrypted Track 1	Encrypted Track 2	N/A	N/A
Monetra	Masked Pan	Masked Track 1	Masked Track 2	N/A	Encrypted Track 3

TransArmor tag DFF21 (Track 3) consists of three items separated by colons:

- The 344-byte Base64 string
- One digit indicating what data was encrypted (1 = Track 1, 2 = Track 2, 3 = PAN for manually entered data)
- Key ID from the `security.dat` file

8.9 EMV Configuration and Flow

8.9.1 Configuration

The terminal is configured for EMV transactions by setting EMV Flag 0019_0001 (Enable EMV Transactions) to 1 in the `config.dfs` file. The terminal can be configured for Online PIN mode by setting tag `T9F33` (Terminal Capabilities) in the EMVCONTACT.XML file to E0 48 C8. This tag is normally set to E0 B8 C8.

8.9.2 Initiate Transaction

Since on-demand processing is the same as the processing for regular flow once a transaction is initiated, the regular flow will be discussed in this section. The sequence is as follows:

1. Initiate a new transaction, including the transaction amount.
2. Insert the EMV card.
3. Confirm (or select) the Application (e.g., VISA, MasterCard).
4. An EMV 33.00.x Transaction Initiation Message should be sent to the POS.
5. Set the payment type to credit (e.g., 04.0B1025 to request credit payment in the amount of $10.25).
6. Send the 13.x Amount Message (e.g., 13.1025 for $10.25).
7. Confirm the purchase amount.
8. Enter PIN when prompted.
9. An EMV 33.03.x Authorization Request Message is sent to the POS. This message contains a `T99` PIN Block tag.

8.9.3 Process Online PIN

Tag T99 supports the DUKPT encrypted PIN Block as well as the Master/Session encrypted PIN Block. You can extract the DUKPT encrypted PIN Block from tag `T99` which will be included in the EMV 33.03.x Authorization Request Message. This PIN Block may be padded with F's, if required. Refer to the following example DUKPT encrypted PIN Block tag:

T99:24:a4228728160715440FFFF9876543210E0004F

Once the required data is sent for online processing, the POS receives the authorization result. This result is sent to the terminal in the EMV 33.04.x Authorization Response Message. From this point, the transaction continues with the standard EMV flow.

Refer to PIN Block Tag Format in Authorization Request Message for more information on the PIN Block tag format.

8.9.4 PIN Block Tag Format in Authorization Request Message

The T99 tag included in the EMV '33.03.x' Authorization Request Message provides the Master/Session encrypted PIN Block or DUKPT encrypted PIN Block and KSN required to obtain the PIN used for online validation. The following table provides a description of this tag.

PIN Block Tag Format in Authorization Request Message

Offset	Length	Format	Description
0	3	Constant	Tag ID "T99"
3	1	Constant	Separator character (colon) ":"
4	2	Alphanum	Data length in hex format (e.g., 24 hex = 36 decimal).
6	1	Constant	Separator character (colon) ":"
7	1	Alphanum	Format Code. • "a" specifies ASCII. • "h" specifies hex ASCII.
8	16	Alphanum	Master/Session or DUKPT Encrypted PIN Block.
24	20	Alphanum	KSN.

8.9.5 EMV Configuration Parameters

The following sections provide more detailed information on parameters and tags used in EMV transactions:

- Notes on EMV Configuration Parameters
- EMV Flags (emv.dat)
- Application ID (AID) Parameters in EMVCONTACT.XML
- Application ID (AID) Tags in EMVCLESS.XML
- Certificate Authority Public Keys in EMVCONTACT.XML
- ICS Tags in EMVCONTACT.XML

- ICS Tags in EMVCLESS.XML

8.9.5.1 Notes on EMV Configuration Parameters

8.9.5.1.1 Merchant and Acquirer Responsibilities and Parameter Management

Ingenico and other terminal vendors cannot define terminal EMV parameters on behalf of acquirers for the majority of parameters described in this document. It is the acquirer's responsibility to define the appropriate settings.

As part of our testing and QA process, Ingenico uses sample configuration files. When a terminal is deployed to production, the replacement of these test values with production values, and the ongoing management of these values, is the responsibility of the merchant and their acquirer/gateway/processor. EMV parameters, such as Floor Limits and Terminal Action Codes, are set according to the acquirer's willingness to accept risk. It is the responsibility of the acquirer to consult with the card associations and to determine and set EMV parameters that are appropriate for their business model.

8.9.5.1.2 Configuration Files

The application separates the configuration for contact EMV and contactless EMV payment into two different files; `EMVCONTACT.XML` and `EMVCLESS.XML`.

Default Values

- Within an XML file, some parameters must be defined for each application ID. Many parameters can be defined in the ICS section and default to the ICS values unless otherwise specified for an application ID. Regardless of where the parameter is listed in this document, the ability to default to ICS values or assign application ID-specific values is the same. The tables in the following sections indicate which parameters are assigned an ICS Default value (refer to the second column in each table).
    - Application ID (AID) Tags in EMVCLESS.XML
    - Application ID (AID) Parameters in EMVCONTACT.XML
    - ICS Tags in EMVCLESS.XML
    - ICS Tags in EMVCONTACT.XML

8.9.5.1.3 Important

These parameters are used to configure the various contactless kernels, and in some cases, **they have an effect regardless of whether or not EMV is enabled**.

8.9.5.1.4 Data element format conventions

The configuration parameter values in `EMVCONTACT.XML` and `EMVCLESS.XML` are represented as sequences of two-digit numbers (hexadecimal or decimal). Interpretation of these sequences depends on the tag. For a list of formats, refer to *EMV Book 3, Data Element Format Conventions*. The following table describes the main formats used in this document.

Data Element Formats

Format	Description
an	Alphanumeric data elements contain a single character per byte. The permitted characters are alphabetic (a to z and A to Z, upper and lower case) and numeric(0 to 9).

Format	Description
ans	Alphanumeric special data elements contain a single character per byte. The permitted characters and their coding are shown in the Common Character Set table in **Annex B of Book 4**. There is one exception: The permitted characters for Application Preferred Name are the non-control characters defined in the ISO/IEC 8859 part designated in the Issuer Code Table Index associated with the Application Preferred Name.
b	These data elements consist of either unsigned binary numbers or bit combinations that are defined elsewhere in the specification. **Binary example**: The Application Transaction Counter (ATC) is defined as b with a length of two bytes. An ATC value of 19 is stored as Hex 00 13. Use caution with bit flags because different specifications may use different bit-numbering conventions. In some cases, bit 1 is the low-order bit; in other cases, bit 8 is the low-order bit.
n	Numeric data elements consist of two numeric digits (having values in the range Hex 0 – 9) per byte. These digits are right-justified and padded with leading hexadecimal zeroes. Other specifications might refer to this data format as Binary Coded Decimal (BCD) or unsigned packed. **Example**: Amount, Authorized (Numeric) is defined as n 12 with a length of six bytes. A value of 12345 is stored in Amount, Authorized (Numeric) as Hex 00 00 00 01 23 45.

8.9.5.2 *Application ID (AID) Parameters in EMVCONTACT.XML*

Application IDs (AIDs) are grouped under the tag `T1000` with individual tag numbers (e.g., `T1001`, `T1002`). Additionally, each AID is identified by tag `T9F06`. The following table lists the parameters included for each AID.

Application ID Parameters

Parameter	ICS Default	Description	Format
T9FFF00	No	User-defined name for an application ID. For Example: 56 49 53 41 = VISA	Hexadecimal ASCII characters
T9F06	No	Application ID (AID). Used to match the AID configured on the EMV card. When defined under node `T1000`, the first byte in the field indicates the length of the AID. The following example illustrates a tag with a length of seven bytes: 07 A0 00 00 00 03 10 10	Hexadecimal Five - 16 Bytes

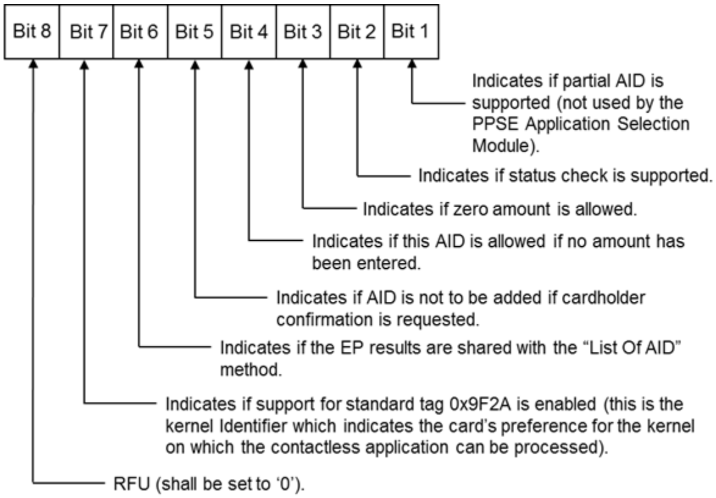
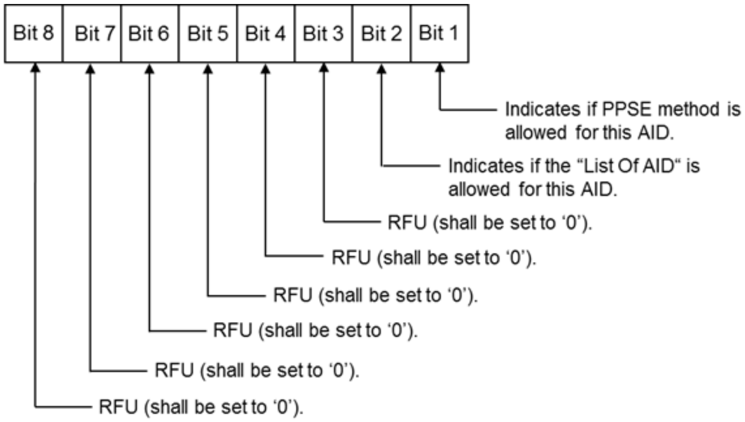
Parameter	ICS Default	Description	Format
T9F1A	Yes	Country code per ISO 3166 (International Standard for Country Codes). Country codes are at http://en.wikipedia.org/wiki/ISO_3166-1_numeric. For example: • 01 24 = Canada • 08 40 = United States	EMV Format n 3 Two bytes
T5F2A	Yes	Currency code per ISO 4217 (International Standard for currency). Currency codes are at http://en.wikipedia.org/wiki/ISO_4217. For example: • 08 26 = United Kingdom pound sterling. • 08 40 = U.S. dollar. • 01 24 = Canadian dollar.	EMV Format n 3 Two bytes
T5F36	Yes	Currency exponent. Indicates the implied position of the decimal point from the right of the transaction amount represented in accordance with ISO 4217. For example: 02 indicates that there are two digits to the right of the decimal point.	EMV Format n 1 One byte
T9F812B	Yes	Threshold value for biased random selection. This amount must be zero or a positive number which is less than the floor limit. Any transaction with an amount less than this value will be subject to selection at random based on the value of tag `T9F8127`. Refer to EMV Book 3, Random Transaction Selection. For example: 00 00 03 E8 (hexadecimal) = 1000 (decimal) = $10.00 in U.S currency.	EMV Format b Binary Four bytes
T9F8124	Yes	Default Dynamic Data Authentication Data Object List (DDOL). Specified by the payment system and used when a DDOL is not present on the EMV card. For example: 9F 37 04	EMV Format b Binary Variable
T9F8125	Yes	Default Transaction Certificate Data Object List (TDOL). Specified by the payment system and used when a TDOL is not present on the EMV card. For example: 9F 02 06 95 05 5F 2A 02 9A 03 9C 01 9F 37 04	EMV Format b Binary Variable

Parameter	ICS Default	Description	Format
T9F8126	Yes	Maximum target percentage used for biased random selection. The value for this parameter must be between 0 to 99 and no less than the value of tag T9F8127. Refer to *EMV Book 3, Random Transaction Selection*. For example: 32 (hexadecimal) = 50 (decimal) = 50%	EMV Format b Binary One byte
T9F8127	Yes	Target Percentage to be Used for Random Selection. The value must be between 0 to 99. Refer to *EMV Book 3, Random Transaction Selection*. For example: 32 (hexadecimal) = 50 (decimal) = 50%	EMV Format b Binary One byte
T9F8128	Yes	Terminal Action Code (TAC) - Default. Specified by the acquirer. Refer to *EMV Book 3, Terminal Action Analysis*. For example: DC 40 00 A8 00	EMV Format b Binary Five bytes
T9F8129	Yes	Terminal Action Code (TAC) - Denial. Specified by the acquirer. Refer to *EMV Book 3, Terminal Action Analysis*. For example: 00 10 00 00 00	EMV Format b Binary Five bytes
T9F812A	Yes	Terminal Action Code (TAC) - Online. Specified by the acquirer. Refer to *EMV Book 3, Terminal Action Analysis* for an explanation of this parameter. For example: DC 40 04 F8 00	EMV Format b Binary Five bytes
T9F09	Yes	Version number assigned by the payment system for the application. For example: 00 8C	EMV Format b Binary Two bytes
T9F1B	Yes	Terminal floor limit. Transaction amounts in excess of the floor limit may require the transaction to be done online. Refer to *EMV Book 3, Floor Limits*. For example: 00 00 27 10 (hexadecimal) = 10000 (decimal) = $100.00 in US currency.	EMV Format b Binary Four bytes
T9F841D	Yes	Allow partial name selection (Application Selection Indicator). For an application on the EMV card to be supported, this parameter indicates whether the associated application ID in the terminal must exactly match the application ID in the card, including the length or only up to the length of the application ID in the terminal. For example: 01	EMV Format b Binary One byte

8.9.5.3 Application ID (AID) Tags in EMVCLESS.XML

Application ID Tags in EMVCLESS.XML

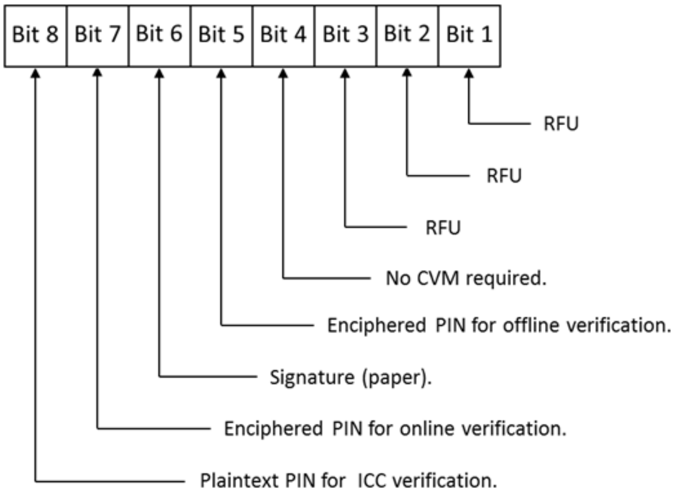
Tag	ICS Default	Description	Format
AID: Maestro PayPass AID for Debit			
T9F06	No	This tag identifies the application. A0 00 00 00 03 10 10	Hexadecimal (Variable)
T9F928101	No	This specifies the contactless kernel to use for the given Application ID. Available kernels include: • 00 02 = MasterCard (PayPass M/Chip and magstripe • 00 03 = VISA (PayWave qVSDC and magstripe) • 00 04 = American Express (ExpressPay EMV and magstripe) • 01 02 = Discover DPAS • 01 03 = Interac	Two bytes

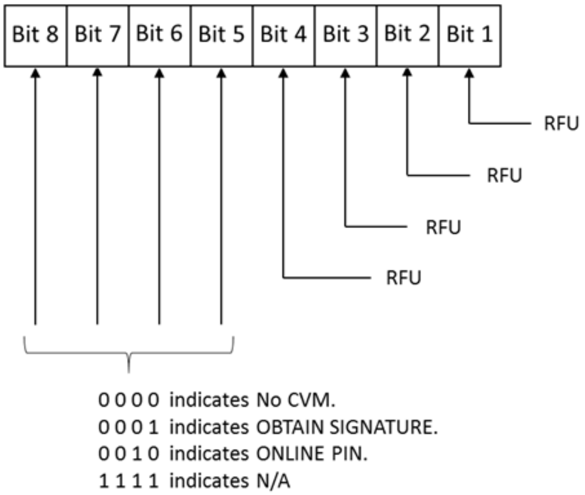
Tag	ICS Default	Description	Format
T9F928100	No	Proximity Payment System Environment (PPSE) Application Selection, which indicates the Application ID options: **Byte 1**:  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Indicates if partial AID is supported (not used by the PPSE Application Selection Module). Indicates if status check is supported. Indicates if zero amount is allowed. Indicates if this AID is allowed if no amount has been entered. Indicates if AID is not to be added if cardholder confirmation is requested. Indicates if the EP results are shared with the "List Of AID" method. Indicates if support for standard tag 0x9F2A is enabled (this is the kernel Identifier which indicates the card's preference for the kernel on which the contactless application can be processed). RFU (shall be set to '0'). **Byte 2**:  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Indicates if PPSE method is allowed for this AID. Indicates if the "List Of AID" is allowed for this AID. RFU (shall be set to '0'). RFU (shall be set to '0'). RFU (shall be set to '0'). RFU (shall be set to '0'). RFU (shall be set to '0'). RFU (shall be set to '0'). **Byte 3 and Byte 4**: RFU (shall be set to 0). For example: 05 03 00 00 indicates Partial AID supported, zero amount allowed, PPSE and List of AID methods allowed.	Four bytes

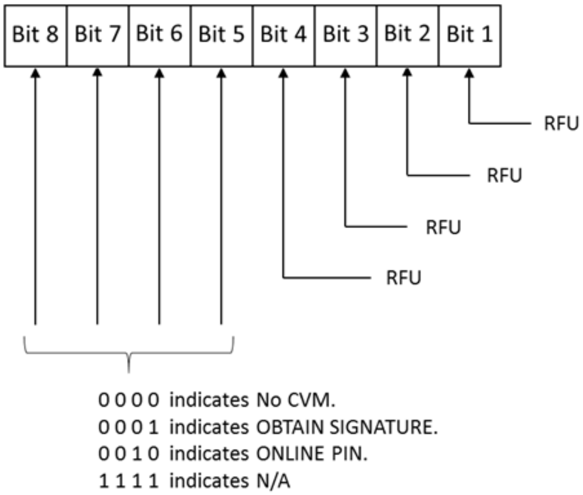
Tag	ICS Defa ult	Description	Format
T9F92810E	No	CVM Required Limit. This value is inclusive. For example: 00 00 00 00 20 00 = $20.00. The transaction amount of $20.00 requires a CVM. A transaction amount off $19.99 does not require a CVM.	EMV Format "n 12" Six bytes
T9F92810F	No	Contactless Floor Limit. This value is exclusive. For example: 00 00 00 00 20 00 = $20.00. A floor limit of $20.01 is supported, but a floor limit of $20.00 is not. See Floor Limit Handling for more information.	EMV Format "n 12" Six bytes
T9F1B	Yes	Terminal Floor Limit. Transaction amounts exceeding the floor limit may require the transaction to be completed online. Refer to EMV Book 3, *Floor Limits*, for an explanation of this parameter. For example: 00 00 27 10 (hexadecimal) = 10000 (decimal) = $100.00 in U.S. currency.	EMV Format "b" (Binary) Four bytes
T9F918502	No	PayPass Default UDOL (PayPass only). For example: 9F 6A 04 - this fixed value is defined in the PayPass specification.	
T9F918709	No	Terminal Action Code -Default. This specifies the acquirers conditions which result in a transaction being rejected if it may have been approved online, but the terminal is unable to process the transaction online. Refer to EMV Book 3, *Terminal Action Analysis*, for an explanation of this parameter. For example: FC 50 0C 88 00	EMV Format "b" (Binary) Five bytes
T9F91870A	No	Terminal Action Code - Denial. This specifies the acquirers conditions which result in the denial of a transaction without attempting to go online. Refer to EMV Book 3, Terminal Action Analysis, for an explanation of this parameter. For example: 00 00 00 00 00	EMV Format "b" (Binary) Five bytes

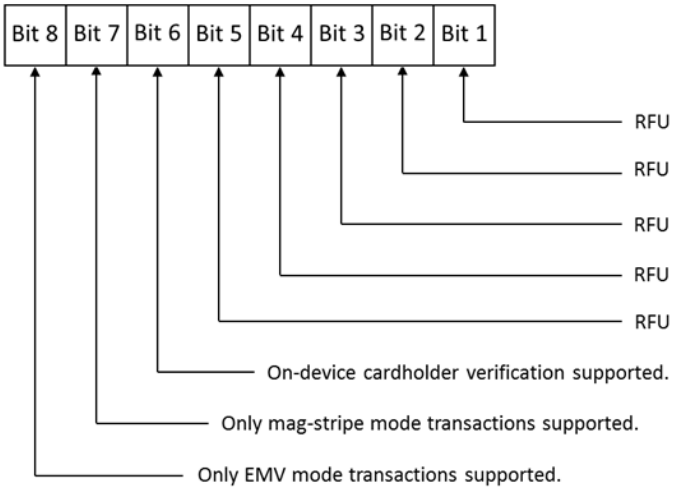
Tag	ICS Default	Description	Format
T9F91870B	No	Terminal Action Code - Online. This specifies the acquirers conditions which result in a transaction being transmitted online. Refer to EMV Book 3, "Terminal Action Analysis," for an explanation of this parameter. For example: FC 50 0C 88 00	EMV Format "b" (Binary) Five bytes
T9F53	No	PayPass Transaction Category Code (PayPass only). This is a data object defined by MasterCard which indicates the current transaction type. This may be used during the Card Risk Management step in the EMV transaction process. For example: 52 = "R"	EMV Format "an" One byte
T9F918706	No	Default Transaction Certificate Data Object List (TDOL). This is specified by the payment system and is to be used if a TDOL is not present on the EMV card. For example: 9F 08 02	EMV Format "b" (Binary) (Variable)
T9F91841D	No	Terminal Supported Languages. For example: 65 6e 65 73 = "enes" where 65 6e = "en" (English) and 65 73 = "es" (Espanol, or Spanish).	Hex ASCII (Two - Eight Bytes)
Tags for PayPass 2 Only			
T9F918504	No	PayPass Terminal Capabilities with CVM Required (PayPass only). Refer to the preceding description for `T9F33`. For example: E0 68 C8 indicates Online PIN, Signature, and No CVM.	
T9F918505	No	PayPass Capabilities with No CVM Required (PayPass only). Refer to the preceding description for `T9F33`. For example: E0 68 C8 indicates No CVM.	
T9F92810D	No	Transaction Limit. This value is exclusive, so the limit value is not allowed. For example: 00 00 00 00 20 00 = $20.00. In this scenario, a transaction amount of $19.99 is allowed, but a transaction amount of $20.00 is not.	EMV Format "n 12" Six bytes

Tag	ICS Default	Description	Format
T9F918503	No	PayPass Magstripe Indicator (PayPass only). • 00 = Magstripe profile is not allowed for this AID. • 01 = Magstripe profile is allowed for this AID. For example, 01 indicates that Magstripe profile is allowed for this AID.	
T9F91850D	No	List of Application Version Numbers (AVNs) for PayPass Magstripe (PayPass only). For example: 01 06 02 01 02 06 indicates versions 0106, 0201, and 0206.	(Variable)
T9F918511	No	List of Application Version Numbers (AVNs) for PayPass M/Chip (PayPass only). For example: 01 05 02 00 02 05 indicates versions 0105, 0200, and 0205.	(Variable)
T9F91851B	No	PayPass Magstripe Only Indicator for PayPass 2. • 00 = Do not force Magstripe only for this AID. • 01 = Force Magstripe only for this AID. For example, 01 indicates that Magstripe only is forced for this AID. If MasterCard PayPass EMV functionality will not be used in a given installation, this tag should be set to 01 to avoid any attempts to interact with a card or mobile terminal while in EMV mode.	
Tags for PayPass 3 Only			
T9F918523	No	Indicates the card data input capability of the Terminal and Reader. • MasterCard PayPass 3 kernel only. • Tag is coded per Annex A.2 of EMV Book 4 (Terminal Capabilities), byte 1. • Corresponds with tag DF8117 value in the MasterCard Kernel C2 specification.	EMV Format "b" (Binary) One byte

Tag	ICS Default	Description	Format
T9F918524	No	Indicates the security capability of the kernel. • MasterCard PayPass 3 kernel only. • Corresponds with tag DF811F value in the MasterCard Kernel C2 specification. • Tag is coded per Annex A.2 of EMV Book 4 (Terminal Capabilities), byte 3:	EMV Format "b" (Binary) One byte
T9F918525	No	Indicates the MChip CVM Capability of the Terminal and Reader when the transaction amount is greater than the Reader CVM Required Limit. • MasterCard PayPass 3 kernel only. • Corresponds with tag DF8118 value in the MasterCard Kernel C2 specification. • Tag is coded per Annex A.2 of EMV Book 4 (Terminal Capabilities), byte 2:  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 RFU RFU RFU No CVM required. Enciphered PIN for offline verification. Signature (paper). Enciphered PIN for online verification. Plaintext PIN for ICC verification. For example: 28 indicates signature only with no CVM.	EMV Format "b" (Binary) One byte
T9F918526	No	PayPass MChip CVM capability, CVM not required. For example, 08 = No CVM only.	EMV Format "b" (Binary) One byte

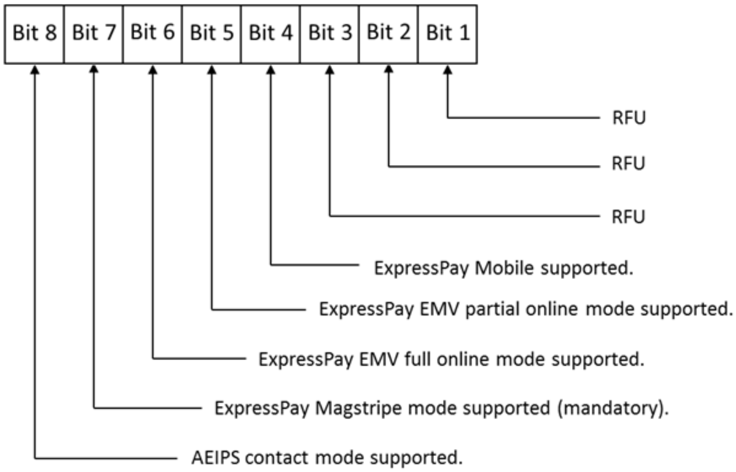
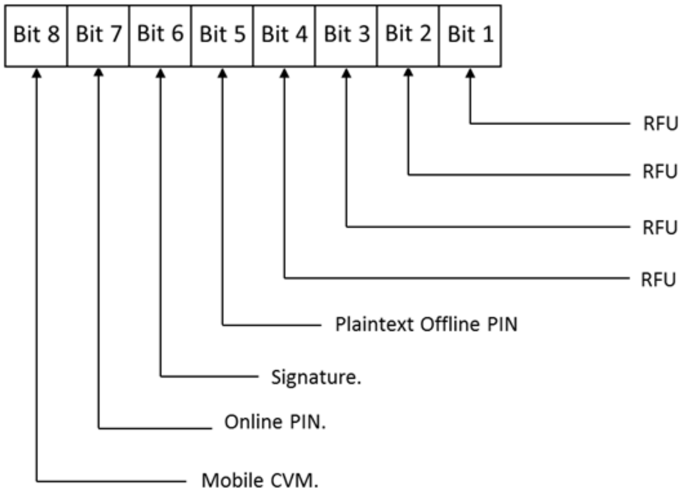
Tag	ICS Default	Description	Format
T9F918527	No	Indicates magnetic stripe CVM Capability of the Terminal and Reader when the transaction amount is greater than the Reader CVM Required Limit. - MasterCard PayPass 3 kernel only. - Corresponds with tag DF811E value in the MasterCard Kernel C2 specification.  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 RFU RFU RFU RFU 0 0 0 0 indicates No CVM. 0 0 0 1 indicates OBTAIN SIGNATURE. 0 0 1 0 indicates ONLINE PIN. 1 1 1 1 indicates N/A For example: 10 indicates that signature is required.	EMV Format "b" (Binary) One byte

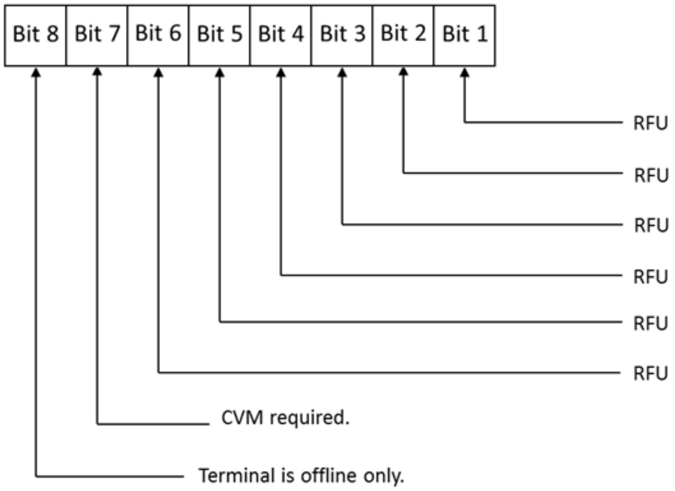
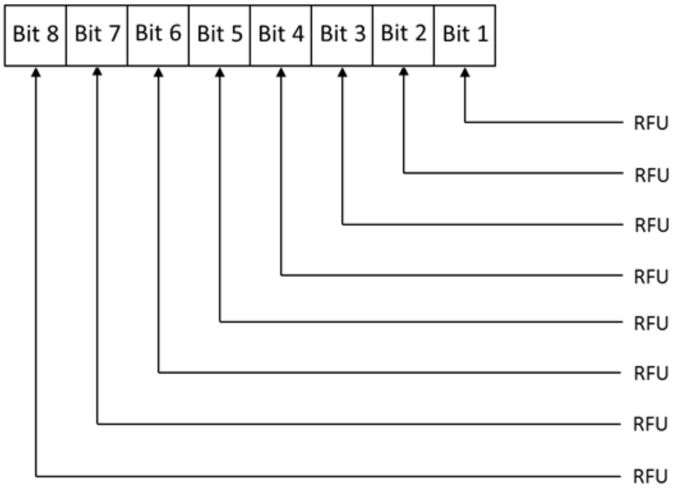
Tag	ICS Default	Description	Format
T9F918528	No	Indicates the Magstripe CVM Capability of the Terminal and Reader when the transaction amount is less than or equal to the Reader CVM Required Limit. • MasterCard PayPass 3 kernel only. • Corresponds with tag DF812C value in the MasterCard Kernel C2 specification.  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 RFU RFU RFU RFU 0 0 0 0 indicates No CVM. 0 0 0 1 indicates OBTAIN SIGNATURE. 0 0 1 0 indicates ONLINE PIN. 1 1 1 1 indicates N/A For example: 00 indicates no CVM is required.	EMV Format "b" (Binary) One byte
T9F91851C	No	Contactless transaction limit for cards (not for mobile devices). This tag indicates the maximum allowed transaction amount when on-device cardholder verification is not supported. • MasterCard PayPass 3 kernel only. • Corresponds with tag DF8124 value in the MasterCard Kernel C2 specification.	EMV Format "b" (Binary) Six bytes
T9F91851D	No	Contactless transaction limit for mobile devices (not for cards). This tag indicates the maximum allowed transaction amount when on-device cardholder verification is supported. • MasterCard PayPass 3 kernel only. • Corresponds with tag DF8125 value in the MasterCard Kernel C2 specification.	EMV Format "b" (Binary) Six bytes

Tag	ICS Default	Description	Format
T9F918522	No	Indicates the kernel configuration options. - MasterCard PayPass 3 kernel only. - This corresponds with tag DF811B value in the MasterCard kernel C2 specification.  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 RFU RFU RFU RFU RFU On-device cardholder verification supported. Only mag-stripe mode transactions supported. Only EMV mode transactions supported. For example: 20 indicates that both EMV mode and magnetic stripe mode are supported along with on-device verification.	EMV Format "b" (Binary) One byte
T9F6D	No	This tag is used for ExpressPay and PayPass. For ExpressPay, this tag indicates the terminals capability to support ExpressPay Magstripe or EMV Contactless where - 00 = ExpressPay 1.0. - 40 = ExpressPay 2.0 Magstripe only. - 80 = ExpressPay 2.0 EMV and Magstripe. - C0 = ExpressPay Mobile. For PayPass, this tag indicates the version number assigned by the payment system for the specific PayPass Magstripe functionality of the application. - Default value = 00 01.	ExpressPay: EMV Format "b" (Binary) One byte PayPass: EMV Format "b" (Binary) Two bytes

Tag	ICS Default	Description	Format
T9F09	No	EMV Application version Number for the terminal. - Default = 00 02	EMV Format "b" (Binary) Two bytes
T9F918565	No	Maximum time (in seconds) that a record can remain in the Torn Transaction Log. - MasterCard PayPass 3 kernel only. - Corresponds with tag DF811C value in the MasterCard Kernel C2 specification. For example: 01 2C = 300 seconds	EMV Format "b" (Binary) Two bytes
T9F918561	No	PayPass maximum number of records that can be contained in the torn log. - Default = 00	EMV Format "b" (Binary) One byte
T9F91852B	No	PayPass default hold time in seconds. - Default = 00 00 13 = 1.9 seconds	EMV Format "b" (Binary) Three bytes
T9F918568	No	Indicates the time that the field is turned off after the transaction is completed if requested to do so by the cardholder device. - MasterCard PayPass 3 kernel only. - Corresponds with tag DF8130 value in the MasterCard Kernel C2 specification. - Hold time value is in units of 100ms. For example: 0D = 1300 ms = 1.3 seconds	EMV Format "b" (Binary) One byte
T9F91854C	No	PayPass timeout value. - Default = 01 F4 = 12 seconds.	EMV Format "b" (Binary) Two bytes

Tag	ICS Default	Description	Format
T9F1D	No	Terminal Risk Management Data. This tag contains application-specific data which is used by the card to perform terminal risk management. • Currently only used for MasterCard PayPass. • Implementation-dependent as it corresponds with CVMs used by the merchant. • Should be set to all 0s when using the PayPass 2 kernel. For example, 00 00 00 00 00 00 00 00	EMV Format "b" (Binary) (8 Bytes)
Tags for Express Pay 3 Only			

Tag	ICS Default	Description	Format
T9F6E	No	ExpressPay terminal capabilities and CVM capabilities. Currently used for ExpressPay kernel 3. Byte 1 - terminal Capabilities:  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 RFU RFU RFU ExpressPay Mobile supported. ExpressPay EMV partial online mode supported. ExpressPay EMV full online mode supported. ExpressPay Magstripe mode supported (mandatory). AEIPS contact mode supported. Byte 2 - Terminal CVM Capabilities:  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 RFU RFU RFU RFU Plaintext Offline PIN Signature. Online PIN. Mobile CVM. Byte 3 - Transaction Capabilities:	EMV Format "b" (Binary) Four bytes

Tag	ICS Default	Description	Format
		 Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 RFU RFU RFU RFU RFU RFU CVM required. Terminal is offline only. Byte 4 - Transaction Capabilities  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 RFU RFU RFU RFU RFU RFU RFU RFU For example: D8 B0 00 00 = AEIPS contact, ExpressPay Magstripe, ExpressPay EMV Partial online mode, ExpressPay Mobile, Mobile CVM, Signature, Plaintext Offline PIN.	
Additional Tags			

Tag	ICS Default	Description	Format
T9F66	No	Terminal Transaction Qualifiers (TTQ). Currently used only for VISA. This indicates card reader capabilities, requirements, and preferences to the card. - TTQ byte 2, bits 8-7 are transient values. These are reset to 0 at the start of the transaction. - All other TTQ bits are static values which are not modified based on transaction conditions. - TTQ byte 3, bit 7 shall be set by the acquirer merchant to "1b." Refer to https://www.eftlab.com.au/index.php/site-map/our-articles/ 161-the-use-of-ctqs-and-ttqs-in-nfc-transactions. **Example supporting Contactless EMV**: B2 A0 40 00 indicates - Contactless MSD. - qVSDC supported. - EMV contact chip supported. - Signature supported. - Online cryptogram required. - Offline PIN supported (for contact chip). - Mobile functionality supported (consumer terminal CVM). **Example supporting Contactless MSR only:** 86 A0 40 00 indicates - Contactless MSD is supported. - Online PIN supported. - Signature supported. - Online cryptogram required. - Offline PIN supported (for Contact chip). - Mobility functionality supported (Consumer terminal CVM).	EMV Format "b" (Binary) Four bytes

Tag	ICS Default	Description	Format
		If VISA PayWave EMV functionality will not be used in a given installation, bit 6 of byte 1 (Contactless qVSDC supported) should be set to 0 in order to avoid any attempts to interact with a card or mobile terminal in EMV mode.	
T9F1A	Yes	Country Code per ISO 3166. Country codes may be referenced at http://en.wikipedia.org/wiki/ISO_3166-1_numeric. As examples: • 01 24 = Canada • 08 40 = United States	EMV Format "n 3" Two bytes
T9F2A	Yes	Transaction Currency Code per ISO 3166. Country codes may be referenced at http://en.wikipedia.org/wiki/ISO_3166-1_numeric. As examples: • 01 24 = Canadian Dollar • 08 40 = United States Dollar	EMV Format "n 3" Two bytes
T9FA15D	No	EMV Scheme Label. For example: 44 45 42 49 54 indicates debit.	Hex ASCII
T9F91831E	No	MSD Disable CVN17 flag (PayWave only), where CVN17 is an enhanced online card authentication feature of VISA PayWave. • 00 = No, do not disable CVN17. • 01 = Yes, disable CVN17.	
T9F918307		Supported FDDA versions (PayWave only). For example: 00 01	
T9F918A11	No	List of Application Version Numbers (AVNs) for Interac (Interac only). For example: 00 02 indicates version 0002.	

Tag	ICS Default	Description	Format
T9F918A04	No	Interac Terminal Capabilities with CVM Required (Interac only). Refer to the preceding description for `T9F33`. For example: E0 08 00 indicates No CVM only, no CDA.	
T9F918A05		Terminal Action Code - Default (Interac only). For example: FC 68 FC F8 00	
T9F58	No	Interac Merchant Type Indicator (Interac only). Must be in the range from 01 to 05. For example: 03	
T9F59	No	Interac Terminal Transaction Information (Interac only). For example: C0 87 00	
T9F5D	No	Interac Receipt Limit (Interac only). For example: 00 00 00 00 00 00	EMV Format "n 12"
T9F5E	No	Interac Terminal Option Status (Interac only). For example: Default is 00 00 for U.S., E0 00 for Canada.	
T9F918200	No	ExpressPay Unpredictable Number Range (ExpressPay only). For example: 03 0C = 60 (decimal).	EMV Format "b" (Binary) Two bytes
T9F91820A	No	List of Application Version Numbers (AVNs) for ExpressPay (ExpressPay only). For example: 00 01 indicates version 0001.	
T9F91820F	No	ExpressPay Full Online EMV Removal Timeout (ExpressPay only). For example: 00 00 27 10 = 2710 (hex) = 10000 (decimal) = 10 seconds.	EMV Format "b" (Binary) Four bytes
T9F918A01	No	Interac Retry Limit (Interac only). Indicates the maximum number of attempts that are allowed before the transaction is rejected.	EMV Format "b" (Binary) One byte

Tag	ICS Default	Description	Format
T9F33	Yes	Terminal Capabilities. This indicates the `card.dat` input, CVM, and security capabilities of the terminal. Refer to EMV Book 4, *Terminal Capabilities*. For example: E0 B8 C8 where • E0 (Card Data Input Capability) = Manual key entry, magnetic stripe, IC with contacts. • B8 (CVM Capability) = Plaintext PIN for ICC verification, Signature (paper), Enciphered PIN for offline verification, No CVM required. • C8 (Security Capability) = SDA, DDA, CDA.	EMV Format "b" (Binary) Three bytes

8.9.5.3.1 Floor Limit Handling

The following table describes the two items used in floor limit handling in `EMVCLESS.XML`:

EMVCLESS.XML Items

Item	Tag Designation	Function
Contactless Floor Limit	T9F92810F TAG_EP_CLESS_FLOOR_LIMIT	Determines whether a transaction requires online authorization
Terminal Floor Limit	T9F1B TAG_EMV_TERMINAL_FLOOR_LIMIT	Entry Point spec and some card schemes state that if Contactless Floor Limit is not present, use the Terminal Floor Limit instead to determine whether a transaction requires online authorization.

The following table describes floor limit checking based on the kernel used to process the contactless transaction:

Floor Limit Checking by Kernel

Kernel	Floor Limit Checking
PayPass3 and ExpressPay3	During Terminal Risk Management, terminals compare the transaction amount to the Contactless Floor Limit T9F92810F if defined, and set the TVR accordingly. TAG_EMV_TERMINAL_FLOOR_LIMIT T9F1B is not supported.

Kernel	Floor Limit Checking
Visa PayWave and UPI QuickPass	The terminal indicates Online Cryptogram Required (set TTQ byte 2 bit 8 to 1b) if the amount authorized is greater than either: • The Contactless Floor Limit T9F92810F, or • The Terminal Floor Limit tag T9F1B if the Contactless Floor Limit is not present. TAG_EP_CLESS_FLOOR_LIMIT supersedes T9F1B.
Discover DPAS	If amount authorized is greater than: • The Contactless Floor Limit T9F92810F, the reader sets TTQ B2b8 to 1 and TVR B4b8 to 1. • The Terminal Floor Limit T9F1B, the reader sets TTQ B2b8 to 1 and TVR B4b8 to 1.
Discover ZIP	All ZIP transactions are authorized online. The floor limit is disregarded.
Interac FLASH	N/A - The Ingenico FLASH kernel is only certified for online transactions.

8.9.5.4 *Certificate Authority Public Keys in EMVCONTACT.XML*

The Certificate Authority (CA) Public Keys are grouped under node `1100`. Individual CA Public Keys tag numbers are a continuation of the group tag number (e.g., `1101`, `1102`). Each CA Public Key is identified by the Registered Application Provider Identifier (RID) and Application ID (AID) together with a key index, tag `T9F22`. The following table lists the parameters included for a given Certificate Authority Public Key (CAPK).

Certificate Authority Public Key Parameters

Tag	Description	Length	Format
0x9F06	Application ID. This is used to match the AID configured on the EMV card. As an example: 'A0 00 00 03 10 10' When tag `T9F06` is defined under node `1100`, it is actually used as an RID, and the length field used under node `1000` is not required.	5 to 16 Bytes	Hexadecimal

Tag	Description	Length	Format
0x9F22	Certificate Authority Public Key Index. This index, in conjunction with the AID, identifies the public key.	1 Byte	EMV Format "b" (Binary)
0x9F8123	Certificate Authority Public Key Modulus. Refer to sample `EMVCONTACT.XML` for examples.		EMV Format "b" (Binary)
0x9F8122	Certificate Authority Public Key Exponent. As an example for exponent 3: '03'	1 Byte for Exponent 3, 3 Bytes for Exponent 65537	EMV Format "b" (Binary)
0x9F8121	Certificate Authority Public Key Check Sum. SHA on Certificate Authority Public Key. As an example: 'EE 15 11 CE C7 10 20 A9 B9 04 43 B3 7B 1D 5F 6E 70 30 30 F6'	20 Bytes	

As an example, these AMEX CAPKs are included in the default `EMVCONTACT.XML` for contact EMV testing.

AMEX Contact EMV Testing Certificate Authority Public Keys

Key File Name	CAPK Index	CAPK Length
1152 PUBLIC KEY.txt	C1	'90' = 144 bytes = 1152 bits
1408 PUBLIC KEY.txt	C2	'B0' = 176 bytes = 1408 bits
1984 PUBLIC KEY.txt	C3	'F8' = 248 bytes = 1984 bits

8.9.5.5 ICS Tags in EMVCONTACT.XML

Implementation Conformance Statement (ICS) configurations are grouped under tag 1300. Individual ICS configuration tag numbers are a continuation of the group tag number (e.g., 1301, 1302). Typically there is only one ICS configuration. The following table lists the parameters included for a given ICS configuration.

ICS Tags in EMVCONTACT.XML

Tag	ICS Default	Description	Format
T9F8450	No	User-defined name for ICS configuration. As an example: '44 45 46 41 55 4C 54 20 43 4F 4E 46 49 47' = DEFAULT CONFIG"	Hex ASCII

Tag	ICS Default	Description	Format
T9F35	No	Terminal Type as defined by EMVCo. Refer to EMV Book 4, "Terminal Types." As an example: '22' = Attended, Offline with online capability, Operational control provided by merchant.	EMV Format "n 2" (1 Byte)
T9F33	Yes	Terminal Capabilities. This indicates the card data input, CVM,and security capabilities of the terminal. Refer to EMV Book 4, "Terminal Types." As an example: 'E0 B8 C8' where • 'E0' (Card Data Input Capability) = Manual key entry, magnetic stripe, IC with contacts. • 'B8' (CVM Capability) = Plaintext PIN for ICC verification, Signature (paper), Enciphered PIN for offline verification, no CVM required. • 'C8' (Security Capability) = SDA, DDA, and CDA.	EMV Format "b" (Binary) (3 Bytes)
T9F1A	Yes	Country code per ISO 3166. Codes may be referenced at http://en.wikipedia.org/wiki/ISO_3166-1_numeric. As examples: • '01 24' = Canada • '08 40' = United States	EMV Format "n 3" (2 Bytes)
T5F2A	Yes	Currency Code per ISO 4217. Codes may be referenced at http://en.wikipedia.org/wiki/ISO_4217. As examples: • '01 24' = Canadian Dollar • '08 40' = U.S. Dollar	EMV Format "n 3" (2 Bytes)

Tag	ICS Default	Description	Format
T9F40	Yes	Additional Terminal Capabilities. This parameter indicates the supported transaction types, data input, and data output capabilities of the terminal. Refer to EMV Book 4, "Additional Terminal Capabilities." As an example: 'F0 00 F0 A0 01' where • 'F0 '00' (Transaction Type Capability) = Cash, Goods, Services, Cashback. • 'F0' (Terminal Data Input Capability) = Numeric keys, Alphabetic and special character keys, Command keys, Function keys. • 'A0 01' (Terminal Output Capability) = Print/ attendant, Display/attendant, Code table 1.	EMV Format "b" (Binary) (5 Bytes)
T9F8142	No	Support PSE Selection Method. • '00' = No. • '01' = Yes. As an example: '01' configures to support PSE Selection Method.	(1 Byte)
T9F8195	No	Support Alternative Option to PSE Algorithm. • '00' = No. • '01' = Yes. As an example: '00' configures to not support Alternative Option to PSE Algorithm.	(1 Byte)
T9F844B	Yes	Support Cardholder Confirmation. • '00' = No. • '01' = Yes. As an example: '01' configures to support Cardholder Confirmation.	(1 Byte)
T9F8440	No	Display Application IDs in Preferred Order. • '00' = No. • '01' = Yes. As an example: '00' configures to not display Application IDs in Preferred Order.	(1 Byte)

Tag	ICS Default	Description	Format
T9F8441	Yes	Support Multiple Languages. • '00' = No. • '01' = Yes. As an example: '01' configures to support Multiple Languages.	(1 Byte)
T9F8442	Yes	Support Certificate Revocation. ***Reserved, not currently used.*** • '00' = No. • '01' = Yes. As an example: '01' configures to support Certificate Revocation.	(1 Byte)
T9F840A	Yes	Support Bypass PIN Entry. • '00' = No. • '01' = Yes. As an example: '01' configures to support Bypass PIN entry.	(1 Byte)
T9F844D	Yes	Support Get Data for PIN Try Counter. • '00' = No. • '01' = Yes. As an example: '01' configures to support Get Data for PIN Try Counter.	(1 Byte)
T9F8443	Yes	Amount is Known Before CVM Process. • '00' = No. • '01' = Yes. As an example: '01' configures for Amount is Known Before CVM Process.	(1 Byte)
T9F8444	Yes	Support Transaction Log. ***Reserved, not currently used.*** • '00' = No. • '01' = Yes. As an example: '01' configures to support Transaction Log.	(1 Byte)

Tag	ICS Default	Description	Format
T9F844C	Yes	Support Exception File. ***Reserved, not currently used.*** • '00' = No. • '01' = Yes. As an example: '01' configures to support Exception File.	(1 Byte)
T9F840E	Yes	Transaction Forced Online Capability. • '00' = No. • '01' = Yes. As an example: '01' configures for Transaction Forced Online Capability.	(1 Byte)
T9F840F	Yes	Transaction Forced Acceptance Capability. • '00' = No. • '01' = Yes. As an example: '01' configures for Transaction Forced Acceptance Capability.	(1 Byte)
T9F840B	Yes	Support Online Advice. • '00' = No. • '01' = Yes. As an example: '01' configures to support Online Advice.	(1 Byte)
T9F8445	Yes	Support Issuer Referral. • '00' = No. • '01' = Yes. As an example: '01' configures to support Issuer Referral.	(1 Byte)
T9F8446	Yes	Support Card Referral. • '00' = No. • '01' = Yes. As an example: '00' configures to not support Card Referral. Referrals initiated by card was removed by EMVCo Bulletin SU-42.	(1 Byte)

Tag	ICS Default	Description	Format
T9F8447	Yes	Support Batch Data Capture. ***Reserved, not currently used.*** • '00' = No. • '01' = Yes. As an example: '00' configures to not support Batch Data Capture.	(1 Byte)
T9F8448	Yes	Support Online Data Capture. • '00' = No. • '01' = Yes. As an example: '00' configures to not support Online Data Capture.	(1 Byte)
T9F844E	Yes	Support POS Entry Mode. • '00' = No. • '01' = Yes. As an example: '01' configures to support POS Entry Mode.	(1 Byte)

Tag	ICS Default	Description	Format
T9F39	Yes	Point-of-Service (POS) Entry Mode. This indicates the method by which the PAN was entered. This is determined by the first two digits of the ISO 8583:1987 POS Entry Mode. Specific values include: • '00' = Unknown. • '01' = Manually keyed (this will pertain to VISA internet transactions as well). • '02' = Magnetic stripe read (general or Track 2). • '04' = OCR code read. • '05' = Integrated circuit card read (CVV data is reliable). • '06' = Magnetic stripe read (Track 1). • '07' = Contactless chip card using VISA Smart Debit in accordance with Credit chip data rules. • '80' = Chip Card capable, unaltered track data read. This is used for EMV fallback where chip card is swiped. • '81' = Manually keyed e-commerce (MasterCard only). • '82' = Contactless Mobile Commerce terminal. • '90' = Entire magnetic stripe is read and transmitted. • '91' = Contactless chip transaction originated using magnetic stripe data rules (VISA only). • '95' = Integrated circuit card read, CVV data is unreliable. As an example: '81' indicates manually keyed e-commerce (MasterCard).	(1 Byte)
T9F8449	Yes	Terminal Equipped with External PIN Pad. • '00' = No. • '01' = Yes. As an example: '00' configures for Terminal not Equipped with External PIN Pad.	(1 Byte)

Tag	ICS Default	Description	Format
T9F844A	Yes	Amount and PIN Entered on Same Key Pad. • '00' = No. • '01' = Yes. As an example: '01' configures for Amount and PIN Entered on Same Key Pad.	(1 Byte)
T9F8452	Yes	Support Default Dynamic Data Authentication Data Object List (DDOL). • '00' = No. • '01' = Yes. As an example: '01' configures to support Default DDOL.	(1 Byte)
T9F8453	Yes	Perform Terminal Floor Limit Checking According to Terminal Type. • '00' = No. • '01' = Yes. As an example: '01' configures to perform Terminal Floor Limit Checking According to Terminal Type.	(1 Byte)
T9F8454	Yes	Perform Random Transaction Selection According to Terminal Type. • '00' = No. • '01' = Yes. As an example: '01' configures to perform Random Transaction Selection According to Terminal Type.	(1 Byte)
T9F8455	Yes	Perform Velocity Checking According to Terminal Type. • '00' = No. • '01' = Yes. As an example: '01' configures to perform Velocity Checking According to Terminal Type.	(1 Byte)

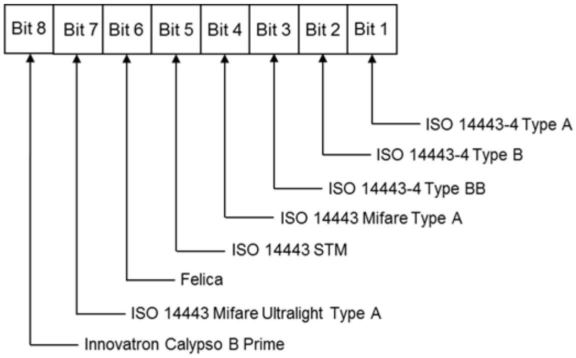
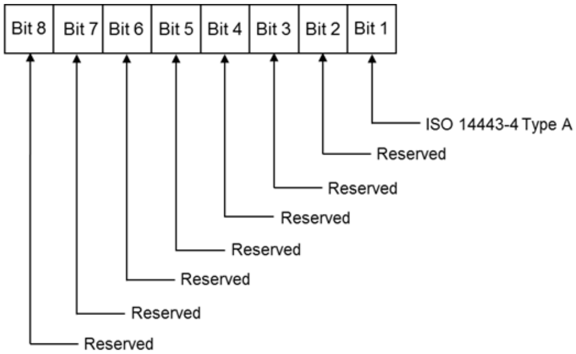
Tag	ICS Default	Description	Format
T9F8456	Yes	Support a Default Transaction Certificate Data Object List (TDOL). • '00' = No. • '01' = Yes. As an example: '01' configures to support a Default TDOL.	(1 Byte)
T9F841E	Yes	Always Perform Terminal Risk Management. • '00' = No. • '01' = Yes. As an example: '00' configures to not Always Perform Terminal Risk Management.	(1 Byte)
T9F845A	Yes	Skip TAC/IAC - Default Processing for Online Only Terminal. • '00' = No. • '01' = Yes. As an example: '01' configures to Skip TAC/IAC - Default Processing for Online Only Terminal.	(1 Byte)
T9F845B	Yes	Skip TAC/IAC - Default Processing for Offline Only Terminal. • '00' = No. • '01' = Yes. As an example: '01' configures to Skip TAC/IAC - Default Processing for Offline Only Terminal.	(1 Byte)
T9F8458	Yes	Perform Offline Data Authentication According to Terminal Type. • '00' = No. • '01' = Yes. As an example: '01' configures to perform Offline Data Authentication According to Terminal Type.	(1 Byte)

Tag	ICS Default	Description	Format
T9F8459	Yes	Select Account Type. • '00' = No. • '01' = Yes. As an example: '00' configures to not Select Account Type.	(1 Byte)
T9F845C	Yes	Detect CDA Failure Before Terminal Action Analysis. • '00' = No. • '01' = Yes. As an example: '01' configures to Detect CDA Failure Before Terminal Action Analysis.	(1 Byte)
T9F845D	Yes	Request CDA for Authorization Request Cryptogram (ARQC.) • '01' = No. • '00' = Yes. As an example: '00' configures to Request CDA for ARQC.	(1 Byte)
T9F845E	Yes	Request CDA for TC in Second GENERATE AC. • '01' = No. • '00' = Yes. As an example: '00' configures to Request CDA for TC in Second GENERATE AC.	(1 Byte)
T9F8431	No	List of Supported Languages. As an example: '65 6e 65 73' = "enes" where '65 6e' = "en" (English) and '65 73' = "es" (Spanish).	Hex ASCII (2 - 8 Bytes)
T9F8460	Yes	When Selecting to Bypass a PIN Method, all Other PIN Methods are Considered Bypassed. • '00' = No. • '01' = Yes. As an example: '01' configures for all Other PIN Methods are Considered Bypassed when Selecting to Bypass a PIN Method	(1 Byte)

8.9.5.6 ICS Tags in EMVCLESS.XML

ICS tags are grouped under tag TBF918803. Individual ICS configuration tags are numbered 1000, 1001, 1002, and so on. There is normally only one ICS configuration. The following table lists the parameters included for a given ICS configuration.

ICS Tags in EMVCLESS.XML

Tag	ICS Default	Description	Format
T9F928210	No	Generic Detection Type. This specifies all of the Level 1 card types which will be detected. **Byte 0**:  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 ISO 14443-4 Type A ISO 14443-4 Type B ISO 14443-4 Type BB ISO 14443 Mifare Type A ISO 14443 STM Felica ISO 14443 Mifare Ultralight Type A Innovatron Calypso B Prime **Byte 1**:  Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 ISO 14443-4 Type A Reserved Reserved Reserved Reserved Reserved Reserved Reserved As an example: '03 00 00 00' specifies ISO 14443-4 Types A and B.	(2 Bytes)
T9F928212	No	Generic Detection Global Timeout. As an example: '00 00 17 70' (hexadecimal) = '6000' (decimal) = 60 seconds.	EMV Format "b" (Binary) (4 Bytes)

Tag	ICS Default	Description	Format
T9F928214	No	Number of Cards Allowed to be Present at the Same Time. As an example: '01' indicates only one card to present at any time.	EMV Format "n 1" (1 Byte)
T9F918804	No	No Card Timeout. As an example: '00 00 5D C0' (hexadecimal) = '24000' (decimal) = 240 seconds = 4 minutes.	EMV Format "b" (Binary) (4 Bytes)
T9F1A	Yes	Country Code per ISO 3166. Country codes may be referenced at http://en.wikipedia.org/wiki/ISO_3166-1_numeric. As examples: • '01 24' = Canada • '08 40' = United States	EMV Format "n 3" (2 Bytes)
T5F2A	Yes	Currency Code per ISO 4217. Codes may be referenced at http://en.wikipedia.org/wiki/ISO_4217. As examples: • '01 24' = Canadian Dollar • '08 40' = U.S. Dollar	• EMV Format "n 3" (2 Bytes)
T5F36	Yes	Currency Exponent. This indicates the implied position of the decimal point from the right of the transaction amount represented according to ISO 4217. As an example: '02' indicates that there are two digits to the right of the decimal point.	EMV Format "n 1" (1 Byte)
T9F40	Yes	Additional Terminal Capabilities. This indicates the supported transaction types, data input, and data output capabilities of the terminal. Refer to EMV Book 4, "Additional Terminal Capabilities." As an example: '60 00 F0 B0 01' where • '60 00' (Transaction Type Capability) = Goods, Services. • 'F0' (Terminal Data Input Capability) = Numeric keys, Alphabetic and special character keys, Command keys and Function keys. • 'B0 01' (Terminal Data Output Capability) = Print/attendant, Display/attendant, Display/cardholder, and Code table 1.	EMV Format "b" (Binary) (5 Bytes)

Tag	ICS Default	Description	Format
T9F35	Yes	Terminal Type as defined by EMVCo. Refer to EMV Book 4, "Terminal Types." As an example: '22' indicates Attended, Offline with online capability, Operational control provided by merchant.	EMV Format "n 2" (1 Byte)
T9F01	Yes	Acquirer Identifier. Uniquely identifies the acquirer within each payment system. As an example: '01 23 45 67 89 01'	EMV Format "n 6-11" (6 Bytes)
T9F15	Yes	Merchant Category Code. This classifies the type of business being conducted by the merchant, represented in accordance with ISO 8583:1993 for Card Acceptor Business Code. As an example: '00 00'	EMV Format "n 4" (2 Bytes)
T9F16	Yes	Merchant Identifier. When concatenated with the Acquirer Identifier, this uniquely identifies a given merchant. As an example: '31 31 32 32 33 33 34 34 35 35 36 36 37 37 38'	EMV Format "ans 15" (15 Bytes)
T9F1C	No	Terminal Identification. This designates the unique location of a terminal at a merchant. As an example: '31 32 33 34 35 36 37 38'	EMV Format "an 8" (8 Bytes)

8.9.6 Configuring the EMV Application

For flexibility, the EMV Library separates the configuration for contact EMV and contactless payment into two different files. Parameters that can be applied to both contact and contactless transactions (e.g., CA public keys for an RID) are only required to be defined under contact EMV parameters. For such parameters, if they are not found under contactless payment parameters, the EMV Library will search for them under contact EMV parameters as well.

The following are the Tags in the major sections of the `EMVCONTACT.XML` and `CLESSCUST.XML` files:

- 1000 - AIDs
- 1100 - CAPK per RID
- 1200 - REVOK per RID
- 1300 - ICS (Terminal Data)

For each entry (e.g., AID) there are tag IDs of 10nn (where ‘nn’ is the AID entry).

A complete description of the XML files for EMV configuration is beyond the scope of this document. Ingenico works with each customer to determine appropriate parameter settings and create the configuration files. An overview of the information required from the customer and acquirer can be found in the EMV Configuration

Bulletin. Note that the specific parameter names and file formats described there are not the same as those used in the XML configuration files.

8.10 Quick Chip, M/Chip Fast, and Fast Quick Chip

Quick Chip and M/Chip Fast offer the ability to remove an inserted EMV card before the end of a transaction, making it more similar to flow of MSR transactions for cardholders. This functionality is present in Visa, MasterCard, Discover, and Amex cards. Ingenico also supports Fast Quick Chip transactions. See Fast Quick Chip for details on how to configure and use the Fast Quick Chip feature.

8.10.1 Requirements

Quick Chip and M/Chip Fast transactions **must** be authorized online with the floor limit set to zero:

- T9F1B (terminal floor limit) = 00000000
- T9F918709 (TAC-Default) Byte-4, Bit-8 must be set to 1 to ensure the transaction is handled online

The TAC-Default for most acquirers already has Byte-4, Bit-8 set for transactions, including Quick Chip and M/Chip Fast. The value can be set either:

- Temporarily by using a EMV 33.09.x Set Tag Data Message during a scheduled suspend step
- Permanently by using a EMV 33.08.x Set Variables Message or loading a DP with a custom `EMVCONTACT.XML` file

Suspend the flow at step U so the POS can control transaction results by specifying step U in either:

- EMV 33.01.x Status Message in standard flow
- EMV 33.00.x Transaction Initiation Message in on-demand

8.10.2 Differences

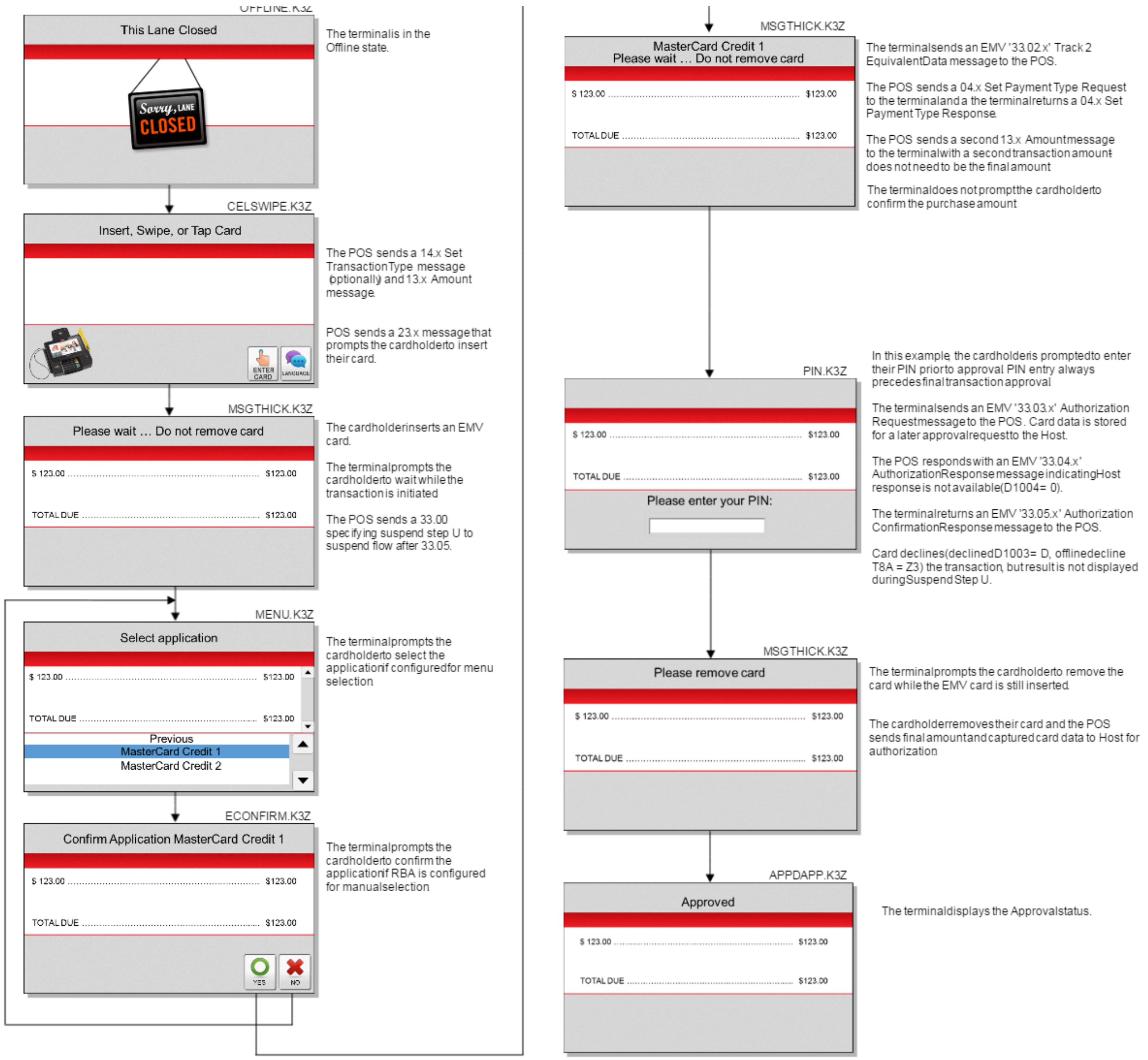
An on-demand or standard-flow EMV transaction using Quick Chip or M/Chip Fast continues typically until the terminal makes an online authorization request.

The differences that occur after the terminal sends EMV 33.03.x Authorization Request Message are as follows:

1. Send EMV 33.04.x Authorization Response Message with D1004 tag set equal to 0 to indicate a host response is not available.
2. The terminal prompts to remove the card, which can then be removed at any time. Its generated cryptogram is stored locally.
3. Since the card must be authorized online, the terminal sends EMV 33.05.x Authorization Confirmation Response Message to indicate that the transaction was declined (D1003 = D), and was declined **offline** (T8A = Z3).
4. The POS sends the saved data to the host for authorization with the correct transaction amount.
5. The POS displays the transaction result (whether it was approved or declined online) on the terminal screen.

8.10.3 On-Demand Quick Chip and M/Chip Fast Transactions

The following diagram illustrates the transaction flow when using Quick Chip or M/Chip Fast in an on-demand transaction.



The terminal is in the Offline state.

The POS sends a 14.x Set TransactionType message (optionally) and 13.x Amount message.

POS sends a 23.x message that prompts the cardholder to insert their card.

The cardholder inserts an EMV card.

The terminal prompts the cardholder to wait while the transaction is initiated

The POS sends a 33.00 specifying suspend step U to suspend flow after 33.05.

The terminal prompts the cardholder to select the application if configured for menu selection

The terminal prompts the cardholder to confirm the application if RBA is configured for manual selection

The terminal sends an EMV '33.02.x' Track 2 EquivalentData message to the POS.

The POS sends a 04.x Set Payment Type Request to the terminal and a the terminal returns a 04.x Set Payment Type Response.

The POS sends a second 13.x Amount message to the terminal with a second transaction amount; does not need to be the final amount

The terminal does not prompt the cardholder to confirm the purchase amount

In this example, the cardholder is prompted to enter their PIN prior to approval. PIN entry always precedes final transaction approval.

The terminal sends an EMV '33.03.x' Authorization Request message to the POS. Card data is stored for a later approval request to the Host.

The POS responds with an EMV '33.04.x' Authorization Response message indicating Host response is not available (D1004= 0).

The terminal returns an EMV '33.05.x' Authorization Confirmation Response message to the POS.

Card declines (declined D1003= D, offline decline T8A = Z3) the transaction, but result is not displayed during Suspend Step U.

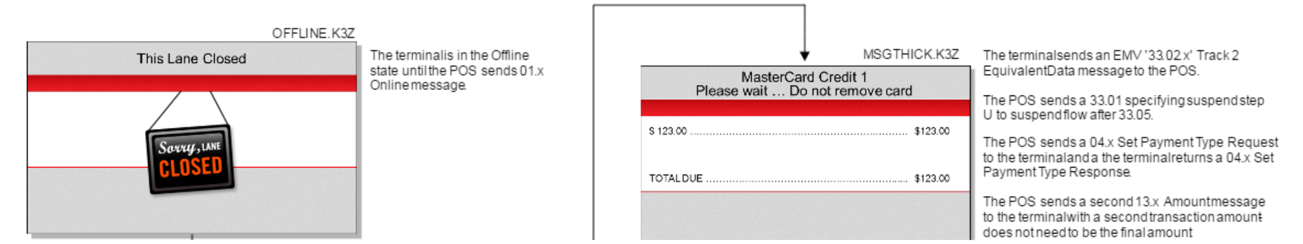
The terminal prompts the cardholder to remove the card while the EMV card is still inserted.

The cardholder removes their card and the POS sends final amount and captured card data to Host for authorization

The terminal displays the Approval status.

8.10.4 Standard Flow Quick Chip and M/Chip Fast Transactions

The following flow diagram shows the transaction flow when using Quick Chip or M/Chip Fast in standard flow.



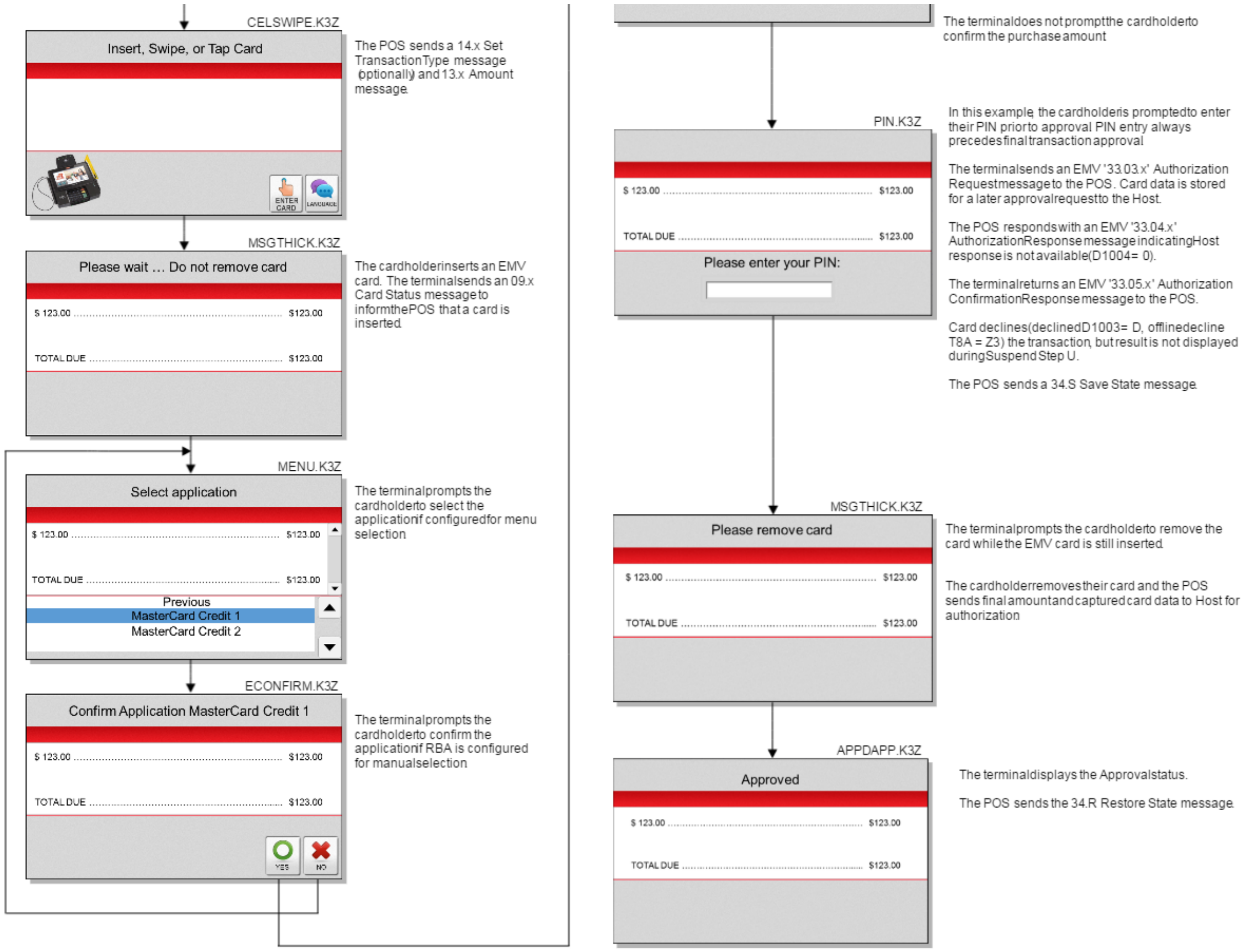
The terminal is in the Offline state until the POS sends 01.x Online message.

The terminal sends an EMV '33.02.x' Track 2 EquivalentData message to the POS.

The POS sends a 33.01 specifying suspend step U to suspend flow after 33.05.

The POS sends a 04.x Set Payment Type Request to the terminal and a the terminal returns a 04.x Set Payment Type Response.

The POS sends a second 13.x Amount message to the terminal with a second transaction amount; does not need to be the final amount



The POS sends a 14.x Set TransactionType message (optionally) and 13.x Amount message.

The cardholder inserts an EMV card. The terminal sends an 09.x Card Status message to inform the POS that a card is inserted.

The terminal prompts the cardholder to select the application if configured for menu selection

The terminal prompts the cardholder to confirm the application if RBA is configured for manual selection

The terminal does not prompt the cardholder to confirm the purchase amount

In this example, the cardholder is prompted to enter their PIN prior to approval. PIN entry always precedes final transaction approval.

The terminal sends an EMV '33.03.x' Authorization Request message to the POS. Card data is stored for a later approval request to the Host.

The POS responds with an EMV '33.04.x' Authorization Response message indicating Host response is not available (D1004= 0).

The terminal returns an EMV '33.05.x' Authorization Confirmation Response message to the POS.

Card declines (declined D1003= D, offline decline T8A = Z3) the transaction, but result is not displayed during Suspend Step U.

The POS sends a 34.S Save State message.

The terminal prompts the cardholder to remove the card while the EMV card is still inserted.

The cardholder removes their card and the POS sends final amount and captured card data to Host for authorization

The terminal displays the Approval status.

The POS sends the 34.R Restore State message.

8.10.5 Fast Quick Chip

The Fast Quick Chip function supports high-speed EMV transactions in a fast-food or other time-sensitive environment. The following two configurations must be set to enable Fast Quick-Chip:

- The variable, 0019_0021, in `emv.dat` must be set to 1
- The `emvaid.dat` flag per AID must be set to 1

8.10.5.1 Process Overview

There are three possible scenarios, depending on whether the Fast Quick Chip settings are enabled:

Scenario	Result
The variable, 0019_0021, in `emv.dat` is not set to 1	The terminal processes the transaction as a typical EMV transaction, regardless of the AID setting.

Scenario	Result
- The variable, 0019_0021, in `emv.dat` is set to 1 - The `emvaid.dat` flag for the AID is set to 1	The terminal processes the transaction as a Fast Quick Chip transaction.
- The variable, 0019_0021, in `emv.dat` is set to 1 - The `emvaid.dat` flag for the AID is set to 0	The terminal begins the transaction as a Fast Quick Chip transaction, but when it encounters the disabled AID, it performs an abbreviated Fast Quick Chip transaction.

Important: Because the AID is not known until after the transaction starts, the POS that supports Fast Quick Chip must be prepared to handle a transaction as either:

- Fast Quick Chip without the 33.03/33.04 sequence
- Abbreviated Fast Quick Chip with the 33.03/33.04 sequence

8.10.5.2 *Fast Quick Chip Process*

The Fast Quick Chip process takes place when both the terminal and the AID are enabled to support Fast Quick Chip.

The transaction can be initiated with the following messages:

- 23.x Card Read Request
- 41.x Card Read
- 87.x On-Guard and KME Card Read Data

Most standard EMV transaction messages are **bypassed and/or handled internally, including:**

- An on-demand EMV 33.00 Transaction Initiation Message
- A 04.x Set Payment Type Request Message
- A second 13.x Amount Message
- An EMV 33.03.x Authorization Request Message

8.10.5.2.1 Transaction Scenario

POS	Terminal
Sends a 14.x Set Transaction Type message. Optional.	**Note:** Fast QuickChip transaction does **NOT** fully/correctly complete for non-full EMV refund/void transactions.
Sends a 13.x Amount message.	
Sends a 23.x Card Read Request message.	
	The cardholder inserts the card.

POS	Terminal
	Sends a 23.0Q card source response message (or 41.Q or 87.0Q)
	The cardholder enters a PIN (if prompted).
	Displays a *Please Remove Card* message.
	Sends an EMV 33.05.x Authorization Confirmation Response Message with T9F27 = first cryptogram = ARQC.
	The cardholder removes the card.
	Sends a 09. R Card Removed Status Message.

8.10.5.2.2 Fast Quick Chip Cash Back Process

To provide cash back for a Fast Quick Chip (FQC) transaction, RBA can dynamically suspend and enable on-demand cash back when the following conditions are met:

Fast Quick Chip Cash Back Requirements

Requirement	Description
1	Configure the FastQuickChipSuspendStep variable, `0019_0022`, in EMV Flags (emv.dat) OR Send a FQC cashback suspend request with a 33.01 message before the transaction start
2	The card's AID Application Usage Control (AUC) must permit cash back OR Forced cash back is enabled for the AID in EMV AID Parameters (emvaid.dat)
3	The card's AID must have at least one CVM enabled for cash back. The CVMs must also be enabled in the terminal capabilities. **Note:** If the final CVM selected by the card is not enabled for cash back, the FQC transaction is canceled if a non-zero cashback amount is selected.

Example Process per AID Type

If the POS requests a FQC cashback suspend step, one of the following example processes is a likely scenario:

AID Type	Condition	Result
Domestic Debit AID	• The terminal is configured to enable some AID CVMs that allow cash back • The AID CVMs are enabled in the terminal capabilities • The card's AID Application Usage Control (AUC) permits cash back for domestic debit transactions	The FQC transaction suspends for cash back
Domestic Debit AID	• The terminal is configured to enable some AID CVMs that allow cash back • The AID CVMs are disabled in the terminal capabilities	The FQC transaction does not suspend for cash back
International Debit AID	• The terminal is configured to enable some AID CVMs that allow cash back • The AID CVMs are enabled in the terminal capabilities • The card's AID Application Usage Control (AUC) permits cash back for international debit transactions • The POS does not support cash back for international debit transactions	The FQC transaction does not suspend for cash back
Credit AID	The terminal is configured to disable all AID CVMs that allow cash back	The FQC transaction does not suspend for cash back

8.10.5.3 Abbreviated Fast Quick Chip Process

The abbreviated Fast Quick Chip process takes place when the terminal is enabled to support Fast Quick Chip, but the AID is not.

The transaction can be initiated with the following messages:

- 23.x Card Read Request
- 41.x Card Read
- 87.x On-Guard and KME Card Read Data

Most standard EMV transaction messages are **bypassed and/or handled internally**:

- Including 04.x Set Payment Type Request messages, a second 13.x Amount Message, or an on-demand EMV 33.00 Transaction Initiation Message
- Excluding EMV 33.03 or 33.04 online authorization messages, which are sent to accommodate the AID preference not to support Fast Quick Chip functionality
    - **Note:** The POS can perform a legacy Fast Quick Chip transaction by sending a 33.04 Host Unavailable message and ignoring the AID preference.

8.10.5.3.1 Transaction Scenario

POS	Terminal
Sends a 14.x Set Transaction Type message. Optional	
Sends a 13.x (dummy) Amount message.	
Sends a 23.x Card Read Request message.	
	The cardholder inserts the card.
	Sends a 23.0Q card source response message.
	Sends an EMV 33.02.x Track 2 Equivalent Data Message, which indicates Fast Quick Chip transaction reverted to a typical EMV transaction because the AID does not support Fast Quick Chip.
Sends 04.x Set Payment Type Request message.	
	Sends a 04.x Set Payment Type Response message.
Sends a 13.x (actual) Amount message.	
	The cardholder confirms the purchase and enters a PIN. Optional.
	Sends an online EMV 33.03.x Authorization Request Message with the first cryptogram that reflects the actual transaction amount. sent in the second 13.x message.
Sends a EMV 33.04.x Authorization Response Message.	
	Sends an EMV 33.05.x Authorization Confirmation Response Message with the second cryptogram.
	Displays a Please Remove Card message.
The cardholder removes the card.	
	Sends a 09.R Card Removed Status Message.

8.11 Support for Voice Referral for EMV

8.11.1 Overview

Voice referral, also referred to as voice authorization, is a situation where voice authorization by phone is required in order to complete the EMV transaction. The merchant is given a phone number to call for transaction approval. A "declined" message will be sent but the terminal will instead display "Voice authorization required" and prompt for card removal. The operator is then given the card to perform the voice referral and complete the transaction. If the transaction is approved, an authorization code will be provided over the phone. This code will be included in the receipt and provided in the settlement.

8.11.2 Functional Description

When the POS sends an EMV '33.04.x' Authorization Response Message with a tag `T8A` value of '01' (referral requested by issuer) or '02' (referral result), the RBA will display "Voice authorization required" and prompt the cardholder to remove the card. EMV tag `D1011` (Transaction Result) has been added to support this feature. The use of this tag will be similar to that of tag `T8A`; '00' for approval and '05' for decline. Tag `D1011` is optional; when not included in the Authorization Response message, the original processing will be used.

A new prompt has been added to the PROMPT.XML file in three languages:

- <Prompt id="284" message="Voice authorization required"/> (English)
- <Prompt id="284" message="Requiere autorización de voz"/> (Spanish)
- <Prompt id="284" message="Autorisation vocale requise"/> (French)

When this prompt is displayed, it will be followed by a second prompt instructing the cardholder to remove their card. The transaction will be completed once the operator calls the provided phone number and receives the voice authorization. There are three important steps in this process:

1. When the terminal receives a voice authorization request from the card issuer, it must terminate the EMV transaction by asking the card for an AAC.
2. Once the voice authorization is processed, acquirers must retain the authorization approval code and ARQC produced by the card.
3. The authorization approval code must be included in the clearing message.

8.11.3 Transaction Example

The following transaction example is based on a tag `T8A` value of '02' indicating that voice authorization is required. Refer to the following table or a list of the transaction steps.

Voice Authorization Required Transaction Example

Sequence	Message	ECR	Terminal
The POS sends an offline message to the terminal.	00.x Offline Message	→	

Sequence	Message	ECR	Terminal
The terminal acknowledges with an offline message and displays the Offline/Lane Closed form.	00.x Offline Message		←
The POS sends a configuration write message to enable EMV transactions.	'60.19[GS]1[GS]1'		→
The terminal acknowledges the configuration write and enables EMV.	'60.2'		←
The POS sends an online message to the terminal.	01.x Online Message		→
The terminal acknowledges the online message.	01.x Online Message		←
The terminal prompts the cardholder to swipe or insert their card.			
The cardholder inserts their EMV card.			
The terminal sends card status message to POS indicating that EMV card has been inserted.	'09.020201I'		←
The terminal briefly displays "Please wait ... Do not remove card" message.			
The terminal prompts the cardholder to select the language (optional, depending on card's application).			
The cardholder selects the language.			
The terminal displays the application (e.g., VISA Debit, MasterCard Credit).			
The POS sends an amount message to terminal.	'13.2000'	→	

Sequence	Message	ECR	Terminal
The terminal returns a Track 2 Equivalent Data message.	EMV '33.02.x' Track 2 Equivalent Data Message	←	
The POS sends Set Payment Type message.	'04.0B2000'	→	
The terminal acknowledges the Set Payment Type message.	'04.0B000'	←	
The POS sends a final amount message.	'13.2500'	→	
The terminal displays the Amount Verification screen and prompts the cardholder to proceed or cancel.			
The cardholder confirms the purchase amount and is prompted to enter their PIN.			
The terminal sends the Authorization Request message to the POS.	'33.03.0000[FS] .... '	←	
The POS returns an Authorization Response message. Tag 8A has a value of '02' indicating that voice authorization is required.	'33.04.0000[FS]T8A:02:a01[FS]'	→	
The terminal sends an Authorization Confirmation Response message.	'33.05.0000[FS] ....'	←	
The terminal displays "**Voice authorization required**" " **Please remove card** " The terminal beeps every 3 seconds until card is removed.			
The cardholder removes their card and hands it to the operator for voice authorization.			

Sequence	Message	ECR	Terminal
The terminal sends the updated Card Status message to POS indicating that the card has been removed.	'09.020201R'	←	
The terminal displays the card swipe/insert prompt.			
The POS sends a hard reset message to the terminal.	'10.'	→	

9 Additional Features

9.1 Amount Function

Purchase Amount + Cash Back Amount = Total Amount

The purchase amount may also be referred to as the amount or total. The purchase amount may be received in the following messages:

- 13.x message, (single or multiple amount field)
- 04.x message, forced payment with amount (single amount field)
- 28.x message, set amount (single amount field)

When a 13.x message with multiple amount fields is used, the host variable `ID_TOTAL` is not available until the payment is selected. The Amount Index can be configured in cards.dat. See Amount Index in Card Configuration (cards.dat).

When the amount value is received in 13.x, 04.x, or 28.x message (all of which use a single amount field), the host variable `ID_TOTAL` is immediately updated.

When the cash back value is selected by the cardholder, its value is added to the purchase amount.

9.1.1 Configuring Amount Function

The RBA local configuration provides the ability to verify the total amount. When verification is enabled, the cardholder can be prompted to verify if the total amount is correct (the total amount includes the purchase amount and cash back amount, if applicable). The selection of the purchase verification function is controlled by a configuration option listed in cards.dat file, individually per card type, called Verify Amount (see Card Configuration (cards.dat) for more information). When prompted to verify the purchase amount, the cardholder can press the YES, NO, or CANCEL button.

- If the cardholder presses YES, the RBA flow goes to the next function.
- If the cardholder presses NO, the RBA does the following:
    - sends out a 10.x reset message if the purchase amount is present
    - checks the configuration option listed in mainFlow.dat file, index '0007_0006' (On Total Amount Incorrect, Return To...), and goes to that state.
- If the cardholder presses CANCEL, the RBA sends out a 10.x reset message, and goes back to the transaction start.

> *See Amount Verification for more information on the Amount Verification process or 13.x Amount Message for more information on amount index setup.*

9.2 Signature Retrieval

The signature process is controlled by the RBA configuration switches listed in the Signature section in `config.dfs`.

Only the stylus attached to the terminal may be used to write the signature on the terminal screen. The signature process starts with the first touch of the screen with the stylus.

When writing of the signature is finished, it can be accepted when the OK button is pressed, or when the pen is not used for a specified time. In the latter case, the terminal automatically accepts the signature after a time specified in `config.dfs`, index '0009_0013'. When accepted, the terminal displays the text “Signature Accepted” below the signature box.

In order to clear the signature from the screen, the customer can press the CLEAR button, or the cashier can send the '15.4' reset signature message. The CLEAR button can be used many times.

The screen signature must be translated from analog form to digital. The average signature is translated into 700 bytes of digital data. When the POS retrieves that data from the terminal, the signature’s digital data is divided into signature blocks. The configuration switch in `sig.dat` file, index '0009_0012', controls data length per block. The default value is set to 200 bytes (this is also the maximum number).

The number of bytes per signature block can be defined in the local configuration file `sig.dat`, but cannot be changed or read by 28.x or 29.x.

9.2.1 Finger Signature

An option in the signature object line of a K3Z file allows finger signature in addition to the stylus. This feature is available on all iSC terminals using Form Builder version 12.1.0.0002 or higher.

9.2.1.1 Enabling Finger Signature

To enable finger signature, the K3Z file for the signature form needs to be updated. Add the following to the signature element:

```
allowfinger='true'
```

9.2.2 Configuring Signature Retrieval

9.2.2.1 Standard

Signature capture is a standard feature for the iSC250, iSC350, and iSC480 terminals. When using RBA defaults, functions are executed in a specific order. The signature function is part of it. When signature capture is finished, RBA goes to the next function. The signature position in the RBA data flow has one adjustment in file config.dfs, 0009_0006 index, called Save State on Signature Capture.

- When Save State on Signature Capture = 0, terminate the transaction before prompting for a signature.
- When Save State on Signature Capture = 1, save the current state, prompt for a signature, and then return to the saved state.

The signature screen can be aborted by messages 00.x, 01.x, 10.x, 15.0, 15.1, 15.6, and 30.x.

9.2.2.2 On-Demand

The signature may be started by the RBA standard process or by the POS 20.x message. When the signature is accepted, the screen shows the signature until the start of the digital version of the signature is uploaded to the host, or it is immediately cleared from the screen. That is controlled by the configuration parameter in sig.dat file, index 0009_0008, Display Signature Until Download Starts (0 = disable, 1 = enable). The 29.xxxxx7yy message is used for uploading the signature to the host. The next process after signature is the Transaction End process.

If the signature is accepted, RBA can send to the POS an unsolicited the 20.x message to inform it that the signature data is ready to be retrieved. Sending the 20.x message is controlled by the configuration switch in `sig.dat` file, index 0009_0002. When this switch is set to 1, it sends the 20.x message, 0 = no message.

The signature on-demand starts when the terminal receives the 20.x message from the POS. Before the message is executed, RBA checks the following conditions.

- If the terminal is not in the signature process, the current process is terminated, and the new signature request is executed. After signature is finished, RBA goes to the Transaction End.
- If the terminal is in the signature state, invoked by RBA or by a previous on-demand 20.x message, the current signature is terminated, and the new message is executed.
    - If the 20.x prompt field is greater than 0, the prompt text is displayed on the signature screen below the signature box.
    - If the 20.x prompt field equals 0, RBA config prompt is used instead. When the signature is accepted, RBA goes to the Transaction End.
- When another on-demand message is received during the execution of 20.x (e.g., 21.x or 23.x), that message is not executed, and a reject response is sent, if available. The current process is continued.
- When signature on-demand is successfully finished, RBA displays “Signature accepted” for three seconds and then goes to Transaction End. After that, based on the RBA configuration selection, it might wait for the host-reset message, go to advertisements, or start a new transaction.

The signature process can be aborted by messages 00.x, 01.x, 10.x, 15.0, 15.1, 15.5, 15.6, and 30.x.

9.2.2.3 EMV and MSD

For PayPass, PayWave, ExpressPay, and D-PAS MSD or EMV cards, consult the following table for determining signature limit:

Configuring Paywave and Discover MSD Signature Limit

Use CVM Limit from EMV .xml Files	Use cards.dat for Signature Threshold
• All EMV cards • PayPass 3 MSD • ExpressPay 3 MSD	• Discover D-PAS MSD • PayWave MSD • PayPass 2 MSD • ExpressPay 2 MSD

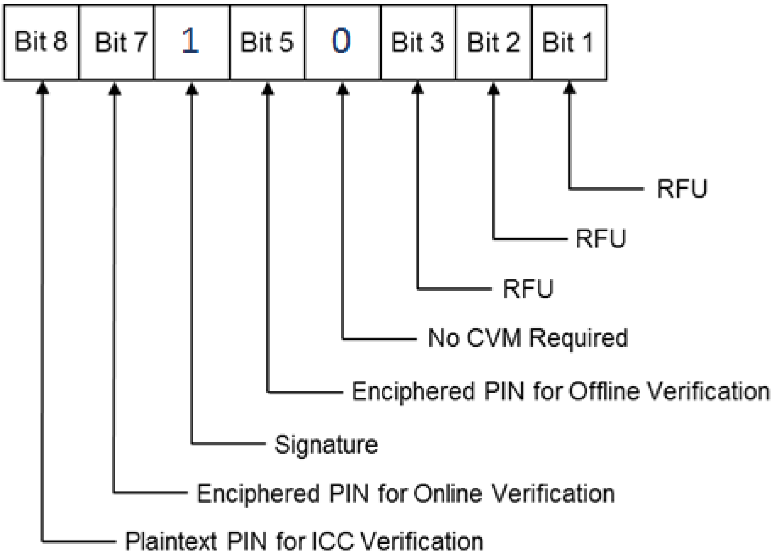
9.2.3 Disabling Electronic Signature on iSC Series Terminals

9.2.3.1 Overview

Some Ingenico merchants prefer a paper signature over the standard electronic signature following transaction approval. A new configuration has been added which enables electronic signature to be bypassed and a paper signature to be recorded in its place. This applies to both standard and on-demand flow. Other customers will not be affected by this change and can continue using electronic signature.

9.2.3.2 Configuring for Paper Signature

To collect a paper signature only, electronic signature must be disabled. Because a signature is still being recorded for the approved transaction, CVM must be enabled and Signature must be specified. Accordingly, the "No CVM Required" flag must be set to '0' and the "Signature" flag must be set to '1' in tag T9F33 as illustrated in the below figure. Refer to the EMV '33.07.x' Terminal Capabilities Message for information on setting the card verification method using the terminal capabilities message.


Bit 8
Bit 7
1
Bit 5
0
Bit 3
Bit 2
Bit 1
RFU
RFU
RFU
No CVM Required
Enciphered PIN for Offline Verification
Signature
Enciphered PIN for Online Verification
Plaintext PIN for ICC Verification

Setting EMV Tag T9F33 Byte 2 Contents for Signature Required

With CVM enabled, the Signature Required tag D1002 must be set to '0' (no electronic signature required) and included in the EMV '33.03.x' Authorization Request and EMV '33.05.x' Authorization Confirmation Response messages sent to the POS. With electronic signature bypassed, the POS should then request cardholder signature on the paper receipt. Anytime signature is specified in the CVM, it is up to the POS to read the setting of tag D1002 and determine if electronic signature was captured or if a paper signature is required.

9.2.4 Retrieval Using Get Variable

The 29.x Get Variable message is the standard method of signature retrieval from the iSC250/iSC350/iSC480. The POS can send the 29.x request at any time, but signature data is only available after one of the following conditions:

- The customer has pressed "OK" on the signature capture page to indicate signature termination.
- The terminal times out after the Pen Up time has expired.

For the iSC350, this state is known only when reported by an 11.x Status Response sent from the POS. Once a signature available response is sent from the iSC250/iSC350/iSC480 terminal to the POS, the POS requests the number of 200-byte blocks the signature data fills. If this number is non-zero, the POS will begin requesting the signature blocks (one at a time), up to the number of blocks needed to rebuild the signature.

Signature Request and Response Using the Get Variable Message

Sequence	Message	POS	Terminal
Begin Authorization Sequence.	50.x Authorization Request.	←	
End Authorization Sequence.	50.x Authorization Response.	→	
Begin Signature Sequence.	11.x Status Request.	→	

Sequence	Message	POS	Terminal
Approved.			
	11.10 Status Response	←	
Terminal displays signature form.			
Customer inputs signature and presses OK.			
	11.x Status Request	→	
End Signature Sequence.	11.11 Status Response (Signature Available)	←	
Begin Signature Transmittal Sequence. Get Variable message determines the number of blocks the signature data covers.	29.10000712	→	
x = the number of signature blocks that contain data, up to 10. The POS is expected to request x number of signature blocks in order to recreate the entire signature. There is no value when no data is available. For example, x = '4' if there are four signature blocks, and those signature blocks are defined as RBA variables `700` through `703`. Thus, the last signature block is `70`(x-1).	27.20000712x	←	
Request first signature block, `700`.	29.10000700	→	
Return first signature block, `700`.	29.20000700{data}	←	
Request second signature block, `701`.	29.10000701	→	

Sequence	Message	POS	Terminal
Return second signature block, 701.	29.20000701{data}	←	
29.x messages continue until the final message in the signature block...	...		
Request last signature block, 70(x-1).	29.1000070(x-1)	→	
Return last signature block, 70(x-1).	29.2000070(x-1){data}	←	
End Signature Transmittal Sequence.	Reset message	→	
Shutdown Sequence.	Offline		

> Omitted in the diagram above, each message flowing either direction prompts an ACK response from the receiving unit.

9.2.5 Signature Format

The iSC250, iSC350, and iSC480 terminals support the SIG_BIN_2 (3-byte ASCII) Signature Format.

See 3-Byte ASCII Signature Format for more information.

> RBA also supports the Hypercom Legacy signature format, but the signature block size is limited to 640 x 128 pixels.

9.3 Card Swipe Function Details

The RBA can process standard Visa card swipes and non-standard card swipes. Also, the RBA is capable of screening out account numbers that do not meet a few basic criteria. This capability, called BIN range checking, checks the length of the account number. An account number with incorrect length is not accepted, and the terminal displays the “Your card is invalid” prompt and returns to the wait for a new card swipe screen. BIN range checking is performed after the RBA collects the account number and after the payment selection.

9.3.1 Standard ISO Cards

When the cardholder is prompted to swipe a card, the magnetic stripe reader or contactless card reader may be successful in reading the data or it may have errors.

- If the card swipe is error-free, the RBA saves the card information and progresses to the next process in the current flow.
- If the card read is unssuccessful:
    - The terminal displays an error prompt for three seconds, for example, “Card read error. Try again”.
    - The terminal returns to the initial card swipe screen.
    - If a consecutive number of bad card swipes reaches the limit specified by parameter '0003_0001' in the Terminal Local MSR Card Swipe Options (msr.dat) section in the RBA configuration file `config.dfs`, the prompt changes to “Please hand card to cashier,” displayed for three seconds.
    - After that, the RBA checks parameter '0003_0002' in the same section of `config.dfs`, which tells the duration (in tenths of a second) that the“Ask for Assistance” prompt should display. If the value is 0, the prompt is not displayed.
    - Next, RBA initializes the bad card swipe counter to the starting value and goes to the initial card swipe screen. If the new card swipe is faulty and prompts from the previous bad card swipe have not yet expired, the new card swipe is ignored. This process filters out card swipes that are too quick, too slow, or too shaky.

9.3.2 Non-Standard Cards

Non-standard cards are accepted in both the on-demand card swipe request and the RBA standard process flow.

The differences between standard cards and non-standard cards are that:

- Standard cards have the expiration date in the MSR Track 2, and the account number must be followed by the equal sign (=), which is followed by the four-digit expiration date. The format of the date is *YYMM*, where *YY* is the two-digit year and *MM* is the two-digit month.
- Non-standard cards do not have the expiration date in the MSR Track 2, and they do not have an equal sign (=) after the account number.

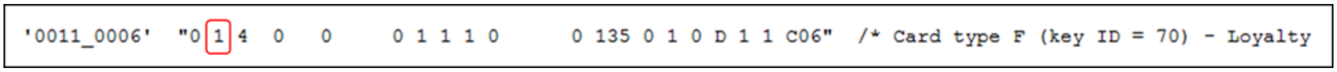
Account data may be received from the 12.x Account Message, MSR card swipe, or contactless reader card tap. The rules for these sources are as follows:

- If the account number is received in the 12.x message, the account is always treated as the payment account. Data from this message is loaded into buffers used to store track data from the local card swipe. Only Track 2 buffers are used. Track 1 remains empty.
    - If the message contains the account number only, with no equal sign (=) or expiration date (non-standard card), the Track 2 expiration date is set to 0000.
    - If the message contains the equal sign (=) followed by the expiration date (standard card), the date is loaded into the Track 2 buffer.
- If the account number is received from the local MSR or contactless reader card swipe, both tracks are checked for an expiration date. If none is found, the date is set to NONE.
- If the expiration date is NONE, and the card is a payment type, such as debit or EBT, the terminal displays the payment selection followed by the Invalid Card prompt and returns to the transaction start.

Sending Card Data in the Clear
You can set the number of card digits to be sent in the clear from the start or end of the card number using parameters set in the Security Parameters.

9.3.3 Non-Payment Cards

A non-payment card refers to a loyalty card, rewards card, points card, advantage card, or club card type. If the expiration date is NONE and the card is a non-payment type, the RBA sends out the 18.x Non-Payment Card Message to the host and returns to the transaction start. The following figure shows an example non-payment card selected as described in the Cards section of the `config.dfs` file.



```
'0011_0006'  "0 1 4  0   0     0 1 1 1 0      0 135 0 1 0 D 1 1 C06"  /* Card type F (key ID = 70) - Loyalty
```

Non-Payment Cards in cards.dat

Note the second parameter circled in the above figure. This is the Card Type parameter which indicates the following:

- 0 = Payment card.
- 1 = Non-Payment card.

9.3.4 On-Demand Requirements

The 23.x Card Read Request (On-Demand) from the POS starts the Card Swipe process. Prompt text displayed on the screen is from the 23.x message. During the execution of this process, only a single card swipe is allowed.

- If the card swipe is good, the terminal displays the "Card Accepted" prompt for three seconds, sends out 23.x response with the card data, and returns to the interrupted process.
- If the card swipe is bad, the terminal displays the "Card Read Cancelled" prompt for three seconds, sends out response with no data status, and returns to the interrupted process.

If the 23.x message does not include prompt text, it will not be executed, and an error status response is returned to the host.

If RBA is executing a 23.x message, any new 23.x message will not be executed, and response with error status is returned to the host.

CANCEL button terminates the process.

Additional information about this message is also available in the On-Demand Transaction Process section.

9.4 Advertising Support

RBA supports server-based advertising on Telium terminals. Please contact your Ingenico Representative for additional information.

9.5 Scrolling (Digital) Receipt

9.5.1 Clearing Line Items

The digital receipt is also called a scroll item or line item.

The upper part of the terminal screen is used for displaying a digital version of the cardholder paper receipt. Digital Receipt is available on all screens which have the upper part of the screen reserved for it.

The Digital Receipt can show the last four purchased items. Terminals with a larger screen can display more items. The number of displayed lines is activated by the size of the scrolling receipt element in the form file.

Text for the digital receipt is received via the 28.x Set Variable Request message. The receipt text is saved in the terminal's buffers and used until the end of the transaction. At the Transaction End, all buffers are cleared.

Clearing line items can be achieved in one of the following ways:

- Clear a single line item via the 28.x message with the text part set to spaces (0x20).
- Clear all digital receipt lines using only the '15.8' soft reset message from POS.
- Clear the whole transaction message.

9.5.2 Compression

The Line Items are displayed on the terminal screen in a dedicated area. The off-the-shelf RBA takes advantage of the whole screen width. The Scrolling Receipt element size can be tailored to fit the background image size, if specified in the form file. The function compresses the line item text to the Scrolling Receipt element size.

Text compression works as follows:

- If the Line Item text received from the host in a message is greater than 40 characters, it is truncated to 40 characters; otherwise it is saved in the RBA internal buffer.
- When it is time to display the text, it is compressed to fit the scroll window width. The width is selected in the form files.

The form element, Scrolling Receipt, defines the height and width of the scroll window size (SWS), and where the line item text is displayed. Here are the compression rules:

- If the text is longer than SWS and has no '$' char, the text is truncated at the window's width.
- If the text is longer than SWS and has a '$' char, part of the text in front of '$' character is removed and '$' together with following chars are shifted forward, example:

  TestLineLengthBabcdefghijkABCDE$FGHIJKLM" - line before compression

  TestLineLengthBabcdef$FGHIJKLM" - line after compression
- If the text is shorter than the SWS width, it is displayed as is.

The compression does not provide '$' character alignment.

9.6 Contactless Key Card Support

9.6.1 Introduction to Contactless Key Card Support

Low-level contactless key card support for Telium RBA is provided by a common Telium application layer called GRAF through newly added Application Interfaces (APIs). The application uses these APIs to service Contactless Key Card communications with the POS. The GRAF APIs are transparent to the POS, which processes RBA messages only.

The following card types are supported:

- MIFARE Classic 1K
- MIFARE Classic 4K
- MIFARE Mini
- MIFARE Ultralight

Refer to the RBA Low-Level Contactless Key Card Support section for more information on Contactless Key Card Support.

9.6.2 RBA Low-Level Contactless Key Card Support

RBA manages the interface between the POS and the low-level contactless key card support in GRAF, which is transparent to the POS. Refer to the following sections for more information:

- Contactless Key Card General Flow
- Setting Contactless Mode (60.x/28.x)
- Enabling Contactless and Requesting Card Tap (01.x/23.x)
- 16.x Contactless Mode Request
- 17.x: Merchant Data Write
- 17.x Merchant Data Write Message Usage Examples
- 28.x: Set Variable Request
- 36.x Notification of Command Execution

The 17.x section includes examples for executing commands, as well as executing commands as a batch. The 28.x page provides information on changing contactless mode.

9.6.2.1 Contactless Key Card General Flow

This section describes the general flow for the Contactless Key Card interface. Contactless mode must first be set to Contactless Key Card mode. Form there, contactless must be enabled and a card tap must be requested through the 23.x Card Read Request message. Once one or more cards have been detected, the terminal will send an unsolicited 16.x message to the POS with identifying card information. The POS will send one or more commands to the terminal. When the POS has completed sending all other commands, it sends a Complete Tap command through the 17.x Merchant Data Write message to indicate to the terminal that the tap is complete. Once the tap is complete, the contactless field must be disabled. The terminal is then ready to initiate another tap sequence or other action such as payment request.

The following table describes the Contactless Key Card General Message Flow. If another tap sequence is started, the first step in the table may be skipped as the contactless mode will still be set to Contactless Key Card mode.

Contactless Key Card General Message Flow

Sequence	Message	POS	Terminal
Set the terminal's contactless mode to Contactless Key Card mode.	60.x Configuration Write or 28.x Set Variable Request	→	
Enable contactless and request a card tap.	23.x Card Read Request (On-Demand)	→	
Once one or more cards have been detected, the terminal will send an unsolicited contactless mode request message to the POS with identifying card information.	16.x Contactless Mode Request	←	
The POS acknowledges and sends one or more commands to the terminal.		→	
The POS sends a Complete Tap command last to indicate to the terminal that the tap is complete.	17.x Merchant Data Write	→	
The POS sends a command to disable the contactless field.	60.x Configuration Write or 28.x Set Variable Request	→	
The terminal is now ready to start another tap sequence or other action such as payment request.			

9.6.2.2 *Setting Contactless Mode (60.x/28.x)*

In order to service Contactless Key Card taps, the terminal must be in Contactless Key Card mode (8). There are two methods for switching between contactless modes. The method depends on whether the mode change needs to be permanent or short term.

1. For permanently setting the contactless mode, use the 60.x Configuration Write message to change parameter '0008_0001' (Enable Contactless Reader) to a value of '8'.
2. For short term setting of the contactless mode (which does not persist following a reboot), use the 28.x Set Variable Request message with variable `412` (Contactless Mode).

9.6.2.3 *Enabling Contactless and Requesting Card Tap (01.x/23.x)*

Once the terminal is in Contactless Key Card mode (8), the contactless must be enabled and a card tap must be requested. There are two methods to implement this in the RBA:

1. The POS sends a 01.x Online Message to display the "swipe" screen, which enables contactless and requests the card tap.
2. The POS sends a 23.x Card Read Request (On-Demand) message to enable contactless and request the card tap. It is recommended that a 23.x message not be sent from the "swipe" screen in order to avoid accidentally starting a second tap sequence from the "swipe" once the 23.x message is complete.

10 Appendices

10.1 Appendix A. Three-Byte ASCII Signature Format

This specification describes the format of signature capture data as three-byte ASCII output from the application. This output method is used to minimize the length of the captured data image while using only printable ASCII characters.

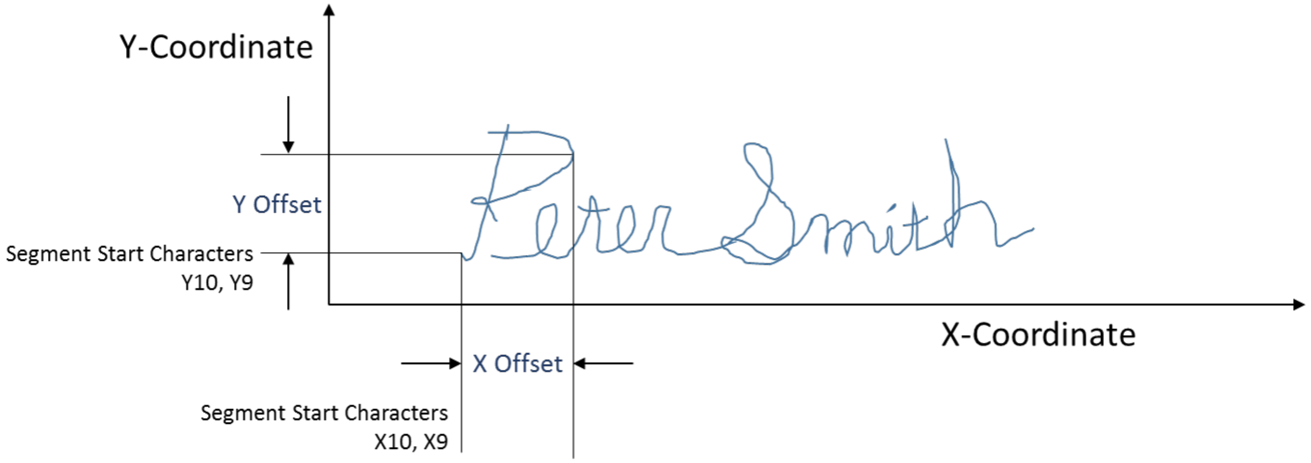
10.1.1 A.1. Specifications

Coordinate data is organized into segments of pen-down data. Each segment is started by a special control code that identifies the local region of signature space to begin the segment. This is called the **Segment Start character**. The Segment Start character is followed by a variable number of coordinate data sets that describe the pen movement throughout the pen-down segment.

Each coordinate data set consists of three characters describing the position of the pen relative to its position described by the immediately preceding coordinate data set. The segment is concluded if a pen-up condition occurs or if two successive coordinate points are separated by a distance that exceeds the range capabilities of the three-character coordinate data set format. A pen-up condition is marked by a special Pen-Up character. Coordinate data is not scaled in this format. Only ASCII characters in the range of 20 hex to 7E hex are used.

10.1.2 A.2. Coordinate Data Reconstruction

The signature box is configured with an X-Y coordinate system. Every point in the box has a unique X and Y coordinate. Each coordinate consists of 11 bytes for the X position and 11 bytes for the Y position. The starting point of the signature is defined as the Segment Start character. All other points are relative to this start character and the coordinates are represented in offset with respect to this point, as shown in the image below.



Signature Offset Example

The X and Y offsets consist of nine-bit values and can be positive or negative. The offset values are coded in pairs that complement notation with the sign bit in the most significant position. Sign bit extension to the 10th and 11th bits must be performed when adding the nine-bit offset values to the previous 11-bit coordinate values.

Computing Successive Coordinates with Data Sets

Data Set	Data Length	Set Contains	Notes
Segment Start	11 bytes (11-bit X position, 11-bit Y position)	X10, X9, Y10, Y9, X8, X7, X6, X5, X4, X3, X2, X1, X0, Y8, Y7, Y6, Y5, Y4, Y3, Y2, Y1, Y0	
Succeeding coordinate offset	Nine bytes (nine-bit X position, nine-bit Y position)	x8, x7, x6, x5, x4, x3, x2, x1, x0, y8, y7, y6, y5, y4, y3, y2, y1, y0	These values are added to the previous set of data to form a new 11- byte data set.
Location of new coordinate	11 bytes (11-bit X position, 11-bit Y position)	X10', X9', X8', X7', X6', X5', X4', X3', X2', X1', X0', Y10', Y9', Y8', Y7', Y6', Y5', Y4', Y3', Y2', Y1', Y0'	Sucessive coordinate offsets are added to this data set until the signature is completed or it exceeds the signature size limit. x8 is added to X10, X9, and X8. y8 is added to Y10, Y9, and Y8. The rest correspond to their respective numbers only (For example: X5 + x5 = X5').

10.1.3 A.3. Format for Signature Data

The signature data transmitted from the application uses the following format:

```
SEGMENT START CHARACTER
|
COORDINATE CHARACTER 1\
COORDINATE CHARACTER 2 |  SET #1
COORDINATE CHARACTER 3/
|
COORDINATE CHARACTER 1\
COORDINATE CHARACTER 2|  SET #2
COORDINATE CHARACTER 3/
.
.
.
COORDINATE CHARACTER 1\
COORDINATE CHARACTER 2|  SET #N
COORDINATE CHARACTER 3/
```

Each coordinate data set is followed by one of the following characters or data set:

- A Pen-Up control character
- A new Segment Start control character
- Another coordinate data set

10.1.4 A.4. Pen-Up Control Character

A Pen-Up control character is always followed by a Segment Start control character unless there are no more segments in the signature.

- **Binary Format**: 0 1 1 1 0 0 0 0
- Range (hex): 70

10.1.5 A.5. Segment Start Control Character

- **Binary Format**: 0 1 1 0 X10 X9 Y10 Y9
- **Range (hex)**: 60 to 6F

10.1.6 A.6. Coordinate Character Data Set

Each of the three characters in the coordinate data set are initially offset by 20 hex. To interpret the true bit values of these characters, subtract 20 hex from them first.

10.1.6.1 Coordinate Data Character 1

- **Binary Format**: 0 0 x8 x7 x6 x5 x4 x3
- **Initial Range (hex)**: 20 to 5F
- **Actual Range (hex)**: 0 to 3F

10.1.6.2 Coordinate Data Character 2

- **Binary Format**: `0 0 y8 y7 y6 y5 y4 y3`
- **Initial Range (hex)**: 20 to 5F
- **Actual Range (hex)**: 0 to 3F

10.1.6.3 Coordinate Data Character 3

- **Binary Format**: `0 0 x2 x1 x0 y2 y1 y0`
- **Initial Range (hex)**: 20 to 5F
- **Actual Range (hex)**: 0 to 3F

10.1.7 A.7. Unused Control Codes

Control codes 71 through 7E hex are reserved for future use.

10.2 Appendix B. RBA Script Language

10.2.1 Overview

A **script** is a text file comprised of a series of statements thath define one or more tags with associated parameters.

- Each statement must begin and end on the same line.
- A script can contain comments to explain the intent of the script and white space (space or tab characters) to enhance readability. Comments and white space are ignored by the script parser.
- Each tag description in a script describes a screen to be displayed when that tag is active and then transitions to screens associated with the other tags.
- The first tag in a script is the first tag to be active and describes the initial screen. The order of other tags is not important.
- Buttons that are not associated with a parameter will terminate the script with their default return value.

10.2.2 Comments

A comment begins with `//` and includes all characters up to the end of the line. The end of line character is not part of the comment, as it may be required to terminate a script statement.

10.2.3 Tags

Each tag description begins with a tag name. A tag name is defined by the following format:

```
[ tag_name ]
```

`Tag_name` can be any combination of non-white space characters except for the `]` character. Currently, tag names are limited to a maximum of 11 characters in length. Each tag name must be unique within the script. Any white space characters that occur between the initial [and final] are stripped from the tag name.

10.2.4 Tag Parameters

A tag definition must have one or more tag parameters. The following parameters can be used:

- **Button** – Used to control transitions from one screen to the next, control a buttons flags or terminate a script and return a key code.
- **Form** – Select the form that is to be displayed for the tag when it is active.
- **Text** – Change the text displayed on form labels.

10.2.5 Button Parameter

Button parameters are used to control the action of buttons on the form associated with the tag when it is active. This action can be to activate another tag in the script, change the appearance of another button on the form via that buttons flags, or to exit the script and return a key code. Optional flags for the button parameter are specified at the end. A maximum of eight button parameters per tag are supported.

The format for the button parameter is as follows:

```
button = buttonID , command [, flags ]
```

where

`buttonID` is the HTML ID for the button that will be effected by the button parameter.

Buttons, bitmap buttons and check boxes can be controlled by the button parameter.

`command` is

- `[ tag_name ]` specifying a new tag to go to.
- (`operator flag operandID`) specifying a change to a buttons flags where
  - `operator` is one of + (turns flag on), - (turns flag off), or ~ (toggles flag)
  - `flag` is one of h (hide button), or d (depressed button, not supported)
  - `operandID` is the HTML ID for the button that will be effected by the `operator` and `flag`
- `keycode` to cause the script to exit. The `keycode` is returned in the script response message. A `keycode` is defined as follows:
  - a control character specified by `^char`
  - a ^ character specified by `^^`
  - a non-white-space printing character (ex. A)

`flags` is one or both of:

- **h** to hide the button
- **d** to depress the button (not supported)

> **Note**
> `flags` is optional.

10.2.6 Form Parameter

A form parameter is used to associate a form with a tag. This form will be displayed when the tag is active. All of the button and text parameters refer to form elements on the specified form. The user must be careful that these parameters are consistent with the form. If no form parameter is specified for a tag, the default message form, defined by 0030_0002 in `forms.dat`, is used.

The format for the form parameter is as follows:

```
form = filename
```

where `filename` is the name of the file that contains the form. Only K3Z format forms are supported.

10.2.7 Text Parameter

A text parameter specifies the text to be displayed by a text label on the form associated with the tag.

The format for the text parameter is as follows:

```
text = textID , “text to be displayed”
```

or

```
text = textID , textfile
```

where

- `textID` is the HTML ID of the text label that will be affected by the text parameter
- `text to be displayed` is the text to be displayed in the text label associated with the `textID`
- `textfile` is a file that contains the text to be displayed in the text label associated with the `textID`

Variable substitution is performed on the second argument of both forms of the text parameter. Labels, line displays and terms and conditions text can be set with the text parameter.

For labels, the text property must be of the form `<?ivtextID?>` to allow text parameter substitution.

For line displays, the `textID` of the text parameter must be one of linedisplay2, linedisplay3, or linedisplay4 to allow text parameter substitution. linedisplay1 is reserved for RBA to display line items only.

For terms and conditions, the `textID` of the text parameter must be SELECTTEXT to allow text parameter substitution.

10.2.8 Sample Script with Comments

The following is a sample script provided to illustrate possible comments.

Annotated Sample Script

`// starts here`	// Indicates a comment in the script
`[init]`	// [name] specifies the name and start of a screen
`form=altpre.icg`	// Specifies name of form for this screen
`scroll=1` text	// Indicates that a scrolling text region is used and it uses element 1.
text=1,"%VAR4%"	// Indicates replacement text for text element 1 is literal “`%VAR4%`”.
text=2,pt%VAR5%.txt	// Indicates replacement text for text element 2 is from file `pt%VAR5%.txt`.

`button=A,[tc]`	// Indicates that when a button that returns “A” is pressed, go to screen ”tc”.
// Any other button terminates the script	
`[tc]`	// New screen definition
`form=altdisp.icg`	// Name of form to use
`scroll=1`	// Text that contains scrolling data (optional)
`text=1,pf%VAR5%.txt`	// Replace text in element 1 with text from a file.
`text=2,pn%VAR5%.txt`	// Replace text in element 2 with text from a file.
`button=B,[can]`	// Button B goes to screen “can”.
`button=@,[sign],h`	// Button @ goes to screen “sign”. (see Note 4, below.)
`button=A,(~h@)`	// Button A executes the command to toggle hidin button “@”. (see Note 5, below.)
`[sign]`	// Screen “sign” definition
`form=sign.icg`	// Use form “sign.icg”.
`button=0,A`	// The button that returns 0 terminates the script with a return key value of “A”.
`[can]`	// Screen “can” definition.
`form=altcan.icg`	// Use form “altcan.icg”.
`text=1,pc%VAR5%.txt`	// Replace text in field 1 with text from a file.
`button=A,[tc]`	// Button A goes to screen “tc”.

10.2.8.1 Notes

- Filenames and text strings are translated using the rules for form strings. If a filename or string contains %delimited variables, the tag is replaced by the contents of the variable.
- If a form name is not specified, the Terms and Conditions form is used.
- Any button that does not direct the script to another screen or run a command, terminates the script and sends a response message.
- “,” indicates that a special attribute follows. If the attribute is “h” then the button is hidden when the form first loads. If the attribute is “d”, the button is shown as depressed. “d” is intended for radio buttons and check boxes only.
- Valid commands are:
    - `-h` Show a button.
    - `+h` Hide a button.
    - `~h` Toggle a buttons hide state.

10.3 Appendix C. PayPal Overview

Authorization using PayPal is included in RBA. Some configuration is required in addition to the need to obtain and add a PayPal public key. This section outlines those configuration needs.

10.3.1 C.1. Minimum Production Requirements

Minimum requirements for using PayPal with RBA include the following:

- RBA Version 2.7.3 or higher.
- Access to GMT Date and Time in addition to Local Date and Time. Configuration is required. (In addition to the information in this Appendix, see also section Configuring GMT Variables for PayPal Authorization.)
- A PayPal public key from PayPal or Ingenico. Follow the Operation and Product Support Guide for this requirement.

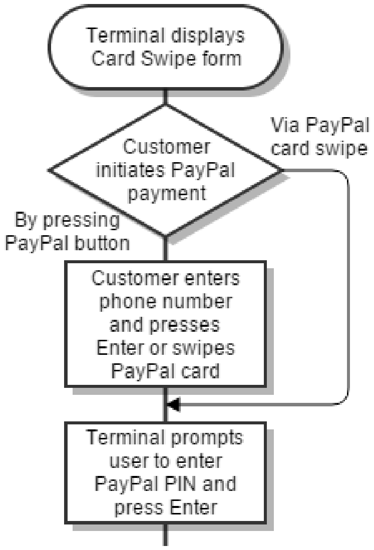
For testing purposes, use `GENERIC_T.PEM`, the test key provided by PayPal. See section PayPal Configuration (paypal.dat) for additional information.

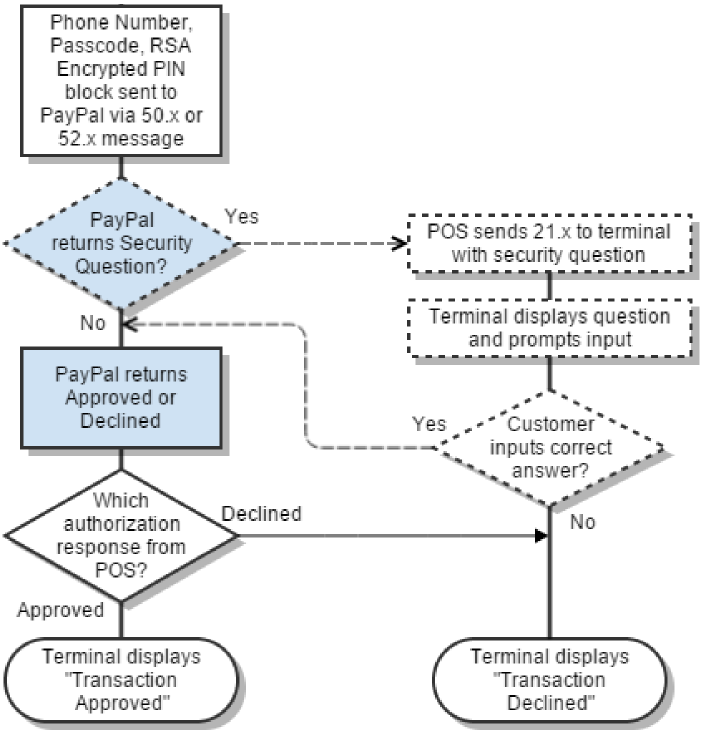
The PayPal PIN block is 138 bytes long. Due to the limitations of the EFT message protocol, the PIN block must be base64 encoded. This expands the PIN block to 184 bytes.

For RBA to recognize PayPal cards, either internal (see parameter '0099_0001') or external (see parameter '0005_0002') BIN lookup must be enabled. The payment type determined by the method used must match the PayPal card table entry in `cards.dat`.

10.3.2 C.2. PayPal Validation Flow

The PayPal validation flow is as follows. The section with dotted lines is an optional flow.


Terminal displays Card Swipe form
Customer initiates PayPal payment
Via PayPal card swipe
By pressing PayPal button
Customer enters phone number and presses Enter or swipes PayPal card
Terminal prompts user to enter PayPal PIN and press Enter


Phone Number, Passcode, RSA Encrypted PIN block sent to PayPal via 50.x or 52.x message
PayPal returns Security Question?
Yes
POS sends 21.x to terminal with security question
Terminal displays question and prompts input
Customer inputs correct answer?
Yes
No
PayPal returns Approved or Declined
Which authorization response from POS?
Declined
No
Approved
Terminal displays "Transaction Approved"
Terminal displays "Transaction Declined"

PayPal Validation Flow

The PayPal enabled 50.x: Authorization Request Message includes values for certain fields that will distinguish PayPal transactions, as outlined, below.

Transaction Code (11th field) (Offset 45, Length 2):

- The two-digit transaction code values for PayPal include
  - 70 = Sale
  - 71 = Void
  - 72 = Return
  - 73 = Void Return
- '0011_0007' in cards.dat: Key ID = 71, Card Type = G (PayPal).

Account Data source has a new value (Offset 61, Length 1):

- Account Data Source “P” has been added where “P” = Phone Number.

10.3.3 C.3. Configuration

10.3.3.1 PayPal.dat

See section PayPal Configuration (paypal.dat) for details.

10.3.3.2 Forms

A number of new forms containing the PayPal logo have been introduced to RBA for use with PayPal's authorization flow. In all but one case, the DFS Data Index is the same as is used for RBA's standard flow.

> See also Form Files (forms.dat) for additional information.

Forms that can be used with PayPal include the following:

PayPal Compatible Parameters

Parameter Name	DFS Data Index	Default Value	Description
Offline Form	0030_0001	PPOFFLINE.K3Z	This is the form that the terminal will display when it is offline. The standard RBA form file name is `OFFLINE.K3Z`.
Swipe Card Form	0030_0004	PSWIPE.K3Z	This is the form that the terminal will display to prompt the cardholder to swipe his magnetic stripe card. The standard RBA flow version of this form is `SWIPE.K3Z`.
Swipe Card Form with Language Buttons	0030_0005	PLSWIPE.K3Z	This is the form that the terminal will display to prompt the cardholder to select a language and to swipe his magnetic stripe card. Displayed if Combine Language Swipe Screens parameter ('0007_0004') is set to 1, <combine screens>. The standard RBA flow version of this form is `LSWIPE.K3Z`.
Swipe card with Contactless	0030_0021	CPSWIPE.K3Z	This is the form that the terminal will display to prompt the cardholder to tap his contactless card on the terminal. The standard RBA flow version of this form is `CSWIPE.K3Z`.

Parameter Name	DFS Data Index	Default Value	Description
Swipe with Language + Contactless	0030_0022	CPLSWIPE.K3Z	This is the form that the terminal will display to prompt the cardholder to select a language and to tap his contactless card on the terminal. Displayed if Combine Language Swipe Screens parameter ('0007_0004') is set to 1, <combine screens>. The standard RBA flow version of this form is `CLSWIPE.K3Z`.
PayPal Data Input	0030_0027	PPALINP.HTM	Used with the 21.x: Numeric Input Request Message, this is the form used by the cardholder to input a variable Code to the terminal. The file type for this form is .HTM (not .K3Z). This form does not share a DFS Data Index ID number with an equivalent form in the standard RBA flow. The standard RBA flow version of this form is DFS Data Index '0030_0016' (`INPUT.K3Z`).
PayPal PIN Entry	PayPal PIN Entry	PPALPCAN.HTM	Requests the cardholder's PayPal PIN for PayPal authorization. This form is in .HTM file format (not .K3Z).
PayPal Please Wait	0030_0029	PPWAIT.K3Z	Requests that the PayPal cardholder wait for approval or denial.

10.3.4 C.4. Calculating GMT Offset (Variable 205)

As described in Configuring GMT Variables for PayPal Authorization, GMT Offset is a value in seconds equal to the difference between Local and GM Time. It is especially important to set the GMT variables in addition to the Local variables (`201-202`) when in a mixed (U32 and Telium) financial environment.

Using the 28.x Set Variable Request message, this variable (variable `205`) must be entered in seconds.

- Local – GMT = difference, or, Local – difference = GMT.
- There are 3600 seconds in an hour.

10.3.4.1 Scenario 1

- If Local time is 6AM (0600 hours) and GMT is 10AM (1000 hours), the difference in time is -4 hours: 0600 - 1000 = -0400.
- -4 hours X 3600 seconds = -14400 seconds.
- '-14400' (with the negative sign in front of the numerals) is the value to enter into variable `205`.

10.3.4.2 Scenario 2

- If Local time is 1 PM (1300 hours) and GMT is 10 AM (1000), the difference in time is 3 hours, therefore, the value to enter into variable `205` is '10800'.

When calculating, remember to pay attention to any adjustments in time due to Daylight Savings Time, British Summer Time, etc., as these events increase or decrease the difference.

See also www.greenwichmeantime.com for additional information.

10.4 Appendix D. RBA Best Practices

10.4.1 D.1. Working With RBA

10.4.1.1 Host Interface Messaging Tips

As a guideline, here are general tips:

- When using status polling, status requests should be sent no more than once per second.
- Follow the protocol to be sure that every properly formatted message sent by the terminal ([STX][message][ETX][LRC]) receives an acknowledgement from POS.
- If you are expecting a response from the terminal, wait for the response before sending any additional request messages.
- Be aware of the possibility of receiving unsolicited RBA host interface messages from the terminal in situations including, but not limited to:
  - 19.x BIN Lookup Message messages sent by the terminal following a card swipe, depending on RBA configuration.
  - Responses to status polling from POS.
  - Responses to transaction totals sent from POS.
  - 50.x Authorization Request messages sent once the customer confirms the Account, Amount and Tender involved in the transaction.
- Once the POS has sent transaction totals to the terminal, status polling can be disabled until a 10.x Hard Reset Message or 50.x: message is received from the terminal.
- When using 19.x BIN Lookup Message message for Tender Lookup, `config.dfs` parameter '0005_0002' can be set to option '2' so that the 19.x message is sent to the POS only after the amount has been received.

- If the customer does not provide input when an on-demand function is called, a '15.6' message needs to be sent to stop action and return to the previous state. If state is not a concern, RBA can be configured to accept any on-demand message (e.g., signature, PIN, clear text, form display, card read) and disregard the previous request. This is the preferred alternative to the standard RBA flow. When using on-demand functions with the default RBA flow, the application returns to its previous state when the on-demand function is complete to preserve the standard transaction flow.
- When using the 28.x Set Variable Request message, use the Response Type parameter to suppress a response if it is not explicitly needed. This replaces the full response message with a single ACK returned to the POS. This can be useful to speed up messaging during scrolling receipt updates where multiple items are sent in short periods of time.

See Host Interface Messages for detailed information on RBA Host Interface messages.

10.4.1.2 Retrieving EPS Encrypted Data

10.4.1.2.1 Building Off an Established Implementation

When an existing RBA configuration is applied to a load supporting EPS encryption, the following changes take place to support EPS functionality:

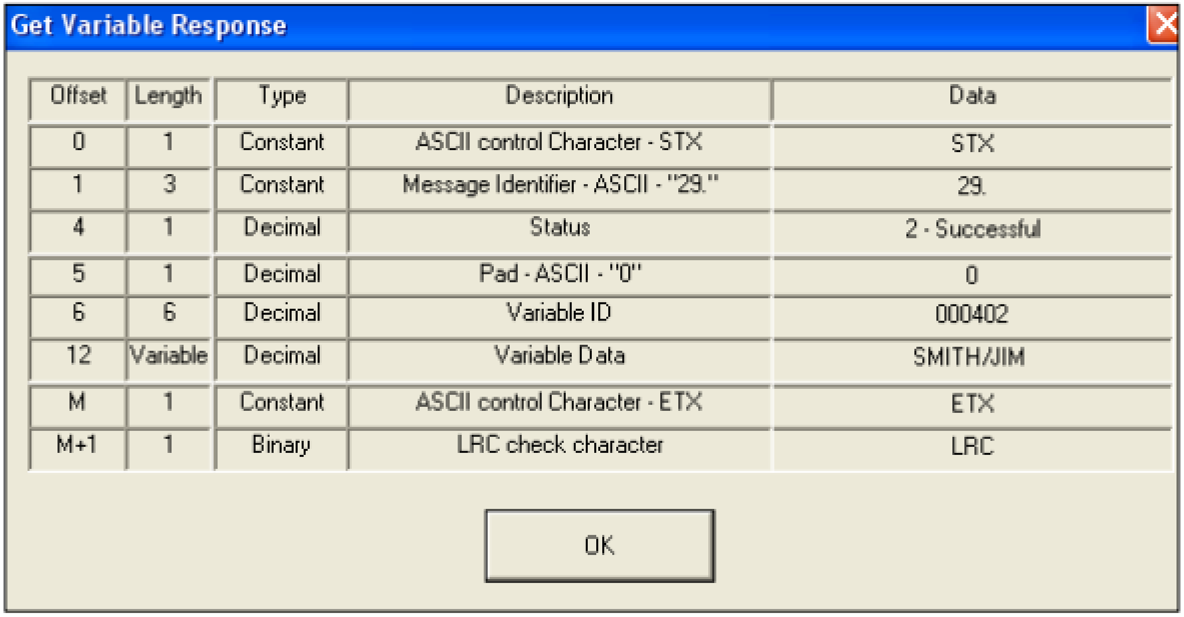
- Cardholder data is encrypted.
- 19.x Message is extended to support EPS functionality.
- New interface call is added to retrieve data for receipt processing.

RBA’s tender processing flow continues to works normally.

Variable number of digits specified in '0005_0008' left in the clear for BIN lookup (set with '0005_0008').

- Middle digits of card number replaced with zeros.
- Variable number of ending digits specified in '0003_0016' left in the clear for printing purposes, '1159' in the example shown above.
- Encrypted Track 1 card data is appended after field separator.
- Encrypted Track 2 card data is appended after field separator.

To retrieve the Account Name for receipt printing, a 29.x Get Variable message must be issued with function type of 402.



Offset	Length	Type	Description	Data
0	1	Constant	ASCII control Character - STX	STX
1	3	Constant	Message Identifier - ASCII - "29."	29.
4	1	Decimal	Status	2 - Successful
5	1	Decimal	Pad - ASCII - "0"	0
6	6	Decimal	Variable ID	000402
12	Variable	Decimal	Variable Data	SMITH/JIM
M	1	Constant	ASCII control Character - ETX	ETX
M+1	1	Binary	LRC check character	LRC

29.x Message Response with Customer Name

10.4.1.2.2 Taking Direct Control of Retrieving Encrypted Data

Even if RBA is configured to omit track data from the 19.x message, the POS can still retrieve encrypted track data with 29.x messages:

- 29.x message with function type of 406 returns EPS encrypted transaction data.
- 29.x message with function type of 407 returns encrypted Track 2 data.

> This method cannot be used to obtain Account Name / Card Number "in the clear." Transaction data returned when using 406 or 407 will consist of the entire encrypted data string.

As with the previous scenario, a 29.x message with function type of 402 can be sent to retrieve the Account Name for receipt printing.

10.4.1.2.3 Retrieving Encrypted Data from a Standard 50.x Message

RBA also sends encrypted Track 1 & 2 data as part of the 50.x Authorization Message.

As with previous scenarios, a 29.x message with function type of 402 can be sent to retrieve the Customer Name for receipt printing.

10.4.2 D.2. Customizing RBA

10.4.2.1 Editing a Prompt

To edit an RBA prompt, follow these steps:

1. On your development PC, open the RBA XML file that contains the prompt(s) that you wish to edit.
2. Edit the existing prompt as desired, then save a new version of the XML file.

3. Build and load a new RBA load package using the updated XML file.

10.4.2.2 Adding a Custom Prompt

To add a new prompt to RBA, follow these steps:

1. On your development PC, open the `CUSTPROMPT.xml` file.
2. Add an entry for the new prompt in the corresponding section of the `CUSTPROMPT.xml` file. Remember to add entries for the new prompt in all languages.
3. Save a new version of `CUSTPROMPT.xml`.
4. Build and load a new RBA load package using the updated XML file.

10.4.2.3 Editing a Form

To edit an RBA form, follow these steps:

1. On your development PC, open the RBA form file (*.K3Z) that you wish to edit.
2. Edit the form file as desired, then save a new version of the K3Z file.
3. Build and load a new RBA load package using the updated K3Z file.

10.4.2.4 Adding a Form

To add a new form to RBA, follow these steps:

1. Design and build the new form using Ingenico's Form Builder tool.
2. Place the new form in the RBA package "media" folder for the intended terminal.
3. Open the intended terminal's RBA package manifest (*.XML) file, and add an entry for the new form in the <UNSIGNED CONTENT> section. Save your changes to the manifest file.
4. Build and load a new RBA load package using the updated manifest file.

10.4.2.5 Editing Config.dfs

To edit `config.dfs`, follow these steps:

1. On your development PC, open `the config.dfs`.
2. Edit the desired `config.dfs` parameters, then save your edits. The updated `config.dfs` should be saved in the `DFS_SRC` directory, e.g., `DP-RGEN00-15.0.5.0357\config\DFS_SRC`.
3. After editing `config.dfs`, generate updated RBA DAT files by running the `GEN_TGZ.BAT` file found in the same folder as the RBA package's CONFIG folder. It is a menu-driven DOS script which also creates the TGZ package after generating the DAT files. A new "package" folder is created containing the M46 and TGZ files.
4. Build and load a new RBA load package using the updated DAT files.

10.4.3 D.3. Packaging a New RBA Load

To generate an RBA load package for an Ingenico terminal, complete the folloing steps:

1. Click the `XXXXXXPackageGZ.BAT` file for the terminal (where "XXXXXX" is the terminal model).

2. The new package is placed in `RBA Data and Parameters\package\<terminal model>`.

When the package is built, it can be loaded to the terminal using LLT.

10.4.4 D.4. Message Retry Best Practices

10.4.4.1 POS Retries

When sending a message, POS should wait up to 3 seconds for an ACK.

- If the terminal sends no response, retry, wait up to 3 seconds again, up to three total tries.
- If the terminal sends a NAK, resend the message immediately. See the RBA Developer's Guide, section Link-Level Responses on how to handle NAK responses. Reasons for NAK response include:
  - ETX is missing.
  - LRC value mismatched or missing. LRC value counts the full message length including the ETX end sentinel, but excludes the STX and LRC character.
  - Message/packet too large.

RBA ignores messages that exclude the STX sentinel, and does send an ACK or NAK in response.

10.4.4.2 RBA Retries

RBA resends a message if:

- Immediately, up to nine times if it receives a NAK from the POS. Once it has sent ten total messages, it will no longer retry. If a large number of retries still results in a NAK, check your message format.
- After three seconds, up to two times if it receives no ACK or NAK from the POS. Once it has sent three total messages, it will no longer retry.

10.4.4.3 Other Information

When RBA receives a message and replies with an ACK, it does not always respond immediately. After responding with an ACK, it adds the requested process to the queue for processing. Wait 3-4 seconds after receiving an ACK for the request to process before sending a retry. This will give RBA time to finish its current task and respond.

For each message with an expected response that RBA receives and replies an ACK, RBA sends a response, but take care not to overload the terminal queue, as it does not respond if overloaded.

10.5 Appendix E: eWIC Implementation

10.5.1 eWIC Overview

10.5.1.1 Overview

Women, Infants and Children (WIC) is a federal assistance program which provides healthcare and nutrition for low-income women who are pregnant, breastfeeding, and/or have infants and children under the age of five.

In the past, WIC authorities disseminated benefits to individuals in the past via paper food stamps, online MSR Electronic Benefits Transfer (EBT) cards, and offline EBT smart cards. Please refer to ANSI 2005 for specification details unless stated otherwise.

10.5.1.2 Card Inventory

WIC smart cards typically maintain a four-month inventory plus a backup recovery month in the event of any card errors. Each month's inventory is provided as a list of quantities of specific items or item types (e.g., dairy, fruits and vegetables, breads) available to debit for each month, irrespective of cash value. WIC items can only be debited from the current month's inventory. Inventory balances cannot be transferred or carried over into later months' inventories.

10.5.1.3 Modes

WIC smart cards have three primary usage modes as described in the table below.

Usage Mode Breakdown

Usage Mode	Description
Redemption Mode	Used by retailers to debit a WIC smart card's benefits balance during normal transactions.
Training Mode	Used by retailers to train employees on how to process WIC transactions. A WIC smart card's benefit balance is typically not debited during training transactions.
Certification Mode	Used by WIC authorizers to certify WIC systems. Debiting the WIC smart card's benefit balance can be toggled on or off for test transactions.

10.5.1.4 Primary Types of WIC Smart Cards

There are three primary types of WIC smart cards. Card types are set by a specific digit in the WIC smart card PAN to any of the following values:

WIC Smart Card Type Breakdown

Card Type	Identifier	Description and Usage
Production Card	1 or 7	Issued to and used by women or families receiving WIC benefits. Intended primarily for use during redemption mode.
Training Card	9	Used by retailers during training mode.
Test Card	0	For training and/or certification modes. May also be used to test redemption mode.

The specific PAN digit that indicates WIC smart card type varies by WIC authority. For instance, most WIC authorities indicate the WIC smart card type in the eighth digit of the PAN. whereas Texas-issued smart cards indicate the WIC smart card type in the seventh PAN digit.

As an example, the seventh digit of the following Texas-issued smart card account is a 1, indicating that the card is a production card.

```
5077171016802632028
```

As another example, the eighth digit of the following Wyoming-issued smart card account is a 9, indicating that the card is a training card.

```
5053495900027662
```

10.5.2 eWIC WMP Messages

10.5.2.1 WMP Message Specifications

The POS interacts with WIC cards and the WIC library via WIC Message Protocol (WMP) messages. There are three specifications currently in effect that detail WIC messaging, typically abbreviated as the release year and the authority that published the specification:

- 2003 Texas (State of Texas WIC Smart Card Interoperability Specification)
- 2005 ANSI (American National Standards Institute WIC Smart Card Interoperability Specification)

The WIC library module most closely adheres to the 2005 ANSI specification in order to conform and interoperate with IBM/Toshiba's eWIC functionality.

> Half of the WIC transaction messages initiate and/or require cardholder-to-terminal interaction, some of which may be performed in varying sequences of steps. Other messages may be fragmented and require multiple messages to be processed or sent.

10.5.2.2 WMP Request/Response Message Pairs

Unlike standard RBA messages which are identified by a two-digit message number followed by a period character (e.g., 01.x: Online Message, 10.x: Hard Reset Message), WMP messages start with a single underscore character followed by a two-digit message number identifier (e.g., _10/_11 Get PAN Messages, _20/_21 Read WIC Balance Message).

The POS sends even-numbered request messages to the terminal and waits for next-sequential odd-numbered response messages from the terminal, as described in the table below.

WMP Messages

Request	Response	Description
_00	_01	WSPM (WIC Service Provider Module) Request/Activation.
_10	_11	Get PAN Card Number Request/Response.
_20	_21	Read WIC Balance.
_30	_31	Debit WIC Balance.
_40	_41	Block Card.

Request	Response	Description
_50	_51	End WIC Transaction.
_60	_61	Authenticate WIC User (i.e. prompt for WIC PIN).
_70	_71	WPSM Shutdown/Deactivation.
_80	_81	Remove WIC Card Request.
_99	_99	Cancel Transaction Request.

> _99 request and response messages are included as a requisite from Toshiba as an option to cancel debit from the POS, though these are redundant as a 10.x: can cancel the transaction and send a _31008C cancel response message.

10.5.2.3 WIC Transaction and RBA Exclusivity

WIC transactions and standard RBA transactions are exclusive of each other and cannot be processed concurrently. In other words, a WIC transaction cannot start in the middle of another RBA transaction (see Note below), nor may a RBA transaction or on-demand messages interrupt a WIC transaction (see below table Allowed Non-WIC Messages for exemptions). In addition, WMP messages are handled atomically and typically cannot interrupt one another until each previous message has been completely handled and responded to.

> If a WIC transaction is prompted during a standard RBA transaction, the following error will occur: `WSPI_TENDER_MISMATCH` error response message _119993 will be sent if the terminal receives _10 Get PAN Card Number Request but non-WIC transaction already started.

10.5.2.4 Non-WIC Exemptions

Non-WIC RBA messages that may be received and handled during a WIC transaction are listed in the table below.

Allowed Non-WIC Messages

Message Type	Message ID
Authentication messages	• 93.x: Terminal Authentication Messages
Configuration messages	• 61.x: Configuration Read
File messages	• 63.x: Find File

Message Type	Message ID
Peripheral messages	• 91.x: Print Message • 94.x and 95.x: Barcode Configuration Messages
Reset messages	• 00.x: Offline Message • 01.x: Online Message • 10.x: Hard Reset Message • 30.x: Advertising Request Message (On-Demand) • 97.x: Reboot • 15.0 and 15.8 Soft Reset Messages While '15.0' and '15.8' soft reset messages are allowed during a WIC transaction, all other 15.x messages are disallowed.
Status messages	• 07.x: Unit Data Request • 08.x: Health Stat • 11.x: Status Message • 22.x: Application ID Request
Variable messages	• 28.x: Set Variable Request • 29.x: Get Variable Request • 70.x: Update Form Element Message

10.5.2.5 WMP Message Format

The following tables describe the WIC Request and Response message formats. See Appendix WIC Transaction Flows for an explanation of usage.

_01 WSPM Reset/Activation Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	Message identifier: WSPM Reset/Activation Response – "_01".
5	4	Alphanum	Error code. See eWIC Error Codes/Displays. Sends "0000" if no error.
6	2	Numeric	Number of WIC authorities – "NN".
8	Variable	Alphanum	List of WIC authorities – "??....".
M	Variable	Alphanum	DLL versions for WIC authorities – "VERS.."
N	1	Constant	ASCII control character – ETX.

Offset	Length	Type	Description
N+1	1	Binary	LRC check character.

All WMP format messages use '0000' as the error code if there is no error to report.See eWIC Error Codes/Displays for a list of all error codes returned during WIC transactions.

_11 Get WIC Card PAN Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	Message identifier: Get Card PAN Response – '_11'.
4	4	Alphanum	Error code.
8	Variable	Alphanum	WIC authority – "??....".
M	2	Numeric	PAN length – "LL".
M+2	Variable	Numeric	PAN – "PAN...".
N	1	Constant	ASCII control character – ETX.
N+1	1	Binary	LRC check character.

_21 Read WIC Balance Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	Message identifier: Read WIC Balance Response– "_21".
4	4	Alphanum	Error code.
8	Variable	Alphanum	WIC Authority – "??".
M	15	Alphanum	Benefits Issuing Entity.
M+15	2	Alphanum	Sequential block number. '00' by default.
M+17	2	Numeric	Total number of items in WIC balance. This value is repeated every block.

Offset	Length	Type	Description
M+19	2	Numeric	Number of items in the current block.
M+21	Variable	Alphanum	WIC item entries, up to 10 bytes each.
N	1	Constant	ASCII control character – ETX.
N+1	1	Binary	LRC check character.

_31 Debit WIC Balance Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	Message identifier: Debit WIC Balance Response– "_31".
4	4	Alphanum	Error code.
8	Variable	Alphanum	WIC Authority – "??".
M	2	Numeric	Debit signature length – "LL".
M+2	Variable	Alphanum	Debit Signature – "SIG...".
M	1	Constant	ASCII control character – ETX.
M+1	1	Binary	LRC check character.

_51 Read WIC Balance Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	Message identifier: End WIC Transaction Response – "_51".
4	4	Alphanum	Error code.
8	Variable	Alphanum	WIC Authority – "??".
M	1	Constant	ASCII control character – ETX.
M+1	1	Binary	LRC check character.

_61 Authenticate WIC User Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	Message identifier: Authenticate WIC UserResponse – "_61".
4	4	Alphanum	Error code.
8	2	Numeric	PIN validity: • `00` = Valid PIN entered. • `FF` = No valid PIN entered.
10	1	Constant	ASCII control character – ETX.
11	1	Binary	LRC check character.

10.5.2.6 _99 eWIC Reset Message

When the WIC tender is cancelled by the cashier on while the terminal displays the "Update Card?" screen, the POS sends a _99 eWIC Reset Message. It is always sent after the _30 Debit Balance message, but before the _31 Debit Balance Response message is returned. The terminal responds with a _99 eWIC Reset Response message. The only data contained in this message is a 1 digit field indicating whether the cancel was successful or not, as shown in the table below:

_61 Authenticate WIC User Response Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX.
1	3	Constant	Message identifier: eWIC Reset Message – '_99'.
4	1	Binary	Success status: • 0 – Cancellation request unsuccessful. Card has been updated. • 1 – Cancellation successful. No update was applied.
5	1	Constant	ASCII control character – ETX.
6	1	Binary	LRC check character.

10.5.3 WIC Transaction Flows

10.5.3.1 Messages in WIC Transactions

The only required WMP message is the _00 WIC Service Provider Module Request/Activation. All other steps are optional so far as the WIC library is concerned. Other flows are possible (e.g., a Read Card Balance flow that skips the debit balance step), though the typical flow is most likely the only reasonable flow. Abnormal sequences of messages might not successfully read/write WIC card data. Thus, a POS/terminal system must be certified as a paired system for compliance following expected WIC procedures/message flows.

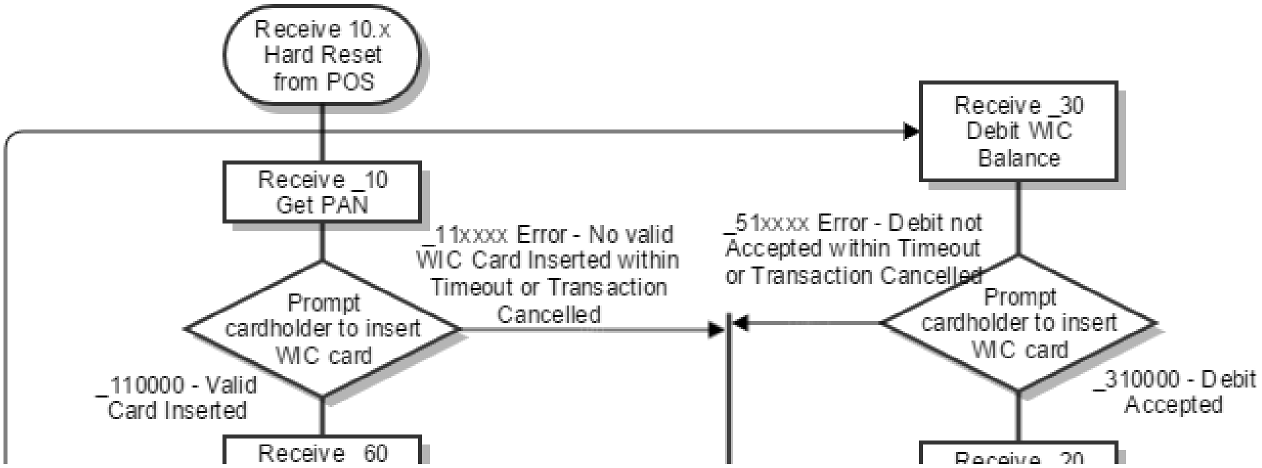
> Half of the WIC transaction messages initiate and/or require cardholder-to-terminal interaction (see also eWIC WMP Messages, Section 'Variable/Interactive messages'). Most balance validation and debit calculations must be performed by the terminal's WIC library (WSPI layer) rather than the smart cards themselves.

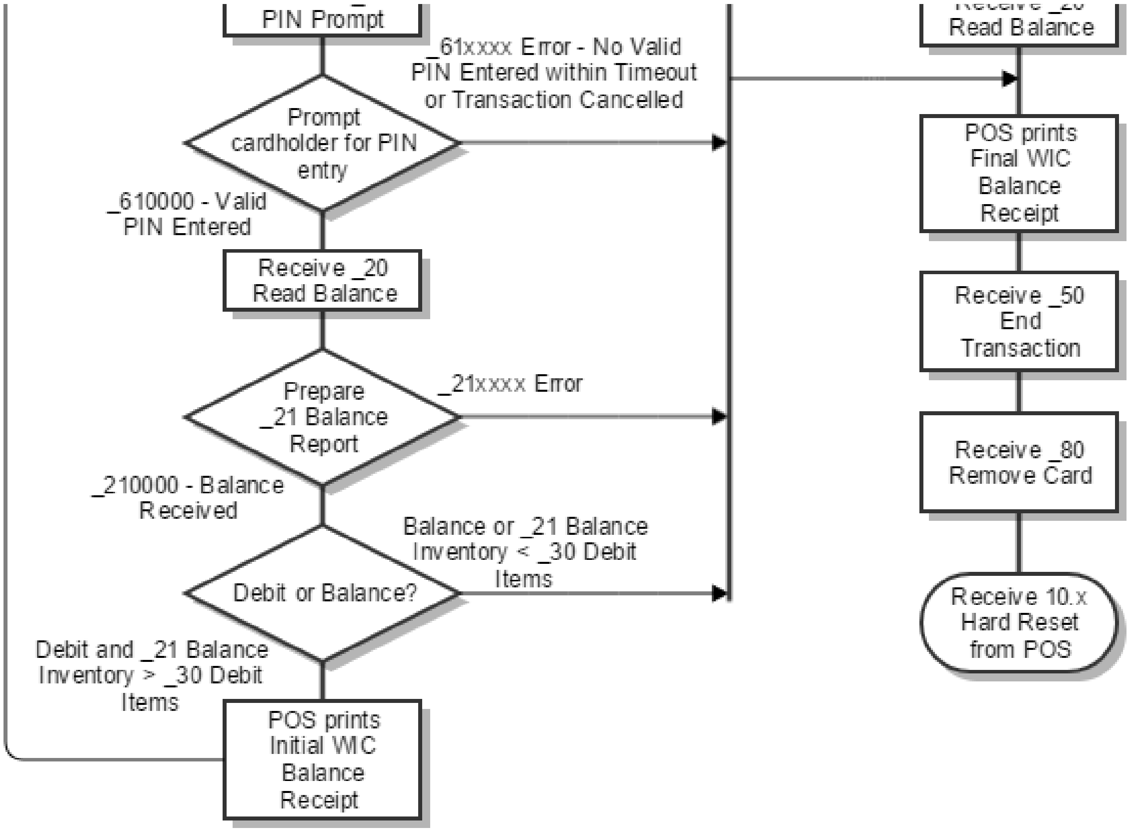
10.5.3.2 POS-Terminal Server/Client System

POS-terminal interaction is a server/client system. Because of this, once the POS starts a WIC transaction with the _10 Get PAN Message, the terminal defers all WIC transaction control to the POS. In other words, once a WIC transaction begins, the terminal will not automatically transition to any display or state without either receiving a message from the POS or input from the cardholder. For WIC PIN entry timeouts, the terminal will automatically cancel the transaction by sending an _61008E PIN Entry Cancelled Error response message but will not automatically return to online state. The terminal will instead wait for the POS to end the WIC transaction and reset the terminal via _50 End WIC Transaction and 10.x Hard Reset Message.

In the case that a WIC smart card is inserted and removed prior to receiving the _10 Get PAN Message, the terminal will briefly display "Card removed / Transaction cancelled" before returning to the online swipe/insert card form since a WIC transaction has not officially started. In order to avoid interrupting the EMV/WIC transaction flow, 09.x Card Status Messages conveying card insertion status (e.g., card inserted, card removed) may only be sent before the transaction is initiated.

Use the following suggested message flows as a guide for WIC transactions:


Receive 10.x Hard Reset from POS
Receive _30 Debit WIC Balance
Receive _10 Get PAN
_11xxxx Error - No valid WIC Card Inserted within Timeout or Transaction Cancelled
_51xxxx Error - Debit not Accepted within Timeout or Transaction Cancelled
Prompt cardholder to insert WIC card
Prompt cardholder to insert WIC card
_110000 - Valid Card Inserted
_310000 - Debit Accepted
Receive _60
Receive _20


PIN Prompt
_61xxxx Error - No Valid PIN Entered within Timeout or Transaction Cancelled
Prompt cardholder for PIN entry
_610000 - Valid PIN Entered
Receive _20 Read Balance
Prepare _21 Balance Report
_21xxxx Error
_210000 - Balance Received
Balance or _21 Balance Inventory < _30 Debit Items
Debit or Balance?
Debit and _21 Balance Inventory > _30 Debit Items
POS prints Initial WIC Balance Receipt
Receive _20 Read Balance
POS prints Final WIC Balance Receipt
Receive _50 End Transaction
Receive _80 Remove Card
Receive 10.x Hard Reset from POS

WIC Transaction Flow

10.5.3.3 *eWIC Sample Message Flows*

The table below illustrates the message flow for enabling WIC transactions.

Initial WIC Configuration

POS	Terminal	Message	Description	11.x Status Return Message
→		`00.`	The POS commands the terminal to go offline.	`11.00Lane Closed`
			The terminal displays the offline message "This Lane Closed".	`11.00Lane Closed`

POS	Terminal	Message	Description	11.x Status Return Message
→		60.20[GS]1 [GS]1	The POS enables WIC transactions.	`11.00Lane Closed`
→		`01.`	The POS commands the terminal to go offline.	`11.01Slide Card`
←		`01.00000000`	The terminal returns an online message.	`11.01Slide Card`
			The terminal displays the online swipe/ insert screen.	`11.01Slide Card`
→		`_00...`	POS configures WIC authorities.	`11.01Slide Card`

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_010000NN??...VERS..</code></td><td><table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td><code>_01</code></td><td>Message identifier: WSPM Reset / Activation response.</td></tr>
<tr><td><code>0000</code></td><td>No error code.</td></tr>
<tr><td><code>NN</code></td><td>Number of WIC authorities.</td></tr>
<tr><td><code>??...</code></td><td>List of WIC authorities.</td></tr>
<tr><td><code>VERS..</code></td><td>DLL versions for WIC authorities.</td></tr>
</table></td><td><code>11.01Slide Card</code></td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.01Slide Card`

10.5.3.4 *Balance Transaction Flow Breakdown*

The two following tables explain the flow of balance checking WIC transactions:

Message order for Typical WIC Card Balance Message Flow

Message ID	Description	Information
`_00`	WSPM Reset/Activation	Typically used only once following a reboot or POS logon, but may precede every transaction.
`_10`	Get Card PAN	May be sent before or after WIC card insertion.
`_60`	Authenticate WIC User	The terminal prompts user for PIN up to a maximum number of retries.
`_20`	Read WIC Balance	Performed by WIC library.
`_50`	End WIC Transaction	
`_80`	Remove Card	
`_70`	WSPM Shutdown/Deactivation	Typically used once following POS logoff but may be ignored altogether.

Successful Balance Inquiry Transaction

POS	Terminal	Message	Description	11.x Status Return Message
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.01Slide Card`

POS	Terminal	Message	Description	11.x Status Return Message
			The terminal displays the online swipe/ insert screen.	`11.01Slide Card`
→		`_10`	The POS sends a WIC Get PAN message.	`11.19WIC`
			The terminal displays the "Please insert card" screen.	`11.19Please insert card`
			The cardholder inserts a WIC smart card. If a WIC smart card is inserted before _10 Get PAN Message, the terminal sends a '09.010201I' message and The terminal displays "WIC-?? / Please wait... Do not Remove Card".	`'11.19WIC-?? / Please wait... Do not Remove Card'` is also returned if a WIC smart card is inserted before _10 Get PAN Message is sent and processed.

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_110000??LLPAN...</code></td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td><code>_11</code></td><td>Message identifier: Get Card PAN response message.</td></tr><tr><td><code>0000</code></td><td>No error code.</td></tr><tr><td><code>??...</code></td><td>WIC authority.</td></tr><tr><td><code>LL</code></td><td>PAN length.</td></tr><tr><td><code>PAN...</code></td><td>PAN.</td></tr></table></td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td><code>11.19WIC-?? / Please wait... Do not Remove Card</code></td></tr>
<tr><td>→</td><td></td><td><code>_60</code></td><td>The POS sends a WIC Authenticate cardholder message.</td><td><code>11.03Please enter your PIN</code></td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "Please enter your PIN".</td><td><code>11.03Please enter your PIN</code></td></tr>
<tr><td></td><td></td><td></td><td>The cardholder enters a valid PIN.</td><td><code>11.03Please enter your PIN</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal briefly displays "PIN OK".</td><td><code>11.19 WIC</code></td></tr>
<tr><td>←</td><td></td><td><code>_61000000</code></td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td><code>_61</code></td><td>Message identifier: Authenticate WIC User response message.</td></tr><tr><td><code>0000</code></td><td>No error code.</td></tr><tr><td><code>00</code></td><td>Valid PIN entered.</td></tr></table></td><td><code>11.19 WIC</code></td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
			The terminal displays "WIC-?? / Please wait... Do not Remove Card".	`11.19WIC-?? / Please wait... Do not Remove Card`
⟶		`_20`	The POS sends a WIC Read Card Balance message.	`11.19WIC`

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td>_210000??
...</td><td><table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td>_21</td><td>Message identifier: Read WIC Balance response message.</td></tr>
<tr><td>0000</td><td>No error code.</td></tr>
<tr><td>??</td><td>WIC authority.</td></tr>
<tr><td>...</td><td>Current month balance data.</td></tr>
</table></td><td>11.19WIC</td></tr>
<tr><td></td><td></td><td></td><td>The POS prints a balance receipt.</td><td>11.19WIC</td></tr>
<tr><td>→</td><td></td><td>_80</td><td>The POS sends a WIC Remove Card message.</td><td>11.20Remove card</td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
←		`_81`	The terminals sends a WIC Remove Card response message.	`11.20SMC`
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.20SMC`

10.5.3.5 Debit Transaction Flow Breakdown

The following three tables explain the flow of WIC debit transactions:

Typical WIC Debit Message Flow

Message ID	Description	Information
`_00`	WSPM Reset/Activation	Typically used only once following a reboot or POS logon, but may precede every transaction.
`_10`	Get PAN Request	May be sent before or after WIC card insertion.
`_60`	Authenticate WIC User	The terminal prompts user for PIN up to a maximum number of retries.
`_20`	Read WIC Balance	This message checks the card's starting balance. Performed by WIC library.
`_30`	Debit WIC Balance	Performed by WIC library.
`_20`	Read WIC Balance	This message checks the card's ending balance. Performed by WIC library.
`_50`	End WIC Transaction	
`_80`	Remove Card	
`_70`	WSPM Shutdown/Deactivation	Typically used once following POS logoff, but may be ignored altogether.

Successful Debit Transaction

POS	Terminal	Message	Description	11.x Status Return Message
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.01Slide Card`
			The terminal displays the online swipe/ insert screen.	`11.01Slide Card`
→		`_10`	The POS sends a WIC Get PAN message.	`11.19WIC`
			The terminal displays the "Please insert card" screen.	`11.19Please insert card`
			The cardholder inserts a WIC smart card. If a WIC smart card is inserted before _10 Get PAN Message, the terminal sends a '09.010201I' message and The terminal displays "WIC-?? / Please wait... Do not Remove Card".	'`11.19WIC-?? / Please wait... Do not Remove Card`' is also returned if a WIC smart card is inserted before _10 Get PAN Message is sent and processed.

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_110000??LLPAN...</code></td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td><code>_11</code></td><td>Message identifier: Get Card PAN response message.</td></tr><tr><td><code>0000</code></td><td>No error code.</td></tr><tr><td><code>??...</code></td><td>WIC authority.</td></tr><tr><td><code>LL</code></td><td>PAN length.</td></tr><tr><td><code>PAN...</code></td><td>PAN.</td></tr></table></td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td>→</td><td><code>_60</code></td><td>The POS sends a WIC Authenticate cardholder message.</td><td><code>11.03Please enter your PIN</code></td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "Please enter your PIN".</td><td><code>11.03Please enter your PIN</code></td></tr>
<tr><td></td><td></td><td></td><td>The cardholder enters a valid PIN.</td><td><code>11.03Please enter your PIN</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal briefly displays "PIN OK".</td><td><code>11.19 WIC</code></td></tr>
<tr><td>←</td><td></td><td><code>_61000000</code></td><td><table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td><code>_61</code></td><td>Message identifier: Authenticate WIC User response message.</td></tr>
<tr><td><code>0000</code></td><td>No error code.</td></tr>
<tr><td><code>00</code></td><td>Valid PIN entered.</td></tr>
</table></td><td><code>11.19 WIC</code></td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
			The terminal displays "WIC-?? / Please wait... Do not Remove Card".	`11.19WIC-?? / Please wait... Do not Remove Card`
→		`_20`	The POS sends a WIC Read Card Balance message.	`11.19WIC`

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td>_210000??
...</td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_21</td><td>Message identifier: Read WIC Balance response message.</td></tr><tr><td>0000</td><td>No error code.</td></tr><tr><td>??</td><td>WIC authority.</td></tr><tr><td>...</td><td>Current month balance data.</td></tr></table></td><td>11.19WIC</td></tr>
<tr><td></td><td></td><td></td><td>The POS prints an initial balance receipt.</td><td>11.19WIC</td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
→		`_30`	The POS sends a WIC "Debit WIC Balance" message. If _21 balance < _30 debit items, the POS may abort debit without sending the _30 message. This will generate the `0x0084 WSPI_INSUFFICIENT_BALANCE` error.	`11.19WIC`
			The terminal displays "Accept changes? / ENTER or CANCEL" with "YES / NO" buttons.	`11.19WICAccept changes? / ENTER or CANCEL`
			The cardholder accepts via "ENTER" or "YES" buttons.	`11.19WIC`
			The terminal briefly displays "Transaction accepted".	`11.19Transaction accepted`
			The terminal displays "Updating card".	`11.19Updating card`

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_310000??LLSIG...</code></td><td>
<table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td><code>_31</code></td><td>Message identifier: Debit WIC Balance response message.</td></tr>
<tr><td><code>0000</code></td><td>No error code.</td></tr>
<tr><td><code>??...</code></td><td>WIC authority.</td></tr>
<tr><td>LL</td><td>Debit signature length.</td></tr>
<tr><td>SIG...</td><td>Debit signature.</td></tr>
</table>
</td><td><code>11.19WIC</code></td></tr>
<tr><td>→</td><td></td><td><code>_20</code></td><td>The POS sends a WIC Read Card Balance message.</td><td><code>11.19WIC</code></td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_210000??...</code></td><td>
<table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td>"_21"</td><td>Message identifier: Read WIC Balance response message.</td></tr>
<tr><td>"0000"</td><td>No error code.</td></tr>
<tr><td>"??..."</td><td>WIC balance information.</td></tr>
</table>
</td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td></td><td></td><td>The POS prints an updated balance receipt.</td><td><code>11.19WIC</code></td></tr>
<tr><td>→</td><td></td><td><code>_50</code></td><td>The POS sends a WIC End Transaction message.</td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "Transaction complete".</td><td><code>11.19Transaction complete</code></td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="2">←</td><td>_510000??</td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_51</td><td>Message identifier: End WIC Transaction response message.</td></tr><tr><td>0000</td><td>No error code.</td></tr><tr><td>??...</td><td>WIC authority.</td></tr></table></td><td>11.19Transaction complete</td></tr>
<tr><td colspan="2">→</td><td>_80</td><td>The POS sends a WIC Remove Card message.</td><td>11.20Remove card</td></tr>
<tr><td colspan="2">←</td><td>_81</td><td>The terminals sends a WIC Remove Card response message.</td><td>11.20SMC</td></tr>
<tr><td colspan="2">→</td><td>10.</td><td>The POS resets the terminal's transaction data for the next transaction.</td><td>11.20SMC</td></tr>
</table>

Cancelled Debit Transaction

POS	Terminal	Message	Description	11.x Status Return Message
———	→	`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.01Slide Card`
			The terminal displays the online swipe/ insert screen.	`11.01Slide Card`
———	→	`_10`	The POS sends a WIC Get PAN message.	`11.19WIC`
			The terminal displays the "Please insert card" screen.	`11.19Please insert card`
			The cardholder inserts a WIC smart card. If a WIC smart card is inserted before _10 Get PAN Message, the terminal sends a '09.010201I' message and The terminal displays "WIC-?? / Please wait... Do not Remove Card".	'11.19WIC' is also returned if a WIC smart card is inserted before _10 Get PAN Message is sent and processed.

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="2">←</td><td><code>_110000??LLPAN...</code></td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td><code>_11</code></td><td>Message identifier: Get Card PAN response message.</td></tr><tr><td><code>0000</code></td><td>No error code.</td></tr><tr><td><code>??...</code></td><td>WIC authority.</td></tr><tr><td><code>LL</code></td><td>PAN length.</td></tr><tr><td><code>PAN...</code></td><td>PAN.</td></tr></table></td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td><code>11.19WIC-?? / Please wait... Do not Remove Card</code></td></tr>
<tr><td colspan="2">→</td><td><code>_60</code></td><td>The POS sends a WIC Authenticate cardholder message.</td><td><code>11.03Please enter your PIN</code></td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "Please enter your PIN".</td><td><code>11.03Please enter your PIN</code></td></tr>
<tr><td></td><td></td><td></td><td>The cardholder enters a valid PIN.</td><td><code>11.03Please enter your PIN</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal briefly displays "PIN OK".</td><td><code>11.19 WIC</code></td></tr>
<tr><td>←</td><td></td><td><code>_61000000</code></td><td><table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td>"_61"</td><td>Message identifier: Authenticate WIC User response message.</td></tr>
<tr><td>"0000"</td><td>No error code.</td></tr>
<tr><td>"00"</td><td>Valid PIN entered.</td></tr>
</table></td><td><code>11.19WIC</code></td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
			The terminal displays "WIC-?? / Please wait... Do not Remove Card".	`11.19WIC`
→		`_20`	The POS sends a WIC Read Card Balance message.	`11.19WIC`

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="2">←</td><td><code>_210000??
...</code></td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>"_21"</td><td>Message identifier: Read WIC Balance response message.</td></tr><tr><td>"0000"</td><td>No error code.</td></tr><tr><td>"??"</td><td>WIC authority.</td></tr><tr><td>"..."</td><td>Current month balance data.</td></tr></table></td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td></td><td></td><td>The POS prints an initial balance receipt.</td><td><code>11.19WIC</code></td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
→		_30	The POS sends a WIC "Debit WIC Balance" message. If _21 balance < _30 debit items, the POS may abort debit without sending the _30 message. This will generate the `0x0084 WSPI_INSUFFICIENT_BALANCE` error.	`11.19WIC`
			The terminal displays "Accept changes? / ENTER or CANCEL" with "YES / NO" buttons.	`11.19Accept changes? / ENTER or CANCEL`
			Cardholder cancels PIN entry via "CANCEL" button or by removing the WIC smart card.	`11.19WIC`
			The terminal displays "WIC update cancelled".	`11.19WIC`

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_31008C??00</code></td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>"_31"</td><td>Message identifier: Debit WIC Balance response message.</td></tr><tr><td>"008C"</td><td>WIC debit transaction cancelled</td></tr><tr><td>"??..."</td><td>WIC authority.</td></tr><tr><td>"00"</td><td>Zero-length debit signature.</td></tr></table></td><td><code>11.19WIC</code></td></tr>
<tr><td>→</td><td></td><td><code>_50</code></td><td>The POS sends a WIC End Transaction message.</td><td><code>11.19WIC</code></td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_510000??</code></td><td>"Transaction accepted" is not displayed since no debit occurred.<table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>"_51"</td><td>Message identifier: End WIC Transaction response message.</td></tr><tr><td>"0000"</td><td>No error code.</td></tr><tr><td>"??..."</td><td>WIC authority.</td></tr></table></td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td>→</td><td><code>_80</code></td><td>The POS sends a WIC Remove Card message.</td><td><code>11.20Remove card</code></td></tr>
<tr><td>←</td><td></td><td><code>_81</code></td><td>The terminals sends a WIC Remove Card response message.</td><td><code>11.20SMC</code></td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.20SMC`

10.5.3.6 *Transactions Canceled at PIN Entry*

A customer can alternatively cancel a transaction during PIN entry. Exceeding the number of allowed PIN attempts will also cancel a WIC transaction.

Invalid WIC PIN Entry

POS	Terminal	Message	Description	11.x Status Return Message
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.01Slide Card`
			The terminal displays the online swipe/ insert screen.	`11.01Slide Card`
→		`_10`	The POS sends a WIC Get PAN message.	`11.19WIC`
			The terminal displays the "Please insert card" screen.	`11.19Please insert card`

POS	Terminal	Message	Description	11.x Status Return Message
			The cardholder inserts a WIC smart card. If a WIC smart card is inserted before _10 Get PAN Message, the terminal sends a '09.010201I' message and The terminal displays "WIC-?? / Please wait... Do not Remove Card".	'11.19WIC' is also returned if a WIC smart card is inserted before _10 Get PAN Message is sent and processed.

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_110000??LLPAN...</code></td><td>
<table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td><code>_11</code></td><td>Message identifier: Get Card PAN response message.</td></tr>
<tr><td><code>0000</code></td><td>No error code.</td></tr>
<tr><td><code>??...</code></td><td>WIC authority.</td></tr>
<tr><td><code>LL</code></td><td>PAN length.</td></tr>
<tr><td><code>PAN...</code></td><td>PAN.</td></tr>
</table>
</td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td><code>11.19WIC-?? / Please wait... Do not Remove Card</code></td></tr>
<tr><td>→</td><td></td><td><code>_60</code></td><td>The POS sends a WIC Authenticate cardholder message.</td><td><code>11.03Please enter your PIN</code></td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
			The terminal displays "Please enter your PIN".	`11.03Please enter your PIN`
			The cardholder enters an invalid PIN.	`11.03Please enter your PIN`
			The terminal briefly displays "Incorrect PIN".	`'11.19Incorrect PIN'`
			The terminal displays "Please enter your PIN".	`11.03Please enter your PIN`
			The cardholder enters an invalid PIN.	`11.03Please enter your PIN`
			The terminal briefly displays "Incorrect PIN".	`'11.19Incorrect PIN'`
			The cardholder enters an invalid WIC PINs up to a maximum number of attempts.	

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td>_610090FF
or
_61000ALL</td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_61</td><td>Message identifier: Authenticate WIC User response message.</td></tr><tr><td>0090</td><td>All WIC PIN entry attempts invalid but card not PIN-blocked.</td></tr><tr><td>FF</td><td>No valid PIN entered.</td></tr></table>or</td><td>11.06Transaction cancelled</td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>"_61"</td><td>Message identifier: Authenticate WIC User response message.</td></tr><tr><td>"000A"</td><td>All WIC PIN entry attempts invalid and card PIN-blocked.</td></tr><tr><td>"LL"</td><td>No valid PIN entered; card locked.</td></tr></table></td><td></td></tr>
<tr><td>→</td><td></td><td>_50</td><td>The POS sends a WIC End Transaction message.</td><td>11.06Transaction cancelled</td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="2">←</td><td>_510000??</td><td>"Transaction accepted" is not displayed since no debit occurred.<table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_51</td><td>Message identifier: End WIC Transaction response message.</td></tr><tr><td>0000</td><td>No error code.</td></tr><tr><td>??...</td><td>WIC authority.</td></tr></table></td><td>11.19WIC</td></tr>
<tr><td colspan="2">→</td><td>_80</td><td>The POS sends a WIC Remove Card message.</td><td>11.20Remove card</td></tr>
<tr><td colspan="2">←</td><td>_81</td><td>The terminals sends a WIC Remove Card response message.</td><td>11.20SMC</td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.20SMC`

10.5.3.7 *Cancelled WIC PIN Entry*

POS	Terminal	Message	Description	11.x Status Return Message
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.01Slide Card`
			The terminal displays the online swipe/ insert screen.	`11.01Slide Card`
→		`_10`	The POS sends a WIC Get PAN message.	`11.19WIC`
			The terminal displays the "Please insert card" screen.	`11.19Please insert card`

POS	Terminal	Message	Description	11.x Status Return Message
			The cardholder inserts a WIC smart card. If a WIC smart card is inserted before _10 Get PAN Message, the terminal sends a '09.010201I' message and The terminal displays "WIC-?? / Please wait... Do not Remove Card".	'11.19WIC' is also returned if a WIC smart card is inserted before _10 Get PAN Message is sent and processed.

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td>←</td><td></td><td><code>_110000??LLPAN...</code></td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td><code>_11</code></td><td>Message identifier: Get Card PAN response message.</td></tr><tr><td><code>0000</code></td><td>No error code.</td></tr><tr><td><code>??...</code></td><td>WIC authority.</td></tr><tr><td><code>LL</code></td><td>PAN length.</td></tr><tr><td><code>PAN...</code></td><td>PAN.</td></tr></table></td><td><code>11.19WIC</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td><code>11.19WIC-?? / Please wait... Do not Remove Card</code></td></tr>
<tr><td>→</td><td></td><td><code>_60</code></td><td>The POS sends a WIC Authenticate cardholder message.</td><td><code>11.03Please enter your PIN</code></td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td></td><td></td><td></td><td>Cardholder cancels PIN entry via "CANCEL" button or by removing the WIC smart card.</td><td><code>11.03Please enter your PIN</code></td></tr>
<tr><td></td><td></td><td></td><td>The terminal displays "Transaction cancelled."</td><td><code>11.19Transaction cancelled</code></td></tr>
<tr><td>←</td><td></td><td><code>_61008EFF</code></td><td><table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td><code>_61</code></td><td>Message identifier: Authenticate WIC User response message.</td></tr>
<tr><td><code>008E</code></td><td>WIC PIN entry cancelled.</td></tr>
<tr><td><code>FF</code></td><td>No valid PIN entered.</td></tr>
</table></td><td><code>11.19WIC</code></td></tr>
</table>

<table>
<tr><th>POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="2">→</td><td><code>_50</code></td><td>The POS sends a WIC End Transaction message.</td><td><code>11.19WIC</code></td></tr>
<tr><td colspan="2">←</td><td><code>_510000??</code></td><td>"Transaction accepted" is not displayed since no debit occurred.
<table>
<tr><th>Message Fragment</th><th>Breakdown</th></tr>
<tr><td><code>_51</code></td><td>Message identifier: End WIC Transaction response message.</td></tr>
<tr><td><code>0000</code></td><td>No error code.</td></tr>
<tr><td><code>??...</code></td><td>WIC authority.</td></tr>
</table>
</td><td><code>11.19WIC</code></td></tr>
<tr><td colspan="2">→</td><td><code>_80</code></td><td>The POS sends a WIC Remove Card message.</td><td><code>11.20Remove card</code></td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
←		`_81`	The terminals sends a WIC Remove Card response message.	`11.20SMC`
→		`10.`	The POS resets the terminal's transaction data for the next transaction.	`11.20SMC`

10.5.4 eWIC Error Codes/Displays

10.5.4.1 WIC Error Codes

The following tables list the WIC WSPI error codes returned in WMP messages and the text displayed on the terminal, and are divided into two categories: ANSI error codes, and Toshiba's proprietary error codes.

ANSI Error Codes

Error Code	Error Code Name	Error Conditions	Terminal Displays
`0x0001`	`WSPI_ACCESS_DENIED`	WIC authorities are not initialized (i.e. no _00 message). Mapped to Actual Error Code: 0x9998 WSPI_NO_STATE_MODULE_FOR_CARD	"WIC problem/See journal message".
`0x0004`	`WSPI_BAD_PARAM`	Parameters in WMP request message are missing or invalid.	"WIC problem/See journal message".
`0x0005`	`WSPI_CARD_ABSENT`	WIC smart card is not (yet) inserted (see also `0x0006 WSPI_CARD_REMOVED` immediately below).	"WIC problem/See journal message".
`0x0006`	`WSPI_CARD_REMOVED`	WIC smart card was removed unexpectedly with WIC transaction in progress.	"Card removed/ Transaction cancelled".
`0x0007`	`WSPI_DELETE_ERROR`	WIC smart card erase data error.	"WIC problem/See journal message".

Error Code	Error Code Name	Error Conditions	Terminal Displays
0x0008	WSPI_INSUFFICIENT_BUFFER	Insufficient buffer was provided to return WMP response message.	"WIC problem/See journal message".
0x000A	WSPI_PIN_LOCKED	WIC smart card is blocked after allowed number of PIN entry attempts exceeded (see also 0x009D WSPI_PINALREADY_BLOCKED).	"Card problem/ Return card to clinic".
0x000B	WSPI_READ_ERROR	WIC smart card read data error.	"Card problem/ Return card to clinic".
0x000E	WSPI_UNKNOWN_ERROR	WIC error unknown or not handled.	"WIC problem/ See journal message".
0x0010	WSPI_UNKNOWN_CARD	Invalid, damaged, or unknown smart card.	"Invalid/Damaged card". or "Card problem/ Return card to clinic".
0x0011	WSPI_READER_UNAVAILABLE	Smart card reader error.	"WIC problem/ See journal message".
0x0012	WSPI_READER_BUSY	Smart card reader is unavailable, possibly busy with another smart card.	No change in display.
0x0080	WSPI_PURSE_ERROR	WIC smart card write balance error.	"Card problem/ Return card to clinic".
0x0081	WSPI_INVALID_CATEGORY	WIC item category is invalid.	"WIC problem/ See journal message".
0x0082	WSPI_INVALID_SUBCATEGORY	WIC item subcategory is invalid.	"WIC problem/ See journal message".
0x0083	WSPI_INVALID_UNIT	WIC item unit value is invalid.	"WIC problem/ See journal message".
0x0084	WSPI_INSUFFICIENT_BALANCE	No or insufficient WIC items are available to debit from WIC smart card's current month (see also 0x008F WSPI_EMPTY_PRESCRIPTION).	"No current WIC".

Error Code	Error Code Name	Error Conditions	Terminal Displays
0x0087	WSPI_BENEFITS_EXPIRED	WIC smart card's benefits are expired (i.e., current date is after all WIC smart card's benefits months).	"No current WIC".
0x0088	WSPI_BENEFITS_CONFLICT	WIC smart card's benefits months are invalid; (i.e., multiple benefits for same month(s), start/end dates out-of-order).	"Card problem/ Return card to clinic".
0x0089	WSPI_ALREADY_APPLIED	WIC debit (has likely) already applied.	No change in display.
0x008A	WSPI_PERIOD_NOT_ON_CARD	WIC smart card's benefits are not yet available; (i.e., current date is before all WIC smart card's benefits months).	"No current WIC".
0x008B	WSPI_CARD_REAUTHENTICATED	WIC smart card was re-authenticated at WIC clinic (after having been added to hot card list).	No change in display.
0x008C	WSPI_WICTRAN_CANCELLED	WIC transaction was cancelled by cardholder.	"Transaction cancelled". or "WIC update cancelled" (for _31 debit response).
0x008D	WSPI_NOPREV_READ	WIC smart card debit was attempted without prior balance read.	"WIC problem/ See journal message".
0x008E	WSPI_PINENTRY_CANCELLED	WIC PIN entry was cancelled by cardholder or timed out.	"Entry timeout/ Transaction cancelled".
0x008F	WSPI_EMPTY_PRESCRIPTION	No WIC items are available on WIC smart card (for current month) [see also 0x0084 WSPI_INSUFFICIENT_BALANCE].	"No current WIC".
0x0090	WSPI_INVALID_PIN	No valid PIN was entered by cardholder (but WIC card Not PIN-blocked; see 0x000A WSPI_PIN_LOCKED).	"Incorrect PIN".
0x0091	WSPI_INVALID_MSGLEN	WMP request message length is invalid.	"WIC problem/ See journal message".
0x0093	WSPI_CARDTYPE_ERROR	WIC smart card type is not enabled for current mode (configured in _00 message, see eWIC WMP Messages).	"Authentication failed".

Error Code	Error Code Name	Error Conditions	Terminal Displays
0x0095	WSPI_CSNREAD_ERROR	WIC smart card read serial number error.	"Card problem/ Return card to clinic".
0x0097	WSPI_WICADMINREAD_ERROR	WIC smart card data error.	"Card problem/ Return card to clinic".
0x0098	WSPI_CRYPTOAUTH_ERROR	WIC smart card cryptographic authentication error.	"Card problem/ Return card to clinic".
0x0099	WSPI_BIN_ERROR	WIC authority for current card/operation is not available or loaded. Mapped to Actual Error Code: 0x9998 WSPI_NO_STATE_MODULE_FOR_CARD	"WIC problem/ See journal message".
0x009A	WSPI_INVALIDPIN_LOCK	WIC smart card is (possibly) not blocked even after exceeding number of allowed PIN entry attempts (see also 0x000A WSPI_PIN_LOCKED).	"Card problem/ Return card to clinic".
0x009B	WSPI_WICAUTHORITY_ERROR	Missing WIC authority configuration file (WIC.INI); WIC.INI's KTK doesn't match injected KTK.	"WIC problem/ See journal message".
0x009C	WSPI_GROCER_BLOCKED	WIC smart card is (already) blocked by terminal.	"Card problem/ Return card to clinic".
0x009D	WSPI_PINALREADY_BLOCKED	WIC smart card is already PIN-blocked (see also 0x000A WSPI_PIN_LOCKED).	"Card problem/ Return card to clinic".
0x009E	WSPI_FUTURE_LOCKCARDDATE	Date to block WIC smart card is after current date.	No change in display.

Toshiba-Proprietary Error Codes

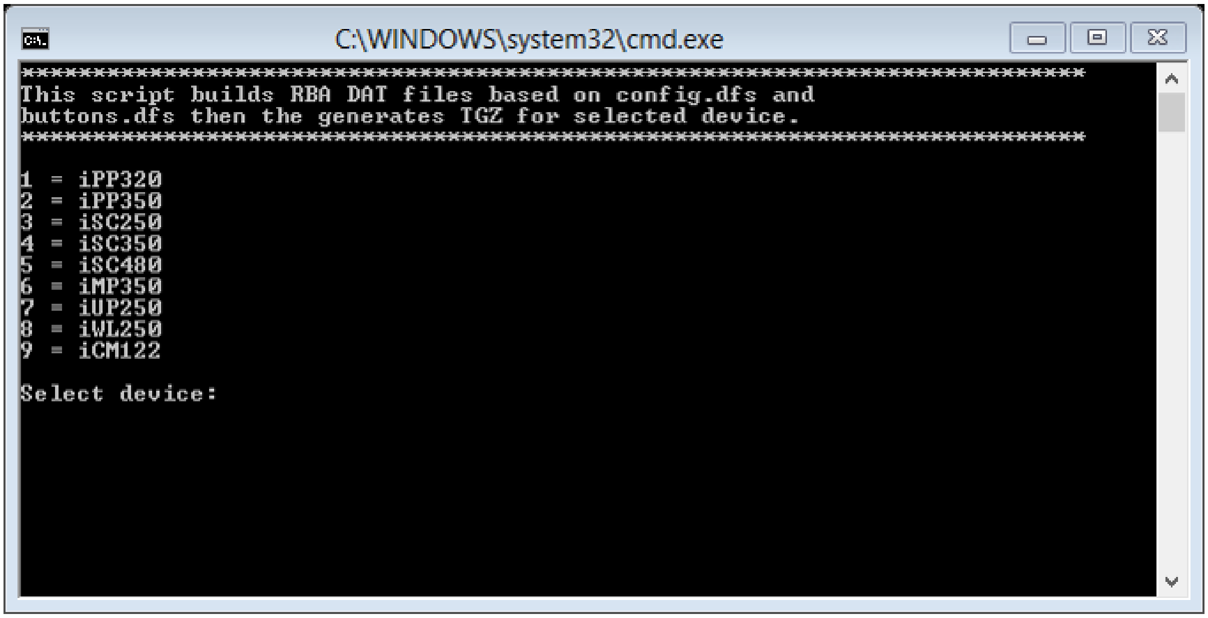
Error Code	Error Code Name	Error Conditions	Terminal Displays
0x9993	WSPI_TENDER_MISMATCH	Cardholder selected another tender on terminal (not WIC).	No change in display.

Error Code	Error Code Name	Error Conditions	Terminal Displays
0x9995	WSPI_NO_ACTIVE_STATE	WIC disabled (i.e. RBA configuration parameter '0020_0001' = '0').	No change in display.
0x9997	WSPI_STATE_MODULE_MISSING	No WIC authority DLLs are available or loaded.	"WIC problem/ See journal message".
0x9998	WSPI_NO_STATE_MODULE_FOR_CARD	No WIC authority DLLs are initialized (via _00 message) [see also 0x0001 WSPI_ACCESS_DENIED, WSPI_BIN_ERROR].	"WIC problem/ See journal message".

10.6 Appendix F: Creating a .TGZ File

This section describes the procedure for creating a .TGZ file. Once the .DFS file is translated to the internal .DAT format for the terminal, the .TGZ file can be generated for downloading to the terminal. Only translated files can be downloaded.

1. Open the **IK-RBA-16.10** folder.
2. Select the **Telium RBA Parameters** folder.
3. Select the **RBA Data and Parameter**s folder.
4. Select the **config** folder.
5. Select the **DFS_SRC** folder.
6. Select the **config.dfs** file and open using NotePad.
7. Edit the file and incorporate the necessary configuration setting changes, then save the file.
8. Return to the **RBA Data and Parameters** folder.
9. Select the **GEN_TGZ** file. Refer to the following screen capture of this file. The script builds RBA .DAT files from Data File System (.DFS) files and generates the .TGZ file for the terminal.



```
**********************************************************************
This script builds RBA DAT files based on config.dfs and
buttons.dfs then the generates TGZ for selected device.
**********************************************************************

1 = iPP320
2 = iPP350
3 = iSC250
4 = iSC350
5 = iSC480
6 = iMP350
7 = iUP250
8 = iWL250
9 = iCM122

Select device:
```

Selecting a Terminal via Command Prompt

10. Select the appropriate terminal from the numeric list (e.g., '5' for iSC480) and select <Enter>.
11. The **Command** window should state "Success."
12. Select the **Package** folder.
13. Select the terminal folder (e.g., iSC480).
14. Open the **RBA Testing Tool**.
15. In the RBA Testing Tool, select to the **Group** menu, which displays all message options.
16. Double-click 62.x **File Write Request** message.
17. Select the **Load 8 Bits** option.
18. Select the appropriate **.TGZ** file (e.g., `iSC480.TGZ`).
19. Select the **Download to Host** option.
20. Select **Execute**.
21. The file download to the terminal is now complete.

10.7 Appendix G: Forms

The RBA uses form files to position text, buttons, and graphics on the screen, and to select colors or font types.

10.7.1 Limitations

The form file name can be up to 12 characters long, including the dot (.) and K3Z extension. 150 total forms can be loaded to the terminal, including the default forms.

10.7.2 Buttons

The .K3Z files provided with the off-the-shelf RBA include the most common buttons that can be used in the RBA. In the button graphic's file name, some .bmp files contain the letter “d” or “u.”

- “d” indicates the down-button position (resembling a pushed-in button)
- “u” indicates up-button position (resembling a button that has not been pushed).

Only the control buttons listed on the process's form are used (CANCEL, CLEAR, ENTER, +, -). All other buttons are ignored.

10.7.3 Languages

The RBA has the ability to display prompts in up to three languages. The prompts are stored in the files `PROMPT.XML` and `SECURPROMPT.XML`. In order to support multiple languages, each prompt is assigned a number. A form can then reference a prompt by its number. For example, the text element in the form `swipe.K3Z` contains the text “&lt;?ivPROMPT3?&gt;”. This instructs the RBA to load prompt three from the current language's prompt file. Prompt three should, in the proper language, instruct the customer to swipe a card.

10.7.4 Form Variables

There is some variable information that may need to be displayed. The following table provides a list of special text that will insert specific information in a prompt. These may be nested. For example, the form “amtv.K3Z” contains a text element that contains the text “&lt;?ivPROMPT10?&gt;”. Prompt 10 is “Amount OK? $&lt;?ivTOTAL?&gt;”. If the transaction total is $123.45, the prompt will display as “Amount OK? $123.45”.

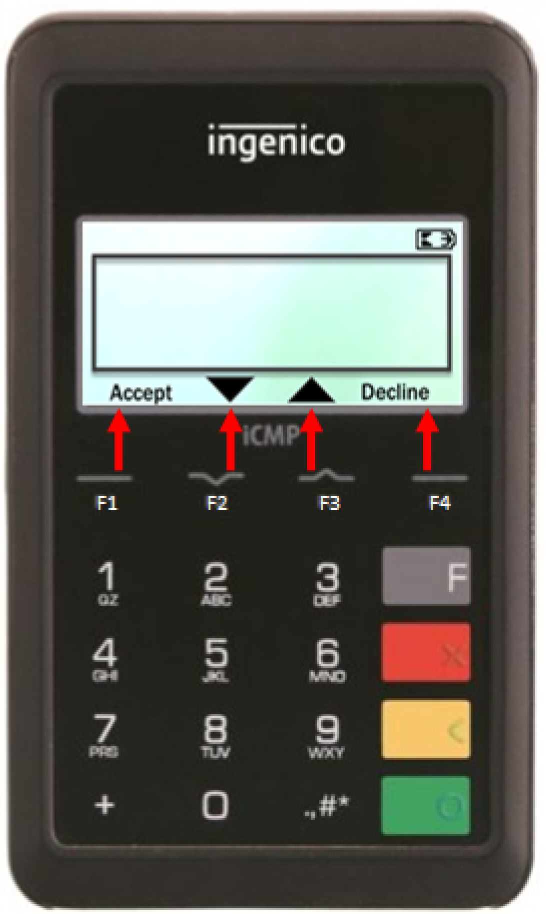
Special Text Display

Special Text	Description
&lt;?ivPROMPT99?&gt;	Insert a specific prompt from the prompt file.
&lt;?ivCARDNUMBER?&gt;	Displays account number with all but last X digits as Xs. The number of digits shown is set by config setting '0003_0011'.
&lt;?ivAMOUNT?&gt;	Purchase amount.
&lt;?ivCASHBACK?&gt;	Cash back amount selected.
&lt;?ivCB_MAX?&gt;	Maximum cash back allowed.
&lt;?ivCB_INC?&gt;	Cash back increment amount.
&lt;?ivCB1?&gt;	Amount for Fast Cash Back Key 1.
&lt;?ivCB2?&gt;	Amount for Fast Cash Back Key.
&lt;?ivCB3?&gt;	Amount for Fast Cash Back Key 3.
&lt;?ivCB4?&gt;	Amount for Fast Cash Back Key 4.
&lt;?ivEXPDATE?&gt;	Expiration Date from card.

Special Text	Description
&lt;?ivCARDHOLDERNAME?&gt;	Cardholder's name (taken from Track 2 of payment card).
&lt;?ivTOTAL?&gt;	Purchase amount + cash back amount.
&lt;?ivVAR1?&gt; through &lt;?ivVAR25?&gt;	User-defined variable (1-25). Set through the file `var.dat` or the 28.x message.
&lt;?ivTYPE?&gt;	Card type defined in `cards.dat`.

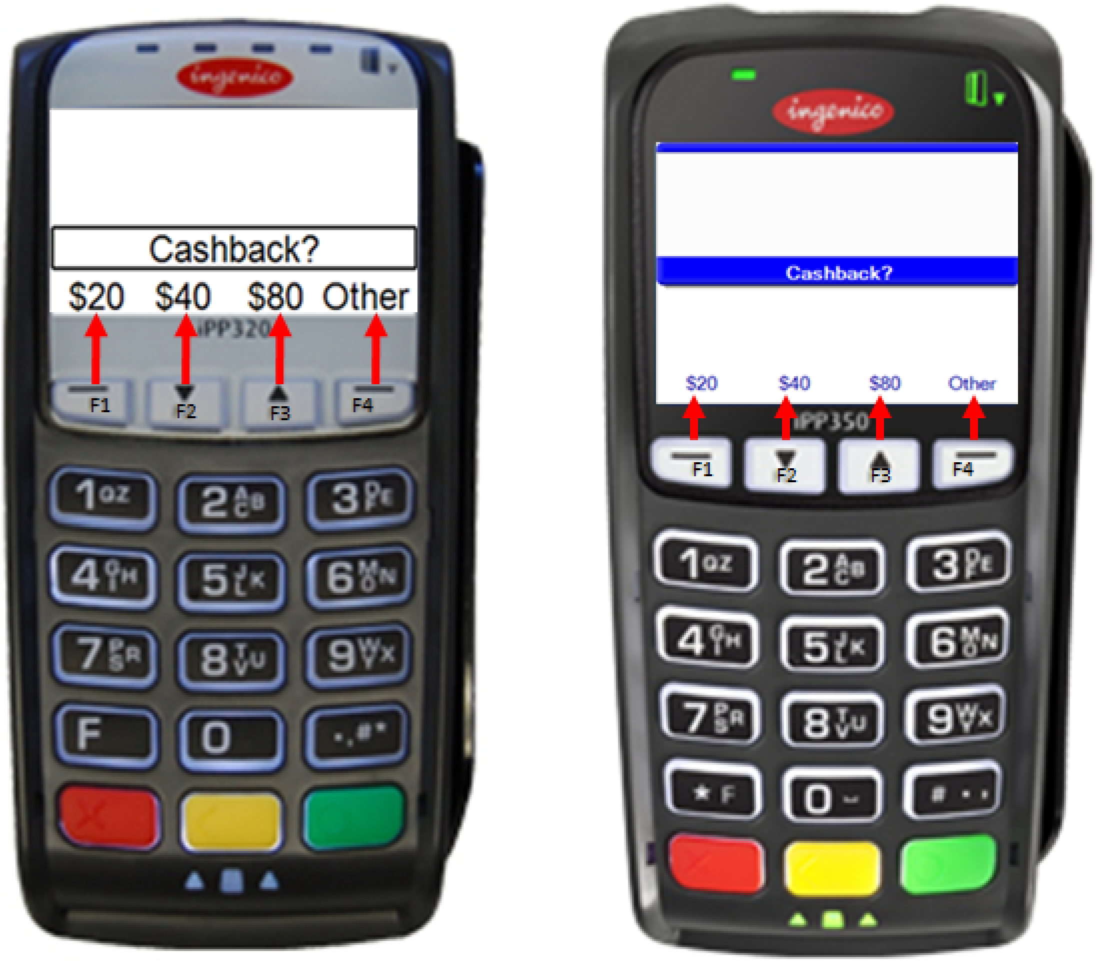
10.7.5 Using the Function Keys to Select Menu Options

For Ingenico's non-touch screen terminals, the four Function keys (F1 - F4) are used to select from menu options displayed on forms. The Function keys are aligned with the menu options, just below the screen. When a form which includes scrollable text is displayed, the two center Function keys (<F2> and <F3>) are used to scroll down or scroll up. The <F2> key is used for scrolling down while the <F3> key is used for scrolling up. In the following example, the <F2> and <F3 keys are used for scrolling while the <F1> and <F4> keys are used for selecting "Accept" or Decline" in the Terms and Conditions form.



Example Use of Function Keys for Scrolling and Selecting Menu Options

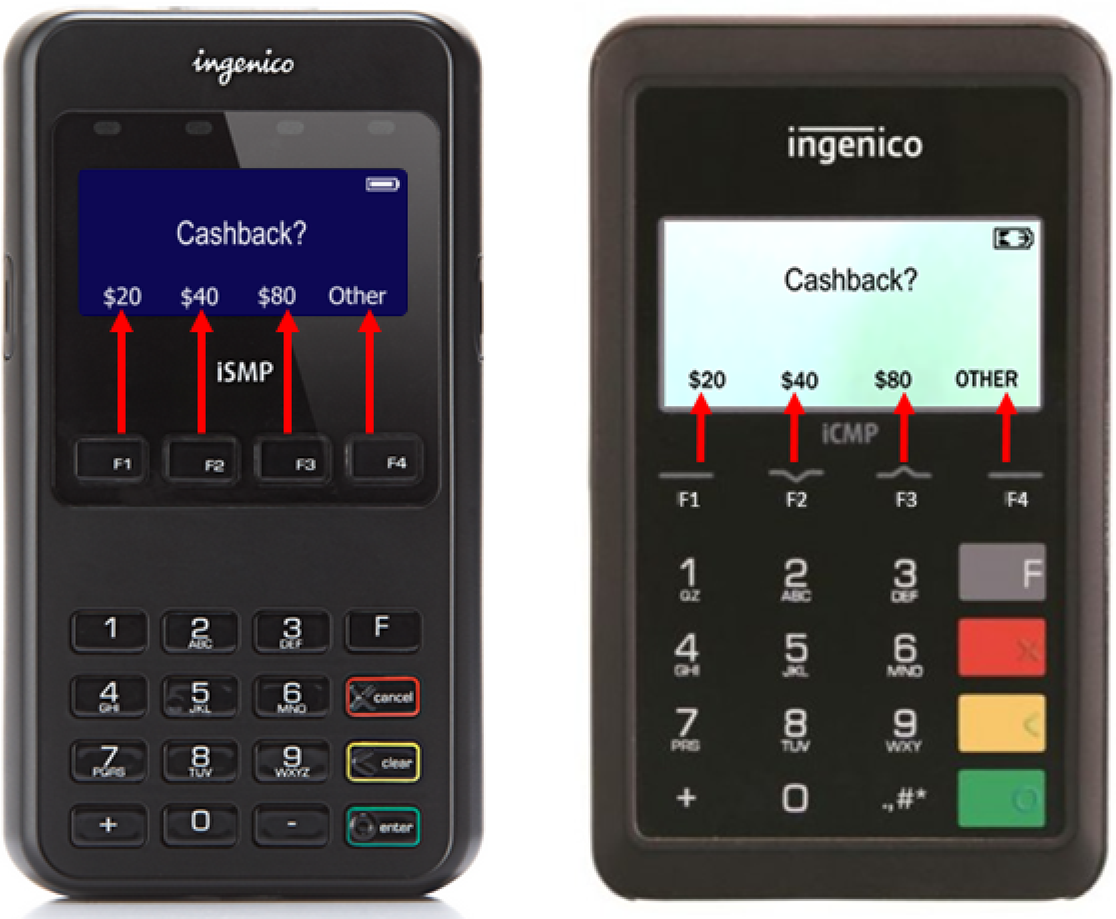
The following illustrations show Function Key usage for selecting menu options on iPP320, iPP350, iSMP, iCMP, and iWL series payment terminals.



iPP320

iPP350

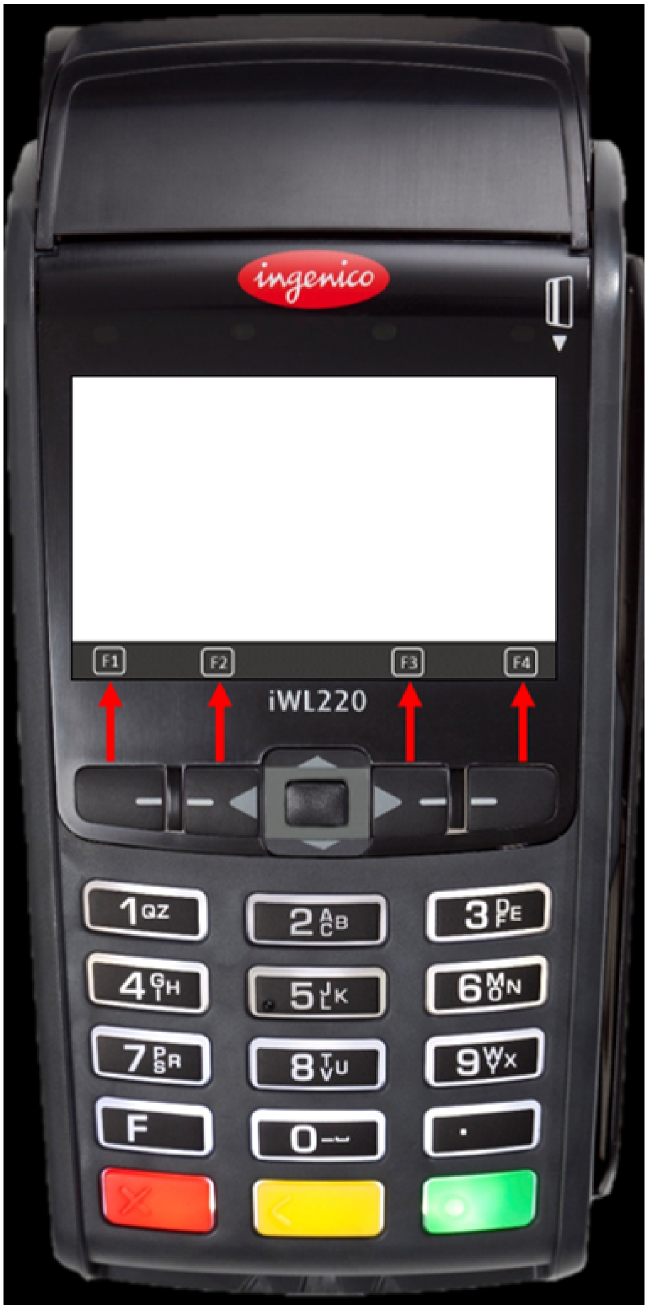
iPP320 and iPP350 Menu Option Selection



iSMP

iCMP

iSMP and iCMP Menu Option Selection



iWL Menu Option Selection

The use of Function keys to select menu options as described in this section does not apply to the iSC250, iSC350, or iSC480 touch screen terminals.

10.7.6 Using iWL Arrow Keys to Select Menu Options

The following form fields allow the up, down, and center keys to select items on a form such as menu.k3z:

- downenabled
- Down-Link
- upenabled
- Up-Link
- centerenabled
- Center-Link

The format is the same as the corresponding fields for the F1, F2, F3, and F4 buttons. The Form Builder does not yet support these fields, but they will function if added manually.

10.7.7 Form Contents and Descriptions

This section includes form images for each terminal and a table describing the parameters, prompts, and buttons used with each form. Additional form information is available in the following sections:

- Form Files (forms.dat) - A list all forms, form file names, and a brief descriptions. All entries in this table are also linked to the actual form description pages.
- Button IDs and Images - A list of buttons, button IDs, and button images for Telium terminals.
- Status Messages (status.dat) - A list of status messages.

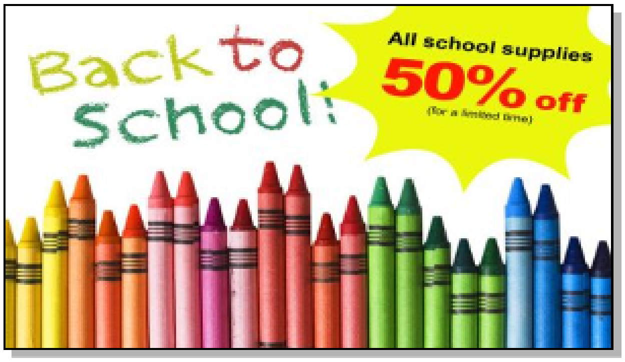
> Each of the PayPal versions noted in this section includes a PayPal button in place of the Enter Card button.

10.7.7.1 Advertising

iPP320

FRESH
FROM THE
OVEN

iPP350/iWL250



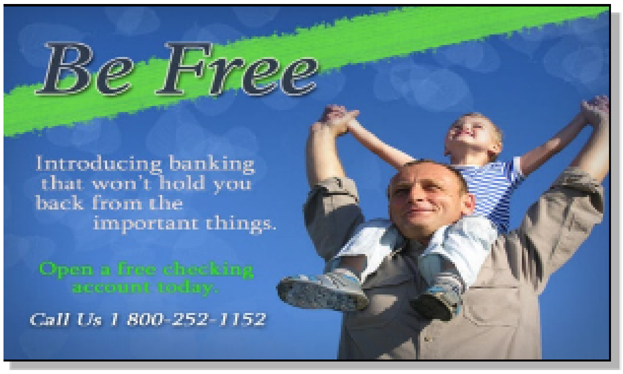
iSC250

iSC350

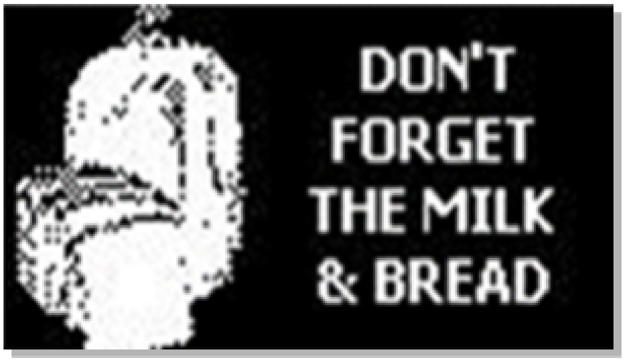
Citrus
SALE!
All citrus produce on sale now through July 4th!

Cherry Tomatoes
Now only $1.99 (PER PINT)

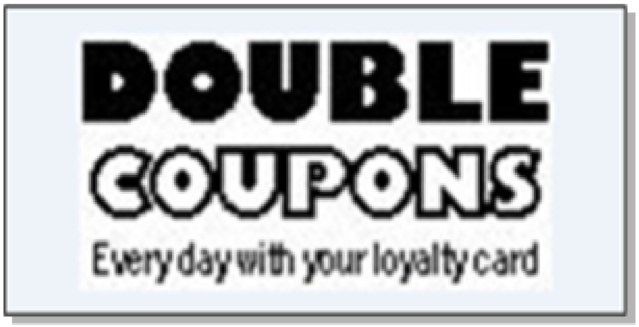
iSC480


Be Free
Introducing banking that won't hold you back from the important things.
Open a free checking account today.
Call Us 1 800-252-1152

iSMP



iCMP

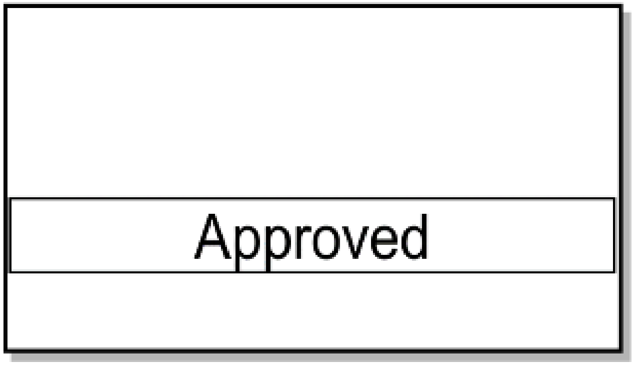


Description	Advertisement images
Initiated By	30.x message
Status Response	11.06
DFS Data Index	0030_0019
Form	ADS.K3Z
Text	N/A
Form Buttons and IDs	N/A

Terminal Buttons Allowed	N/A

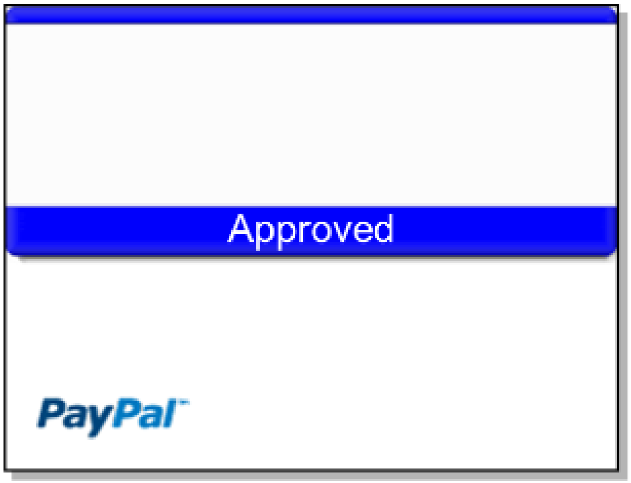
10.7.7.2 Approved/Disapproved

iPP320



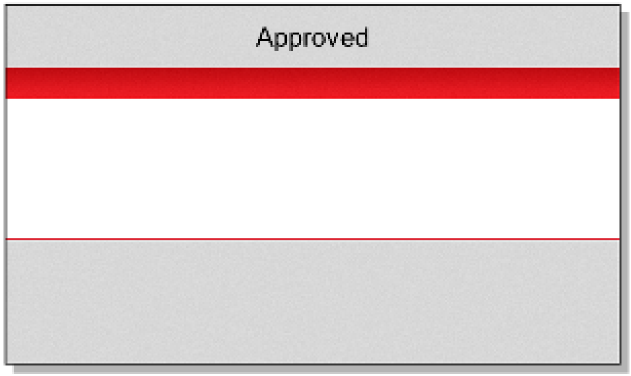
Approved

iPP350 & iWL250



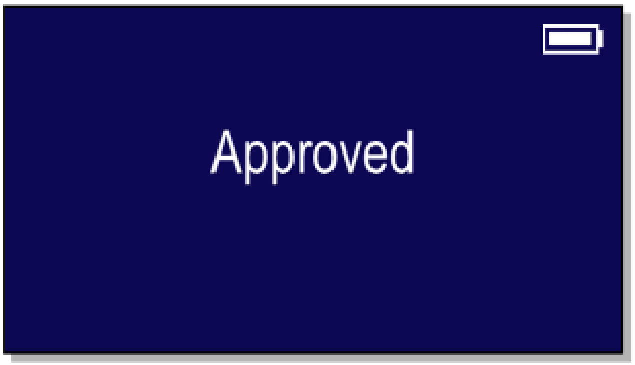
Approved

iSC250 & iSC350 & iSC480



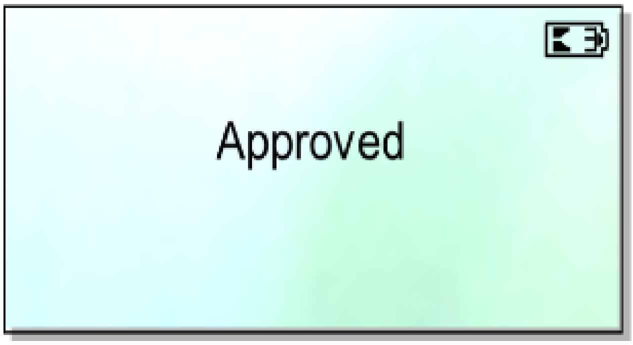
Approved

iSMP & iSMPc



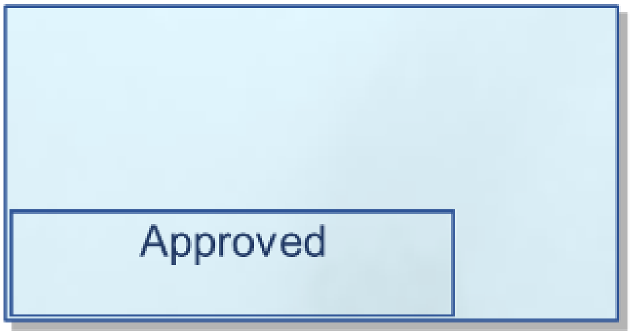
Approved

iCMP



Approved

iUP250



Approved

Description	Transaction card approval / disapproval.
Initiated By	01.x Online message and language disabled
Status Response	11.06 Approved/Declined.
DFS Data Index	0030_0024
Form	APPDAPP.K3Z
Text	"Approved" or "Declined."
Form Buttons and IDs	
Terminal Buttons Allowed	

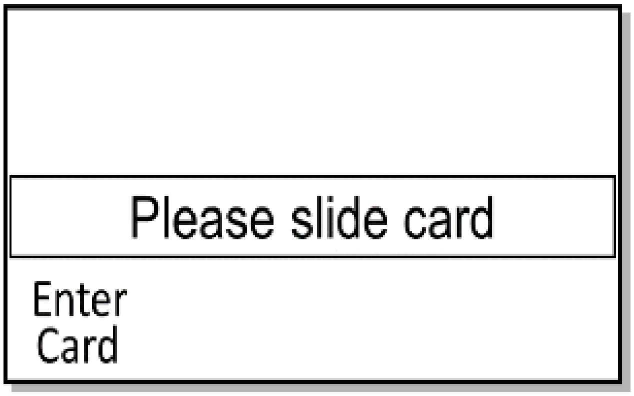
10.7.7.3 Card Swipe Forms

This section includes the following forms:

- Card Swipe
- Card Swipe with Language Selection
- Card Swipe On Demand
- Remove Inserted Card

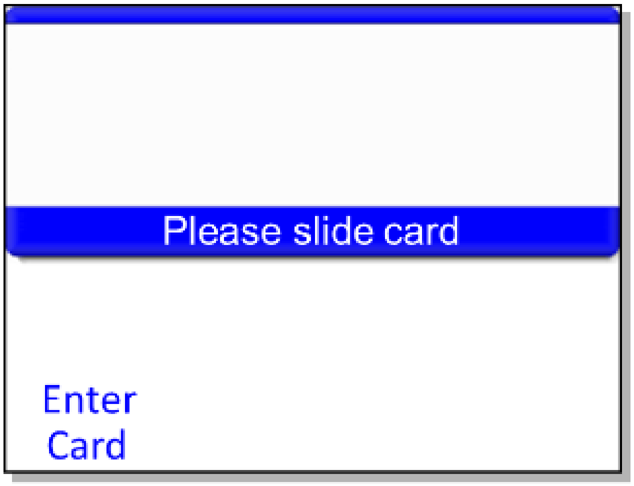
10.7.7.3.1 Card Swipe

iPP320



Please slide card

iPP350 & iWL250



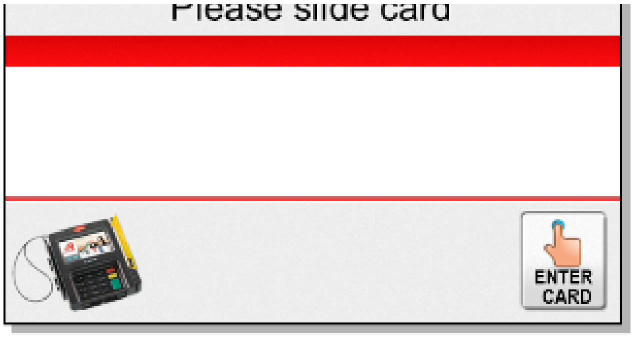
Please slide card

iSC250

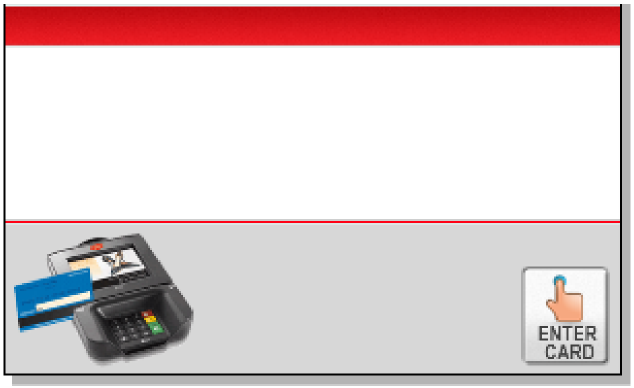
Please slide card

iSC350

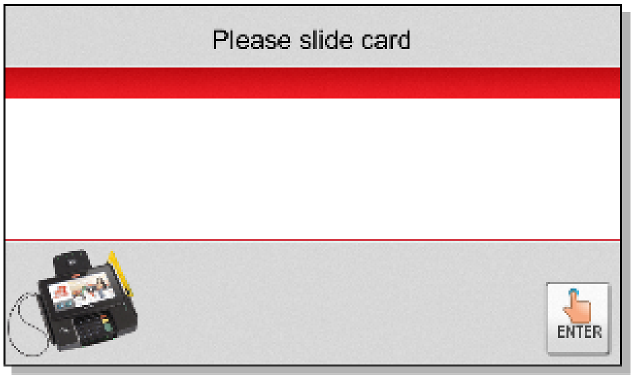
Please slide card


Please slide card

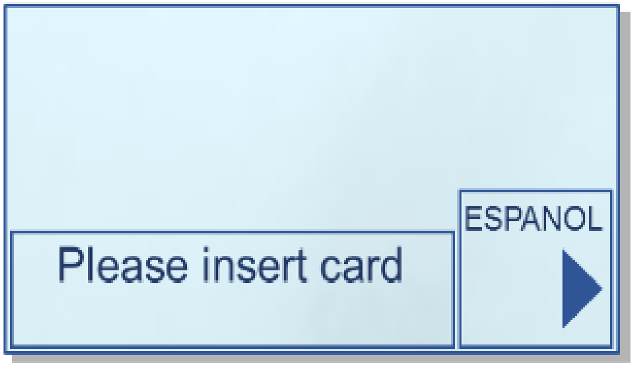
iSC480



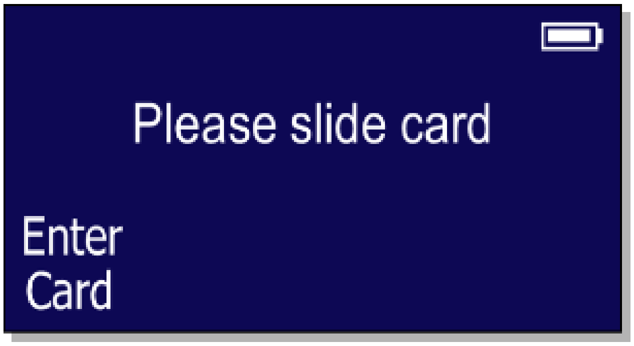
iUP250

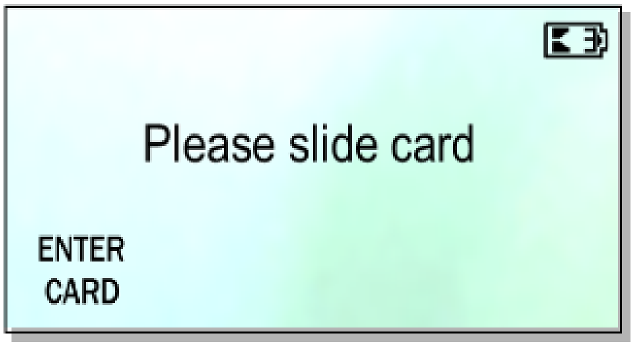

Please slide card

iSMP & iSMPc


Please insert card

iCMP


Please slide card


Please slide card

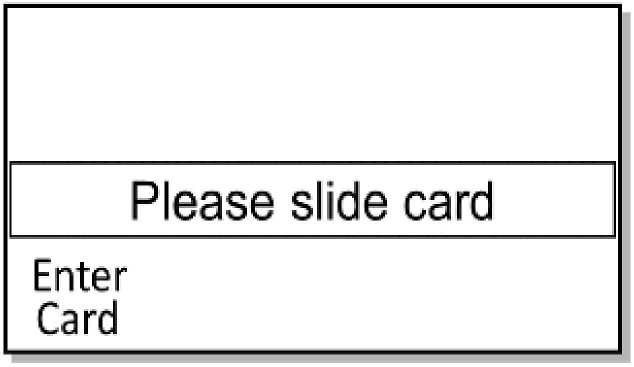
Description	Card swipe.
Initiated By	01.x Online message and language disabled
Status Response	11.01SlideCard
DFS Data Index	0030_0004
Form	SWIPE.K3Z

Text	“Please slide card”
Form Buttons and IDs	“Enter Card” – M
Terminal Buttons Allowed	Cancel

> The PayPal version of this form is `PSWIPE.K3Z`. Note that the PayPal form is not available for iMP350, iMP352, iPP320 and iPP350 terminals.

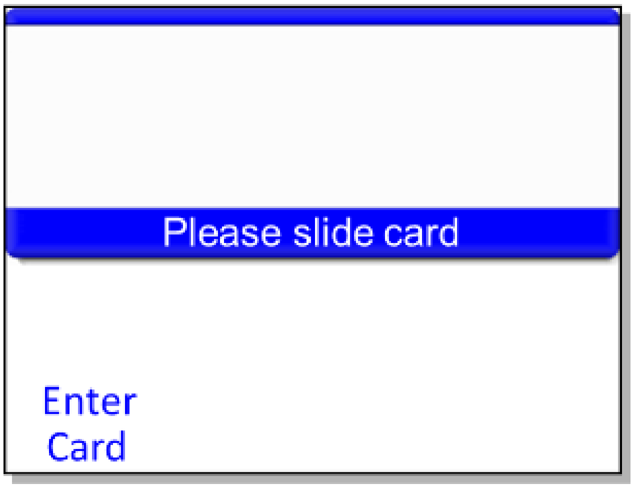
10.7.7.3.2 Card Swipe On Demand

iPP320



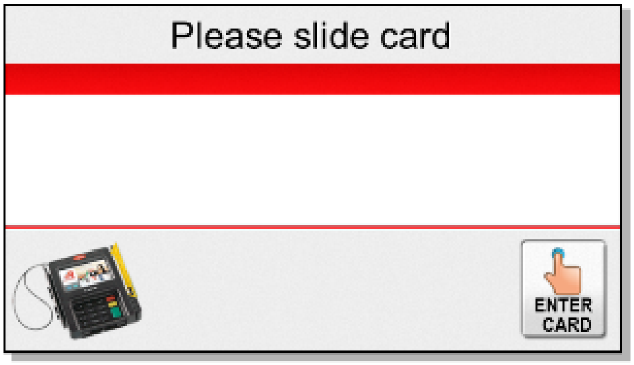
Please slide card

iPP350 & iWL250



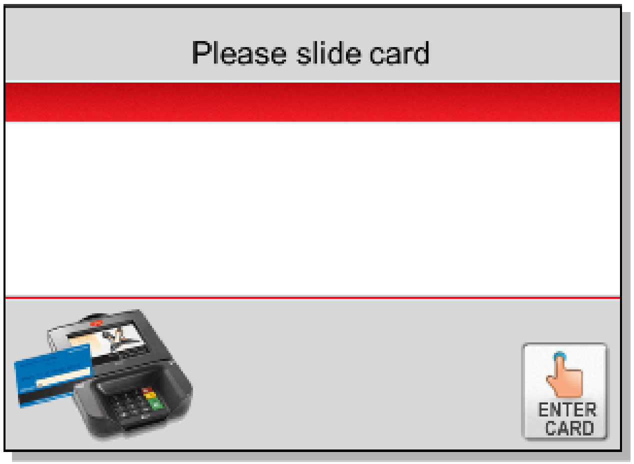
Please slide card

iSC250



Please slide card

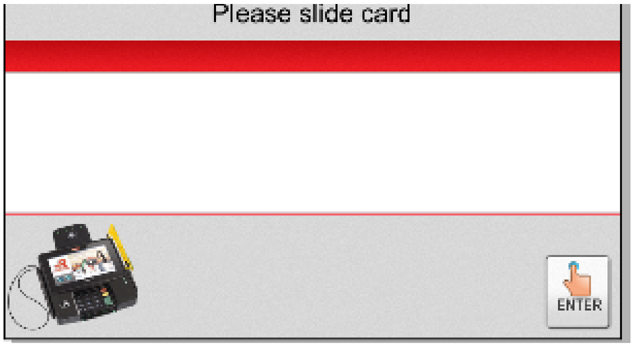
iSC350



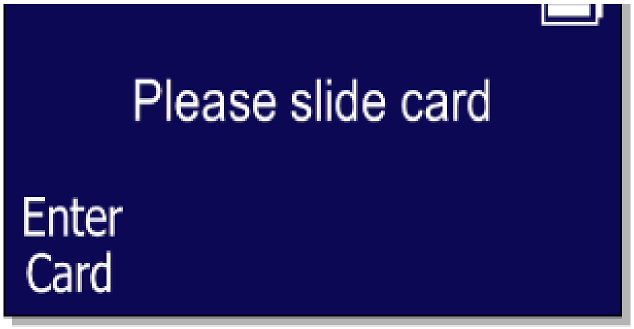
Please slide card

iSC480

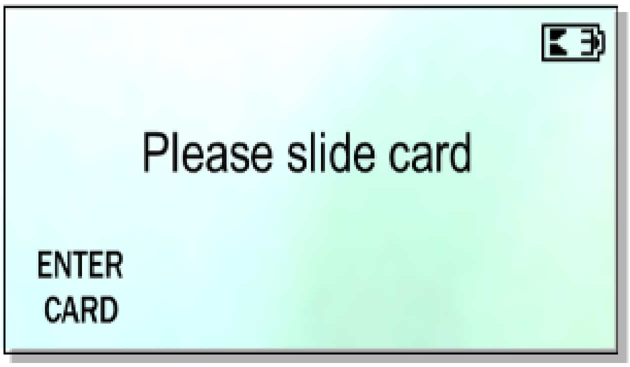
iSMP & iSMPc

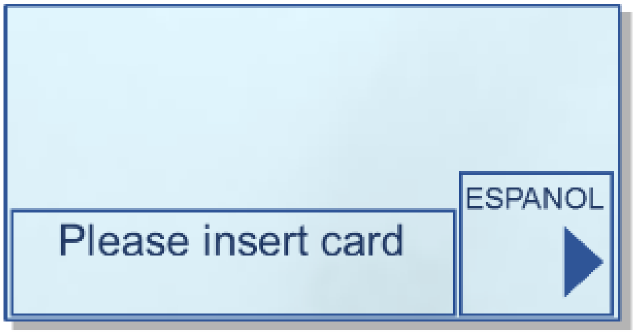

Please slide card

iCMP


Please slide card

iUP250


Please slide card

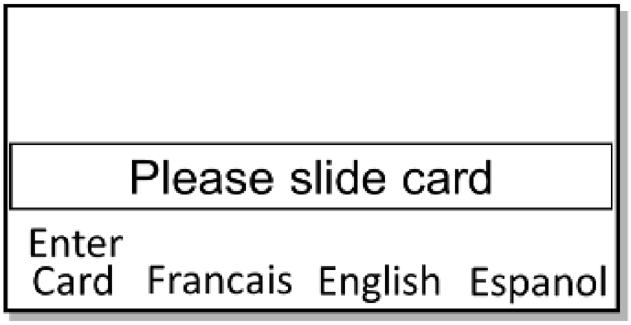

Please insert card

Description	Card read request (prompts user to swipe magnetic stripe card)
Initiated By	23.x Card Read Request message
Status Response upon display	11.12Input
Status Response after swipe	11.13Input Accepted
DFS Data Index	0030_0014
Form	COD.K3Z
Text	Variable
Form Buttons and IDs	iSC250/iSC350/iSC480 • "ENTER CARD"
Terminal Buttons Allowed	Cancel

The "Enter Card" button for manual entry is hidden when the Display “Enter Card” Prompt (configuration parameter '0007_0029') is set to '0'. In order for the manual entry button to be displayed, this configuration parameter must be set to a value of '1' to '4' as described in the Main Flow (mainFlow.dat) section of this document.

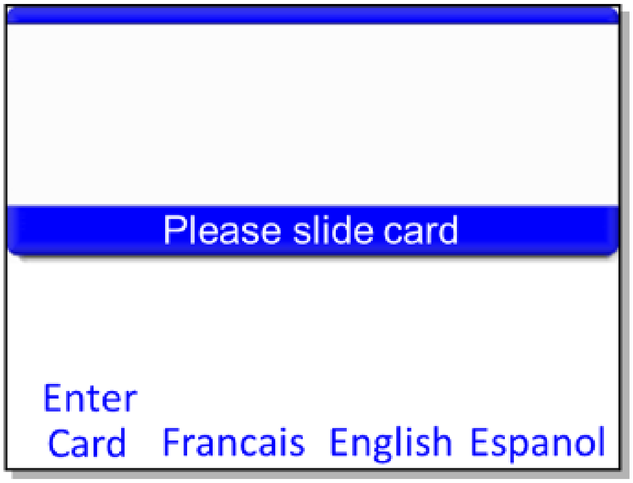
10.7.7.3.3 Card Swipe with Language Selection

iPP320



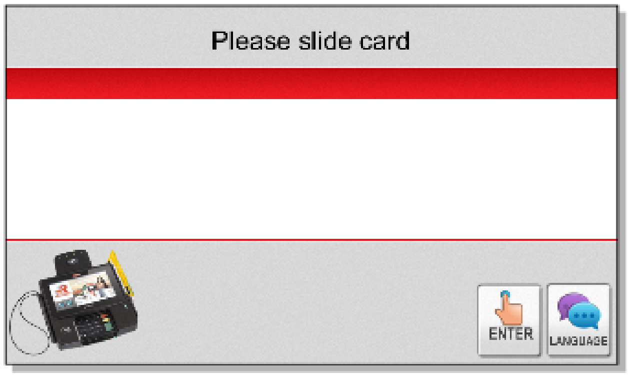
Please slide card

iPP350 & iWL250



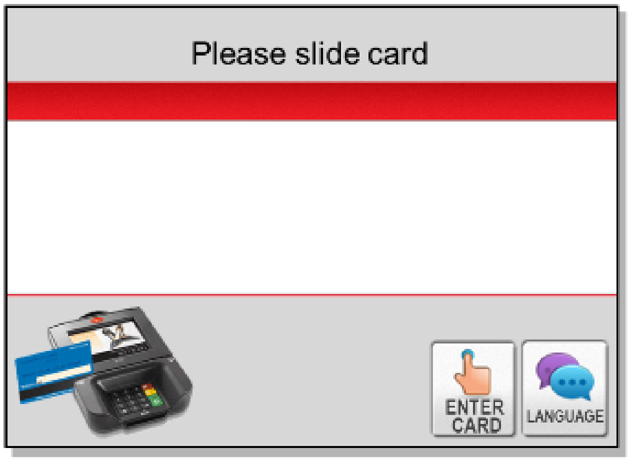
Please slide card

iSC250



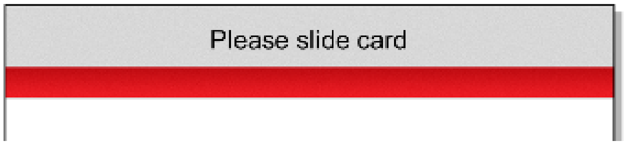
Please slide card

iSC350



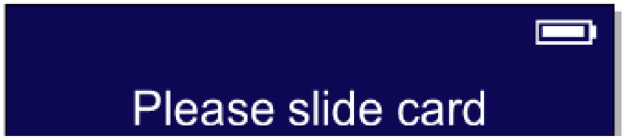
Please slide card

iPSC480

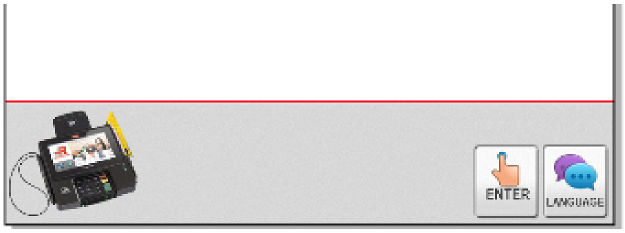


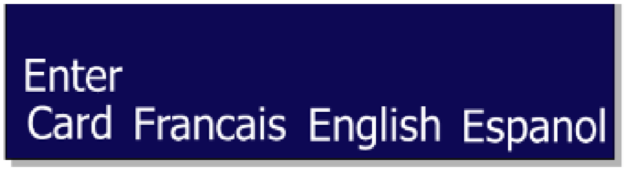
Please slide card

iSMP & iSMPc

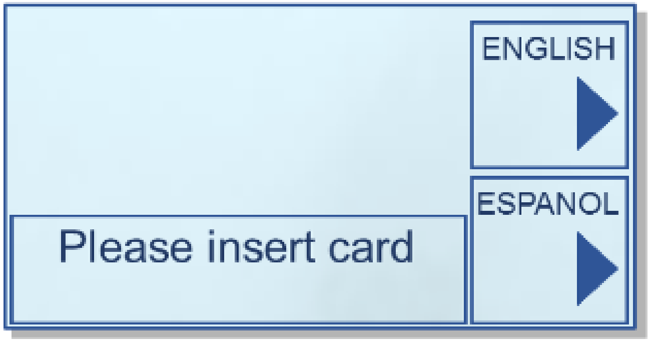


Please slide card



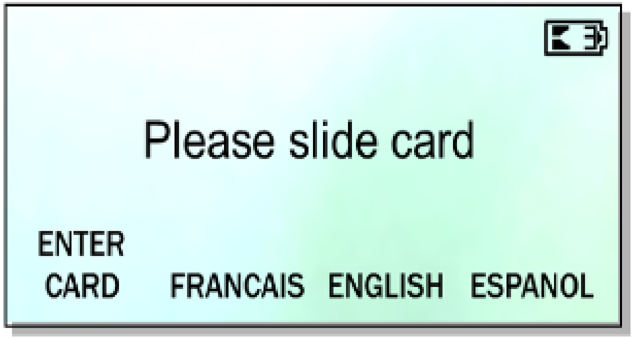

Enter
Card Francais English Espanol

iUP250



Please insert card

iCMP


Please slide card

Description	Card swipe with language. Settings are as follows: **Parameter** | **Setting** | **Description** 0007_0004 | 1 | Combine language and swipe screen. 0007_0005 | 0 | Swipe card first, then select payment type.
Initiated By	01.x Online message
Status Response	11.01SlideCard
DFS Data Index	0030_0005
Form	LSWIPE.K3Z
Text	“Please slide card”

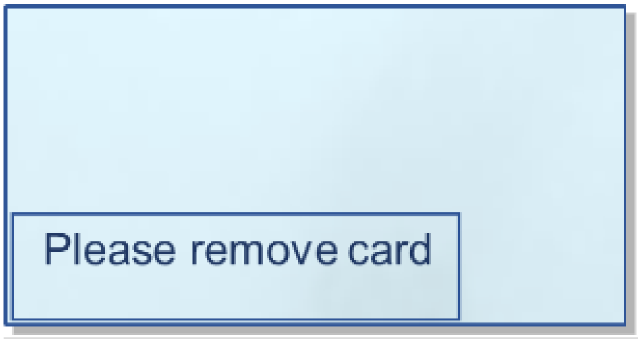
Form Buttons and IDs	iPP320/iPP350/iWL250/iMP350 • “Enter Card” – M • “English” – 1 • “Francais” – 3 • “Español” – 2 iSC250/iSC350/iSC480 • "ENTER CARD" - M • "LANGUAGE" iUP250 • "ENGLISH" - 1 • "ESPANOL" - 2 Language selection for the iSC250/iSC350/iSC480 is selected by pressing the LANGUAGE button which takes the user to another screen to select the language.
Terminal Buttons Allowed	Cancel

> The PayPal version of this form is `PLSWIPE.K3Z`. Note that the PayPal form is not available for iPP320 and iPP350 terminals.

10.7.7.3.4 Remove Inserted Card

This form is specific to the iUP250 terminal.

iUP250



Please remove card

Description	Prompts the customer to remove the card once it has been inserted.

Initiated By	Card is inserted to the slot (is essentially the second swipe card screen).
Status Response	11.01 Slide Card
DFS Data Index	0030_0030
Form	REMOVE.K3Z
Text	“Please remove card quickly”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

10.7.7.4 Cash Back Forms

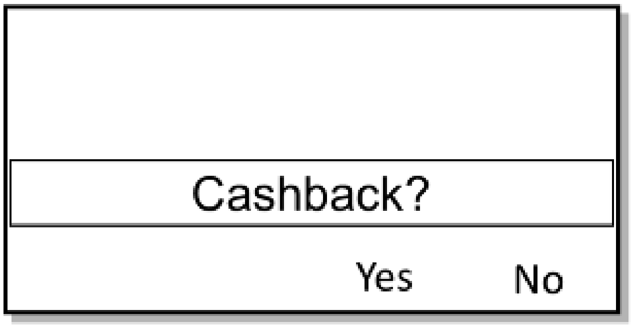
This section includes the following cash back forms:

- Cash Back Other
- Cash Back Selection
- Cash Back Selection without No
- Cash Back Verification

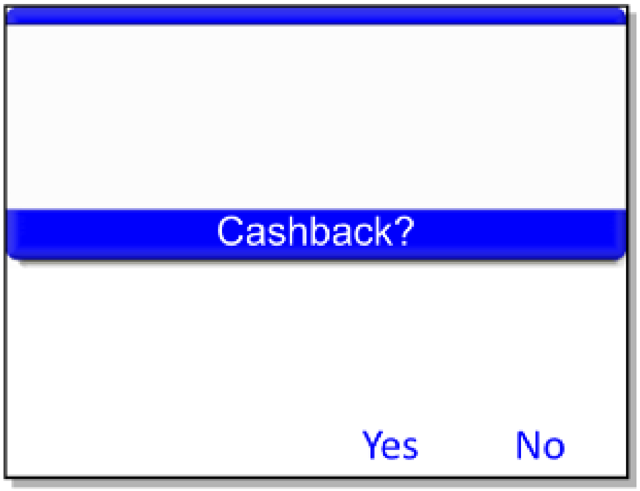
10.7.7.4.1 Cash Back Selection

The Cash Back Selection screen prompts the cardholder to select cashback during a transaction. Some terminals provide simple "Yes" and "No" options, and then display another screen to prompt the cardholder for the cash back amount if the selected option is "Yes". The iSC250, iSC350 and iSC480 provide cashback amount selection options on the current screen. It is important to note that selecting the virtual "No" button on the display for all terminals simply selects no cash back. Pressing the physical "Cancel" key, however, cancels the current payment on all terminals. Refer to the following form illustrations for the Cash Back Selection screen.

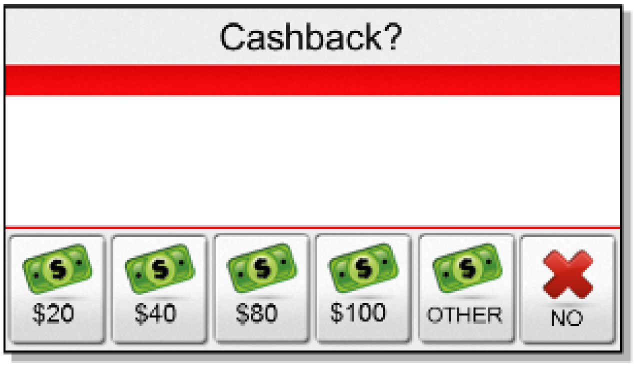
iPP320



iPP350 & iWL250

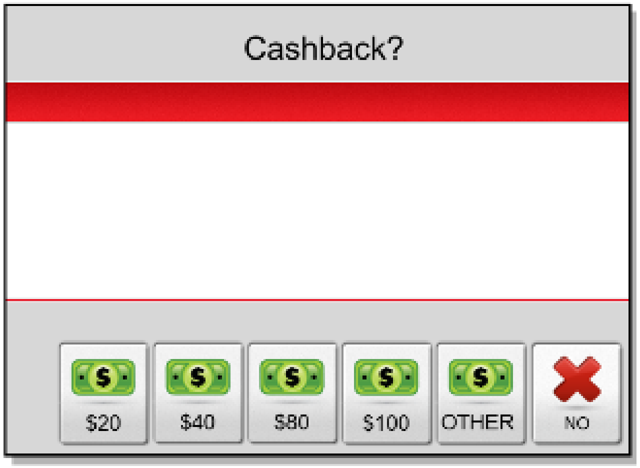


iSC250 & iSC480



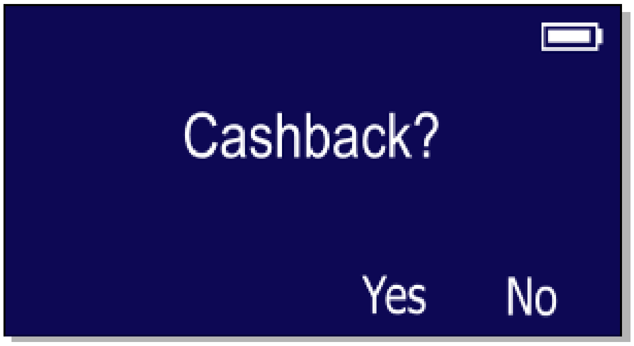
Cashback?

iSC350



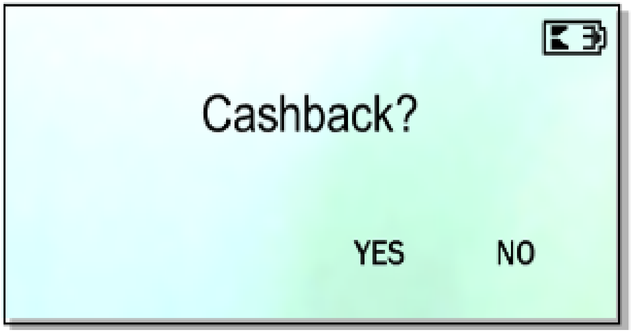
Cashback?

iSMP & iSMPc



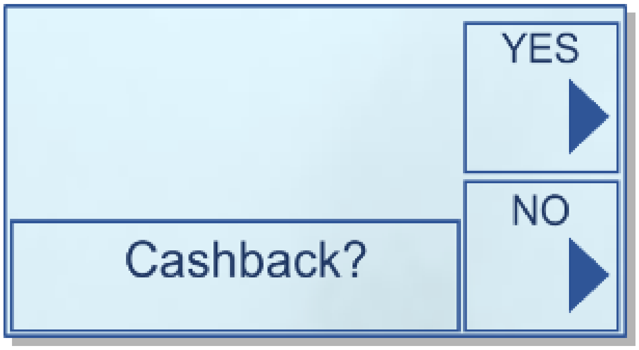
Cashback?

iCMP



Cashback?

iUP250

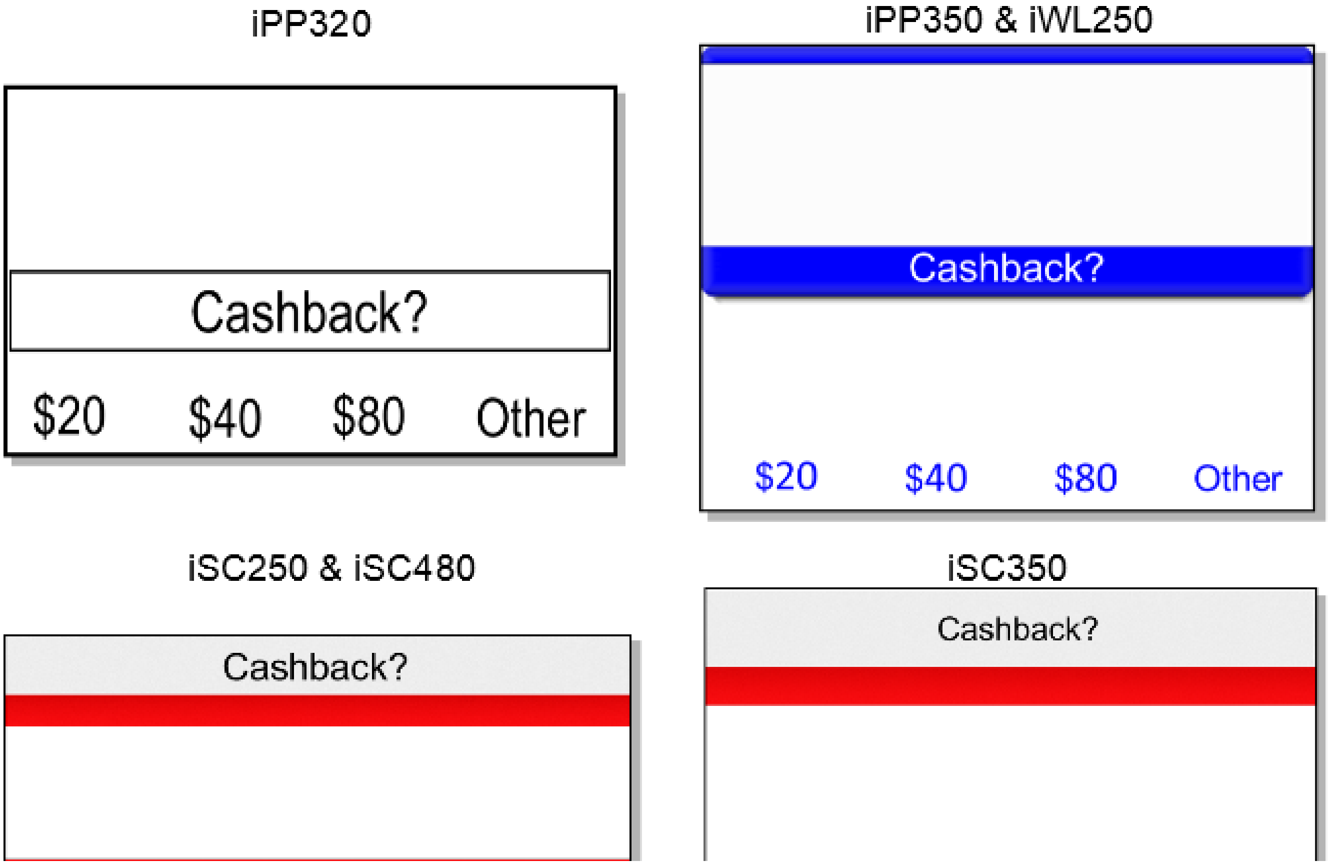


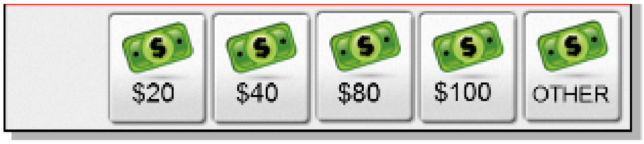
Cashback?

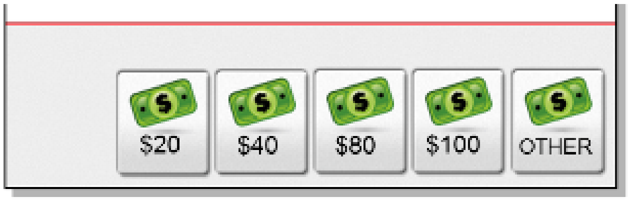
Description	Cash back selection
Initiated By	Cash back being enabled, after debit or EBT selection
Status Response	11.04Cash Back

DFS Data Index	0030_0007
Form	CASHB.K3Z
Text	“Cash back?”
Form Buttons and IDs	iSC250/iSC350/iSC480: • “Other” – O • &lt;?ivCB1?&gt; – 1 • &lt;?ivCB3?&gt; – 2, • &lt;?ivCB2?&gt; – 3 • &lt;?ivCB4?&gt; – 4 All other terminals: • “Yes” – Y • “No” – N
Terminal Buttons Allowed	Cancel

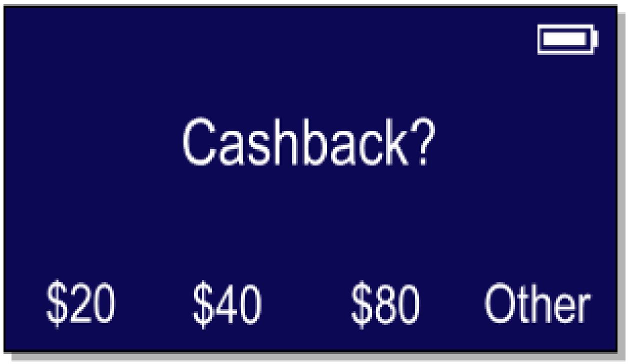
10.7.7.4.2 Cash Back Selection without No



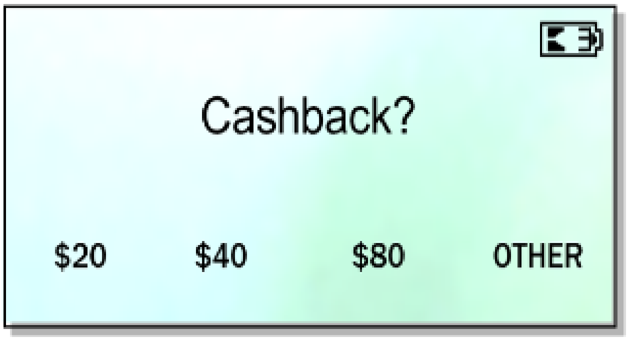




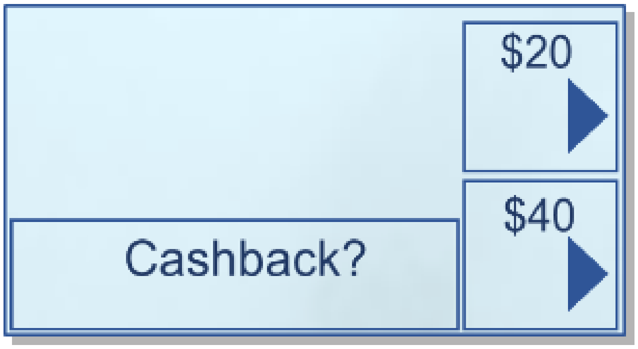
iSMP & iSMPc



iCMP



iUP250

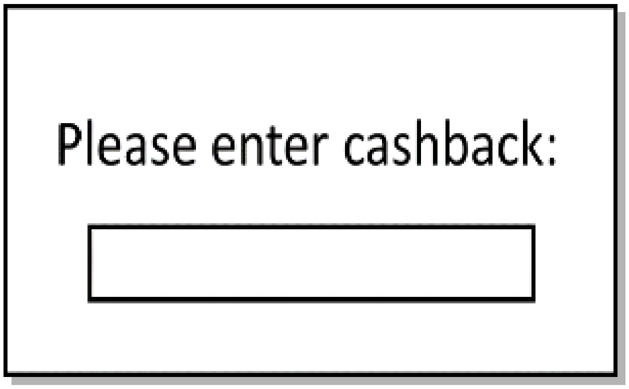


Description	Cash back selection without No button
Initiated By	Cash back being enabled, after debit or EBT selection
Status Response	11.04Cash Back
DFS Data Index	0030_0008
Form	CASHBA.K3Z
Text	“Cash back?”

Form Buttons and IDs	• “Other” • &lt;?ivCB1?&gt; – 1 • &lt;?ivCB3?&gt; – 2 • &lt;?ivCB2?&gt; – 3 • &lt;?ivCB4?&gt; – 4
Terminal Buttons Allowed	Cancel

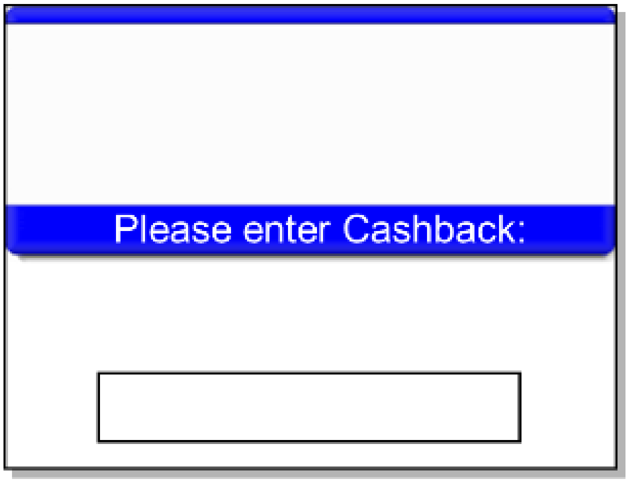
10.7.7.4.3 Cash Back Other

iPP320



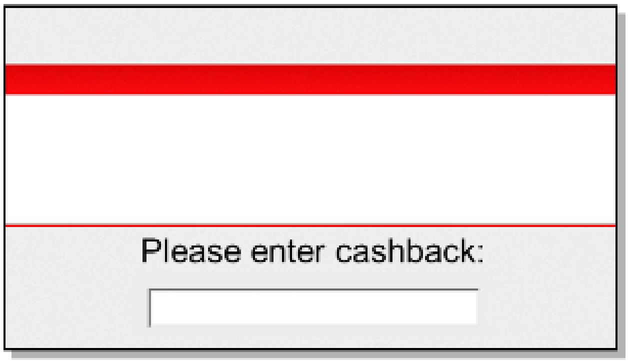
Please enter cashback:

iPP350 & iWL250



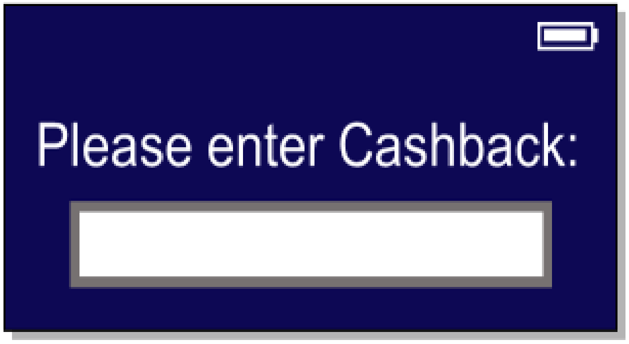
Please enter Cashback:

iSC250 & iSC350 & iSC480



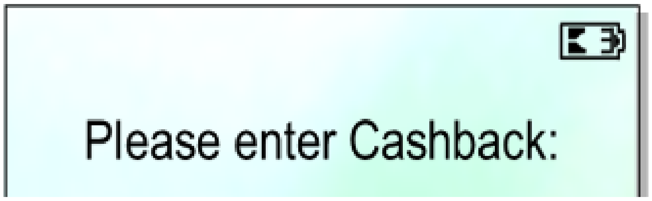
Please enter cashback:

iSMP & iSMPc



Please enter Cashback:

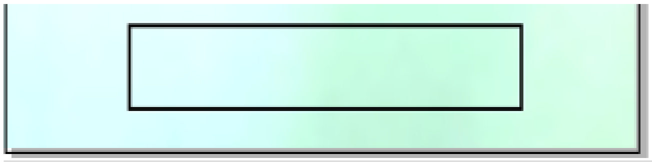
iCMP



Please enter Cashback:

iUP250

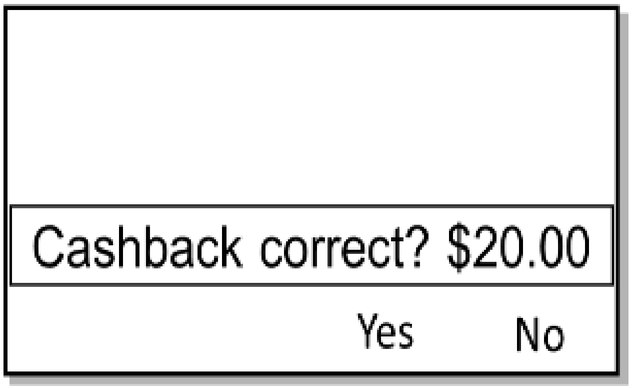
Please enter Cashback:



Description	Manual entry of cash back amount
Initiated By	Other button in cash back selection forms
Status Response	11.04Cash Back
DFS Data Index	0030_0018
Form	CASHBO.K3Z
Text	“Please Enter cash back:”
Form Buttons and IDs	N/A

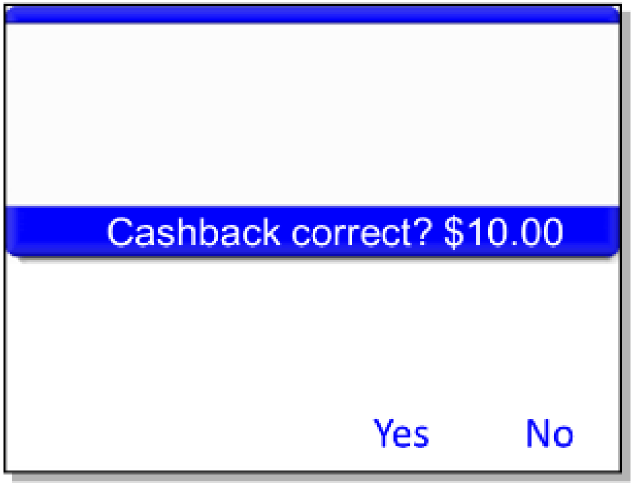
10.7.7.4.4 Cash Back Verification

iPP320



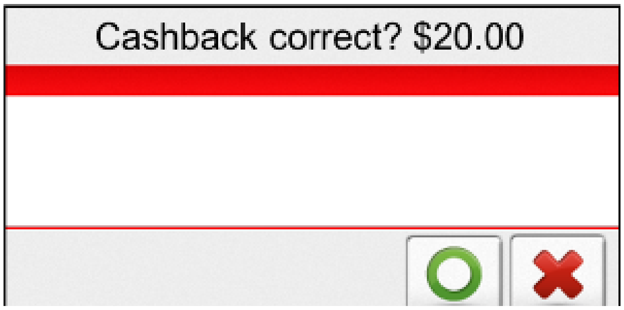
Cashback correct? $20.00

iPP350 & iWL250



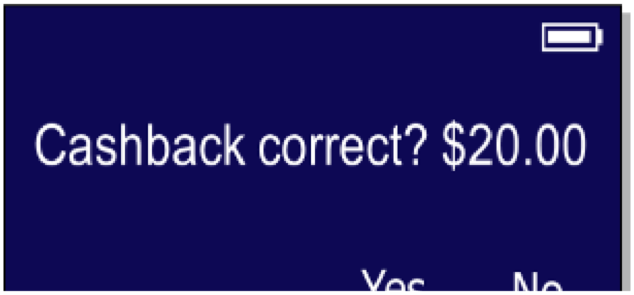
Cashback correct? $10.00

iSC250 & iSC350 & iSC480



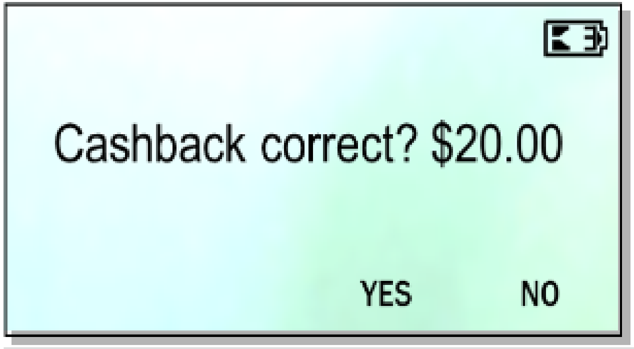
Cashback correct? $20.00

iSMP & iSMPc

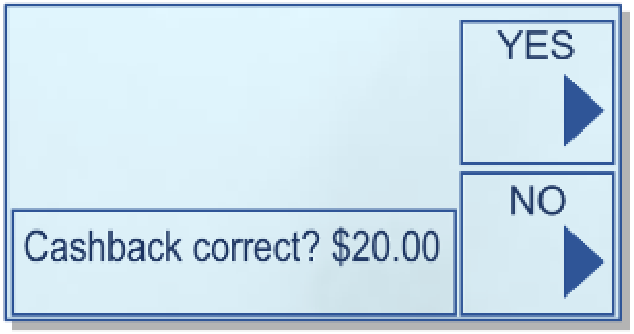


Cashback correct? $20.00

iCMP


Cashback correct? $20.00

iUP250


Cashback correct? $20.00

Description	Cash back verification
Initiated By	Cash back selection
Status Response	11.04Cash Back
DFS Data Index	0030_0009
Form	CASHBV.K3Z
Text	"Cash back correct? $xx.xx"
Form Buttons and IDs	• "Yes" – Y • "No" – N
Terminal Buttons Allowed	Cancel

10.7.7.5 Contactless Enabled Forms

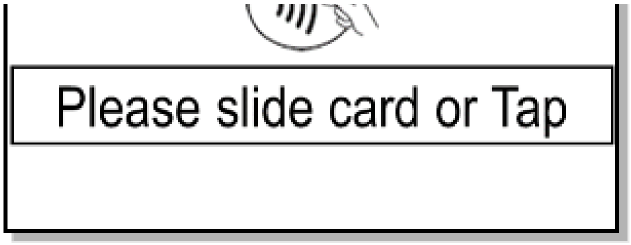
This section includes the following contactless forms:

- Contactless Card Read Request
- Contactless Card Swipe
- Contactless Card Swipe with Language Selection

10.7.7.5.1 Contactless Card Read Request

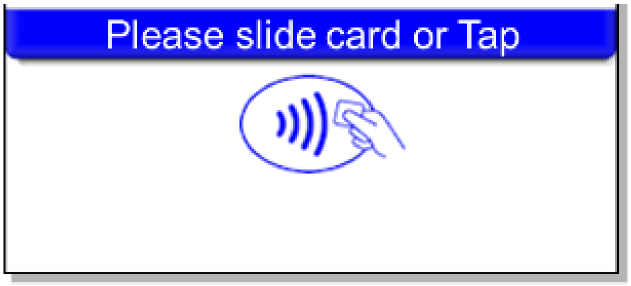
iPP320

iPP350 & iWL250



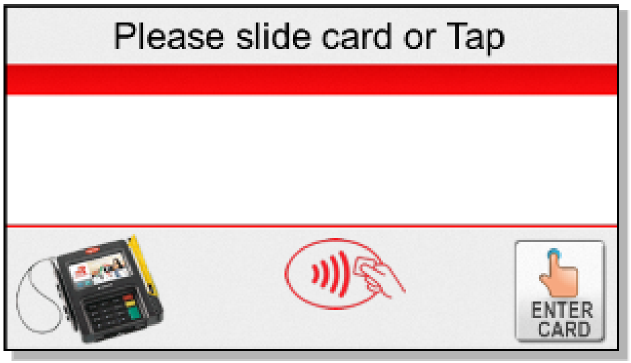
Please slide card or Tap

iSC250



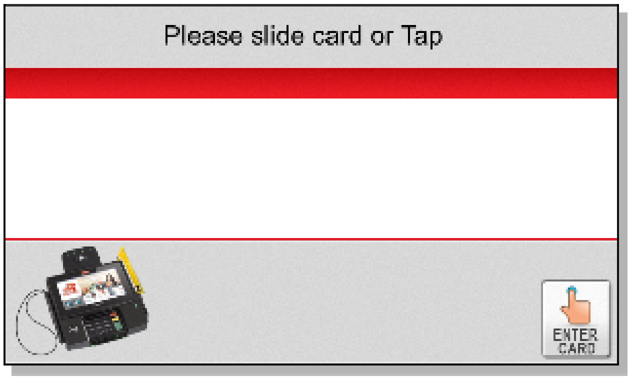
Please slide card or Tap

iSC480



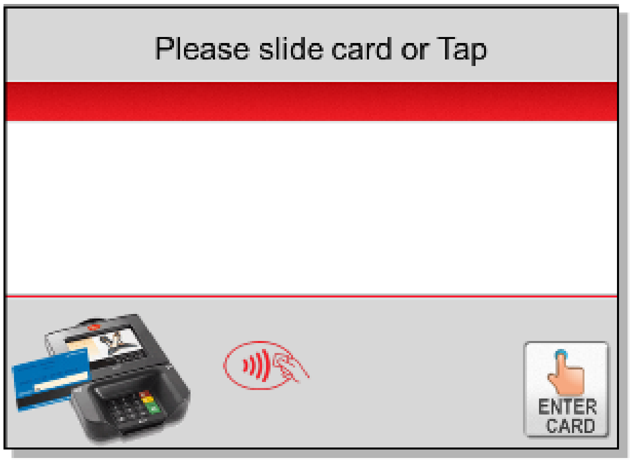
Please slide card or Tap

iSC350



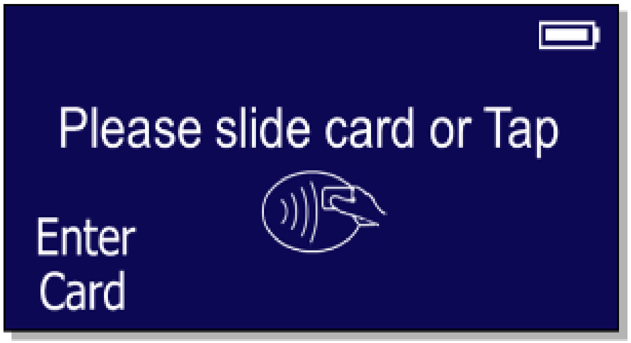
Please slide card or Tap

iSMP & iSMPc



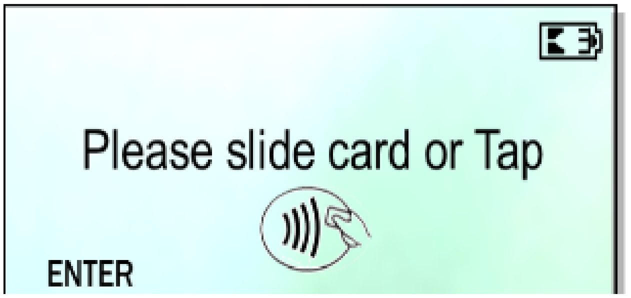
Please slide card or Tap

iCMP

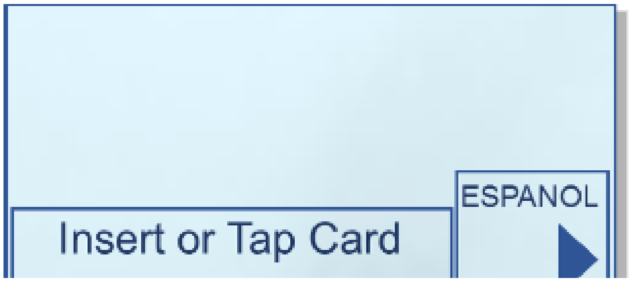


Please slide card or Tap

iUP250



Please slide card or Tap



Insert or Tap Card

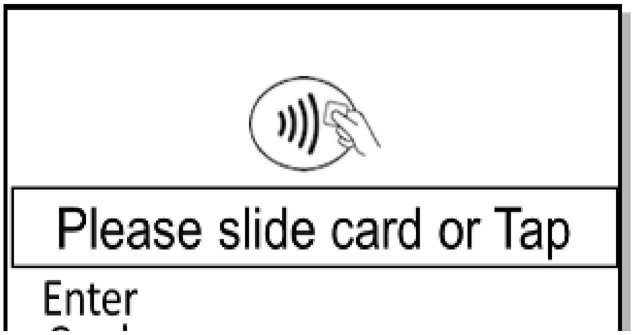
<table>
<tr><td>Description</td><td>Contactless card read request. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0008_0001</td><td>1</td><td>Enable contactless.</td></tr>
</table>
</td></tr>
<tr><td>Initiated By</td><td>23.x Card Read Request message</td></tr>
<tr><td>Status Response upon display</td><td>11.12Input</td></tr>
<tr><td>Status Response after swipe</td><td>11.13Input Accepted</td></tr>
<tr><td>DFS Data Index</td><td>0030_0023</td></tr>
<tr><td>Form</td><td>CCOD.K3Z</td></tr>
<tr><td>Text</td><td>Variable</td></tr>
<tr><td>Form Buttons and IDs</td><td>N/A</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

Info

The "Enter Card" button for manual entry is hidden when the Display “Enter Card” Prompt (configuration parameter '0007_0029') is set to '0'. In order for the manual entry button to be displayed, this configuration parameter must be set to a value of '1' to '4' as described in the Main Flow (mainFlow.dat) section of this document.

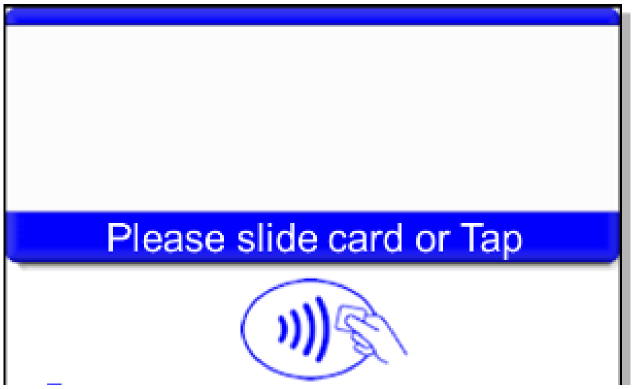
10.7.7.5.2 Contactless Card Swipe

iPP320



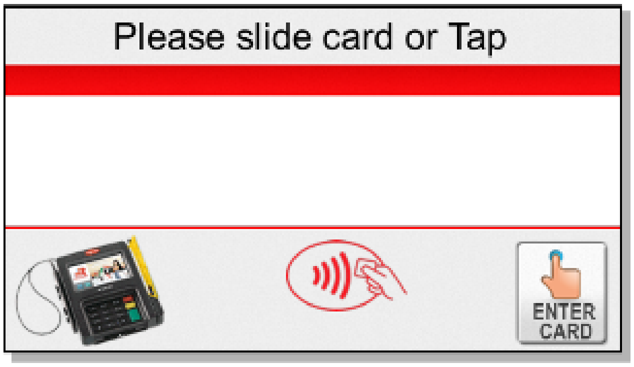
Please slide card or Tap

iPP350 & iWL250

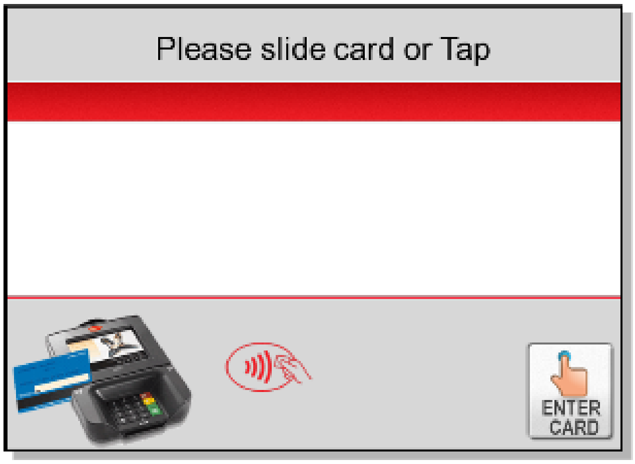


Please slide card or Tap

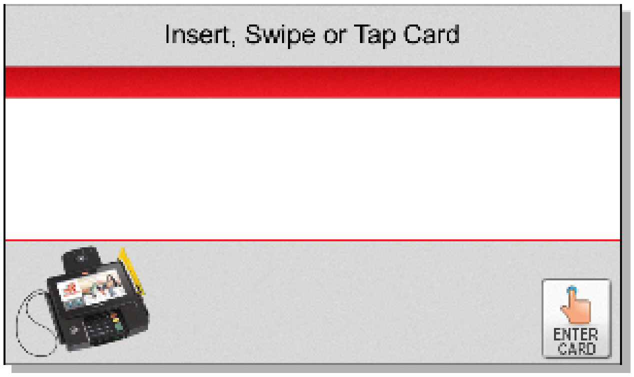
iSC250


Please slide card or Tap

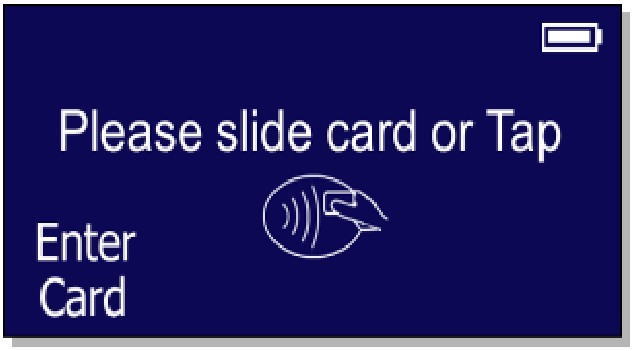
iSC350


Please slide card or Tap

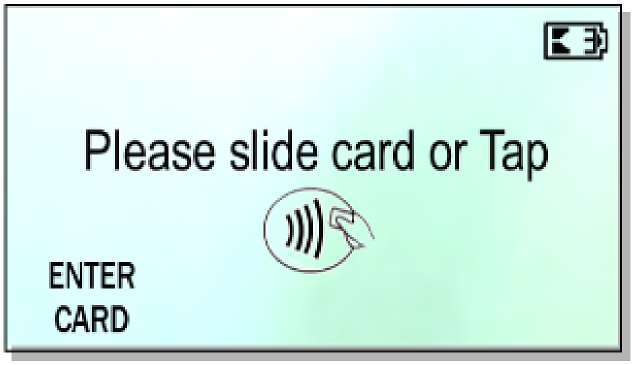
iSC480


Insert, Swipe or Tap Card

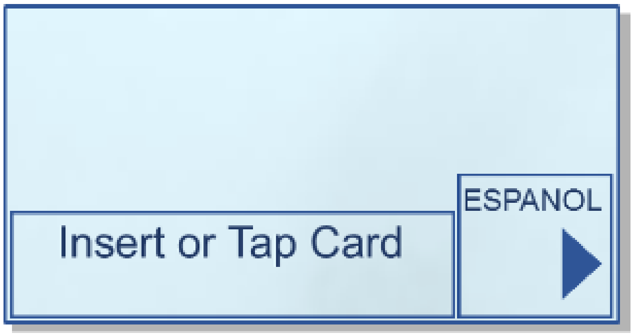
iSMP & iSMPc


Please slide card or Tap

iCMP


Please slide card or Tap

iUP250

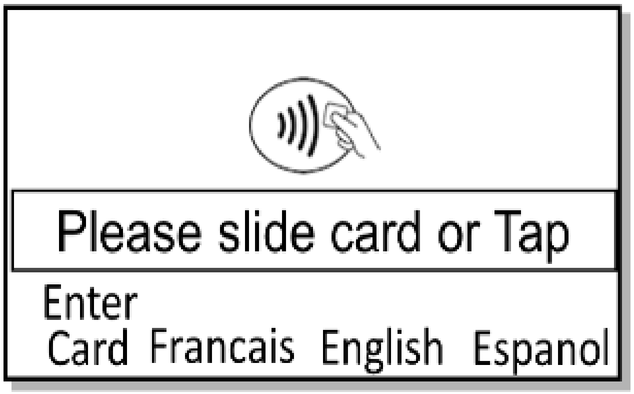

Insert or Tap Card

<table>
<tr><td>Description</td><td>Contactless card swipe. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0008_0001</td><td>1</td><td>Enable contactless.</td></tr>
</table>
</td></tr>
<tr><td>Initiated By</td><td>01.x Online message and language disabled</td></tr>
<tr><td>Status Response</td><td>11.01SlideCard</td></tr>
<tr><td>DFS Data Index</td><td>0030_0021</td></tr>
<tr><td>Form</td><td>CSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Please slide card or Tap”</td></tr>
<tr><td>Form Buttons and IDs</td><td>• “Enter Card” – M
• "ESPANOL", iUP250 only</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

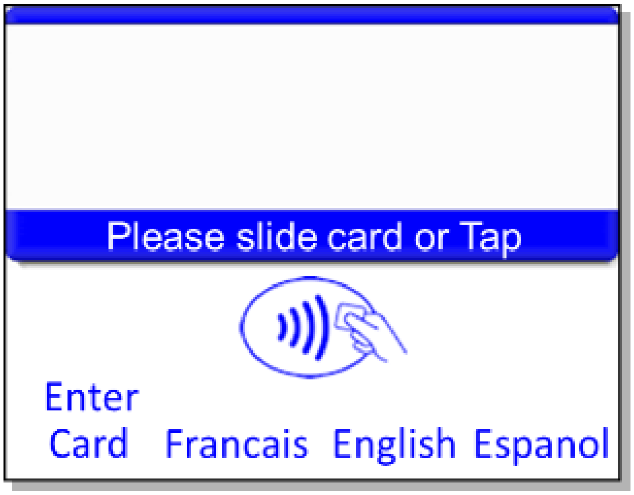
> The PayPal version of this form is `CPSWIPE.K3Z`. Note that the PayPal form is not available for iPP320 and iPP350 terminals.

10.7.7.5.3 Contactless Card Swipe with Language Selection

iPP320

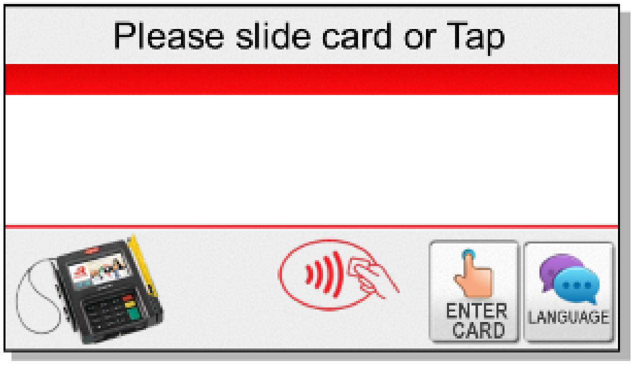


iPP350 & iWL250

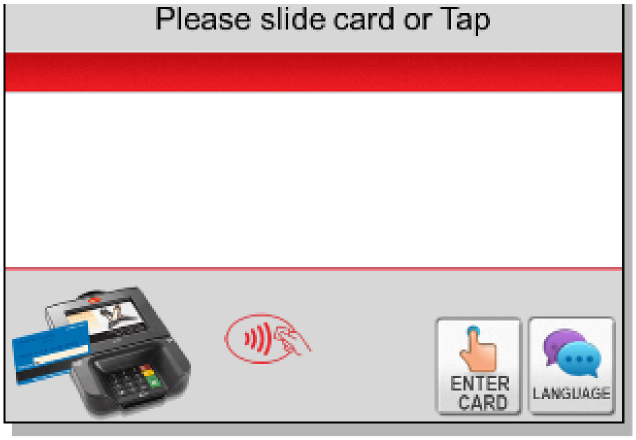


iSC250

iSC350



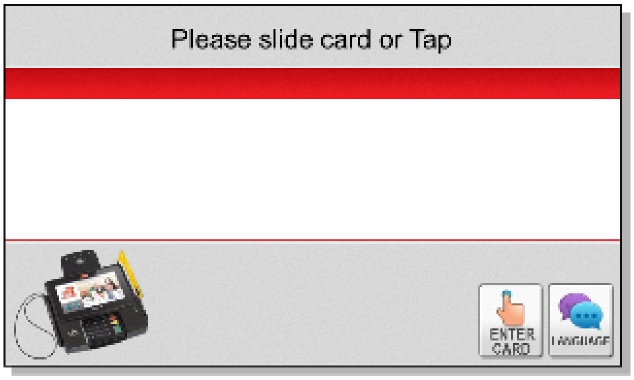
Please slide card or Tap



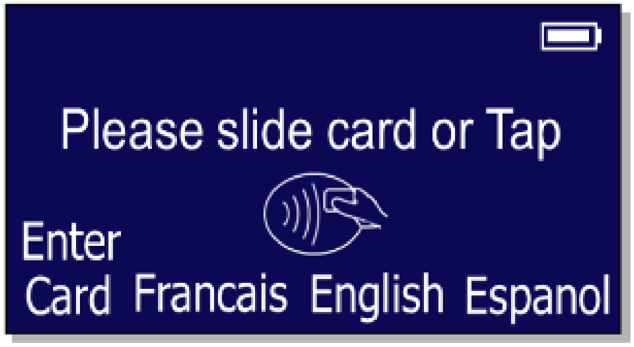
Please slide card or Tap

iSC480

iSMP & iSMPc



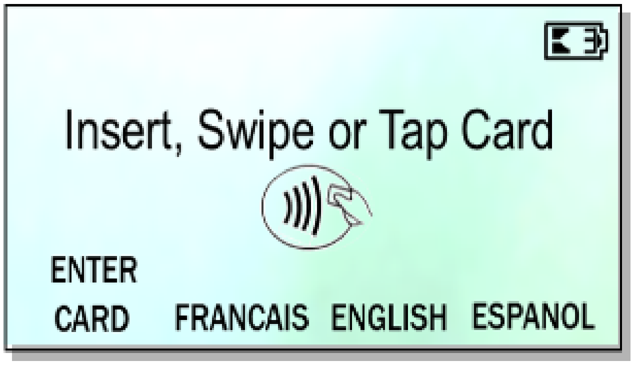
Please slide card or Tap



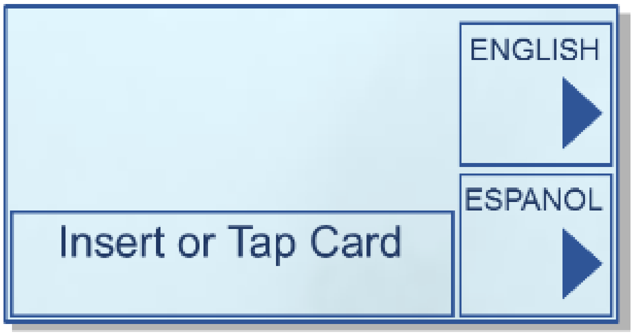
Please slide card or Tap

iCMP

iUP250



Insert, Swipe or Tap Card



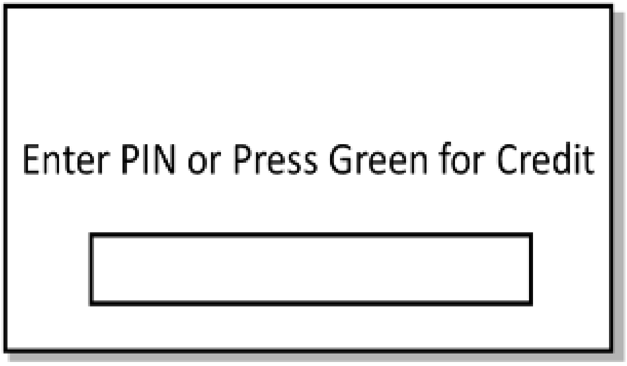
Insert or Tap Card

Description	Contactless EMV and Swipe with Language. Settings are as follows: **Parameter** / **Setting** / **Description** 0019_0001 / 1 / Enable EMV transactions. 0008_0001 / 1 / Enable contactless. 0007_0004 / 1 / Combine language with swipe screen. 0007_0005 / 0 / Swipe card first, then select payment type. 0007_0029 / 1 / Display "Enter Card" button and prompt for card number, expiration date and CVV. 0008_0001 / 9 / Enable contactless for EMV. This parameter controls the display of the "Enter Card" button on the card swipe screen and enables cardholder prompts for manual entry. When not set to '0', the "Enter Card" button will be displayed with prompt options. Additional settings for parameter '0007_0029' used during manual entry include: • 2 = Display "Enter Card" button and prompt for card number and expiration date (no CVV). • 3 = Display "Enter Card" button and prompt for card number and CVV (no expiration date). • 4 = Display "Enter Card" button and prompt for card number (no expiration date or CVV).
Initiated By	01.x Online message
Status Response	11.01 Slide/Insert/Tap Card
DFS Data Index	0030_0039
Form	CELSWIPE.K3Z
Text	“Insert, Swipe or Tap Card”

Form Buttons and IDs	iPP320/iPP350/iWL250/iMP350/iMP352 - “Enter Card” – M - "Francais" - "English" - "Espanol" iSC250/iSC350/iSC480 - "ENTER CARD" - M - "LANGUAGE" - L iUP250 - "English" - "Espanol"

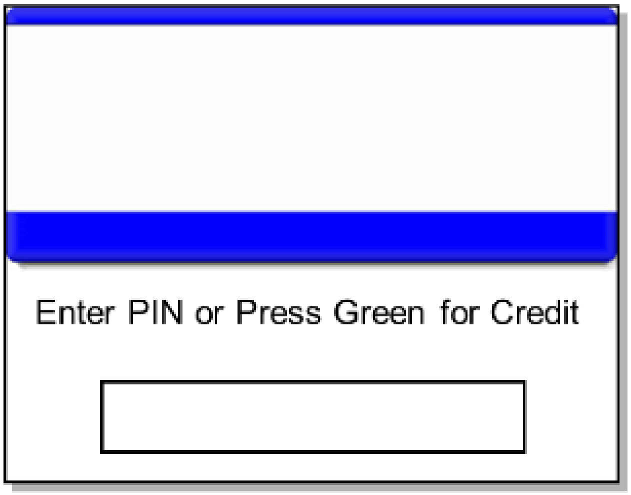
10.7.7.6 Enter PIN or Press Green for Credit

iPP320



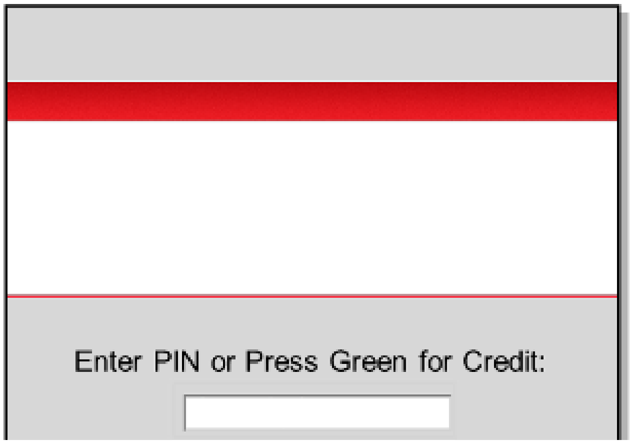
Enter PIN or Press Green for Credit

iPP350 & iWL250



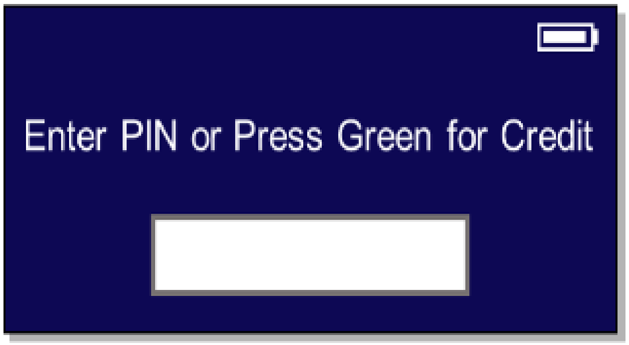
Enter PIN or Press Green for Credit

iSC250 & iSC350 & iSC480



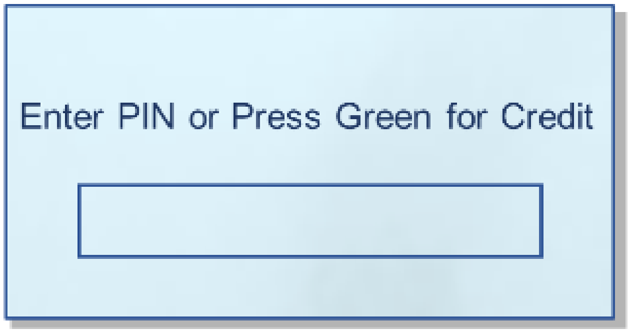
Enter PIN or Press Green for Credit:

iSMP & iSMPc



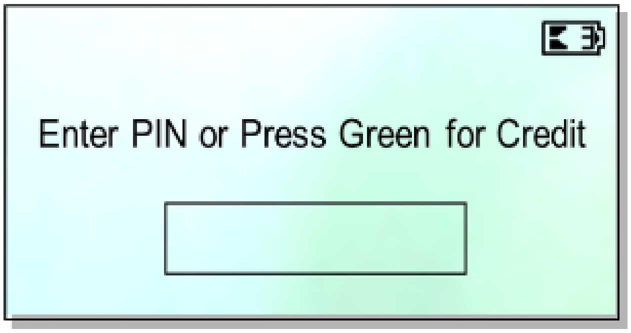
Enter PIN or Press Green for Credit

iUP250



Enter PIN or Press Green for Credit

iCMP

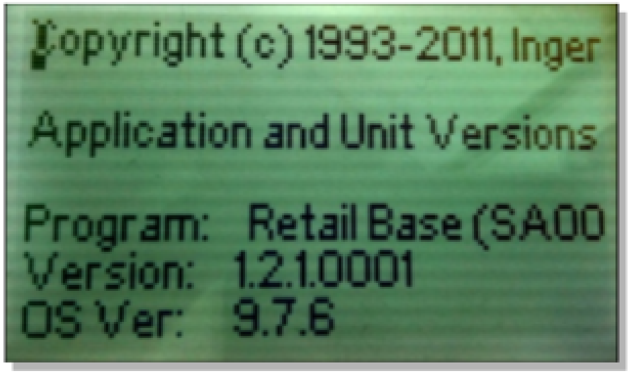


Enter PIN or Press Green for Credit

Description	PIN entry or Credit Selection
Initiated By	Debit or EBT selection
Status Response	11.12 Input[FS]
DFS Data Index	0030_0020
Form	INPUT.K3Z /*Amount manual entry */
Text	“Enter PIN or Press Green for Credit”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	All

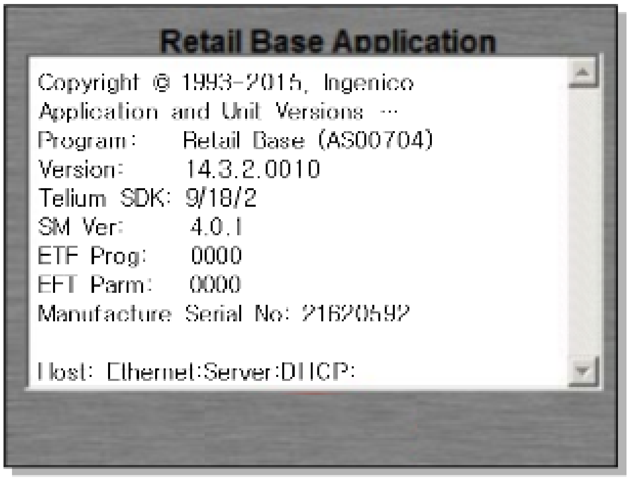
10.7.7.7 Initialisation Screen

iPP320



```
Copyright (c) 1993-2011, Inger
Application and Unit Versions
Program:  Retail Base (SA00
Version:  1.2.1.0001
OS Ver:   9.7.6
```

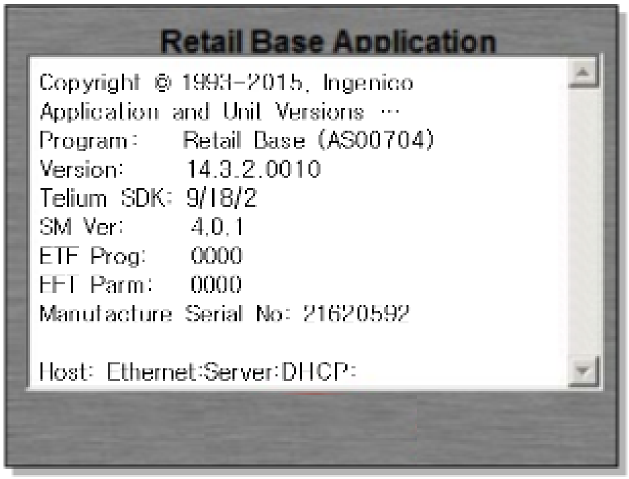
iPP350 & iWL250



Retail Base Application

```
Copyright © 1993-2015, Ingenico
Application and Unit Versions ...
Program:     Retail Base (AS00704)
Version:     14.3.2.0010
Telium SDK:  9/18/2
SM Ver:      4.0.1
ETF Prog:    0000
EFT Parm:    0000
Manufacture Serial No: 21620592

Host: Ethernet:Server:DHCP:
```

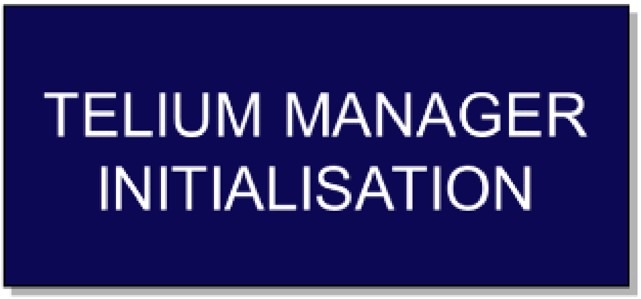
iSC250 & iSC350 & iSC480



Retail Base Application

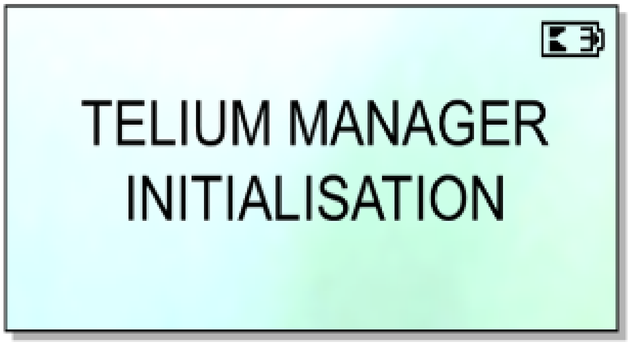
```
Copyright © 1993-2015, Ingenico
Application and Unit Versions ...
Program:     Retail Base (AS00704)
Version:      14.3.2.0010
Telium SDK:  9/18/2
SM Ver:       4.0.1
ETF Prog:     0000
EFT Parm:     0000
Manufacture Serial No: 21620592

Host: Ethernet:Server:DHCP:
```

iSMP & iSMPc



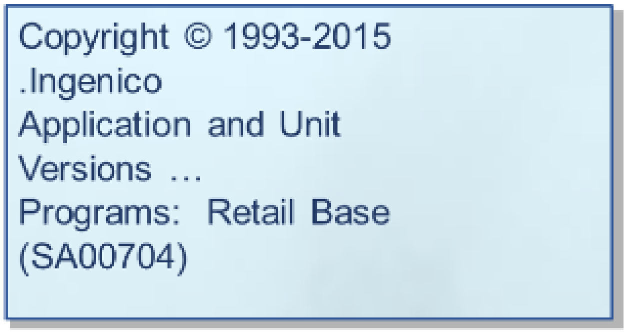
TELIUM MANAGER
INITIALISATION

iCMP



TELIUM MANAGER
INITIALISATION

iUP250



Copyright © 1993-2015
.Ingenico
Application and Unit
Versions ...
Programs: Retail Base
(SA00704)

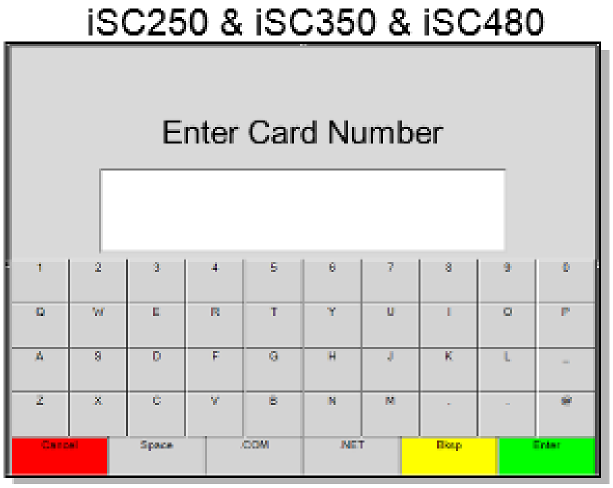
Description	Boot screen
Initiated By	Terminal boot sequence
Status Response	N/A
DFS Data Index	N/A
Form	BOOT.K3Z
Text	Fixed diagnostic information
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

10.7.7.8 Input Entry Forms

This section includes the following forms:

- Alphanumeric Entry
- Input Entry

10.7.7.8.1 Alphanumeric Entry



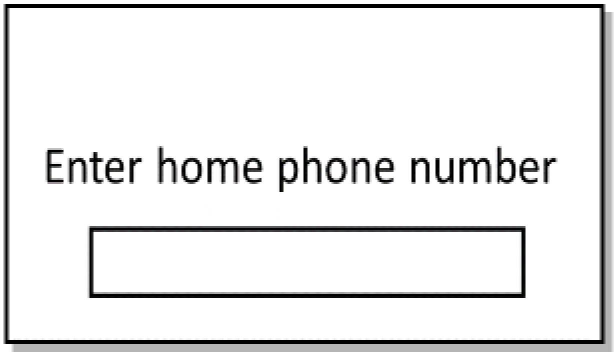
Enter Card Number

This form is not available for the iCMP, iSMP, iSMPc, iPP320, iPP350 or iWL250.

Description	Data entry form
Initiated By	27.x: Alpha Input Message (On-Demand)
Status Response upon display	11.12 Input
Status Response after data entry	11.13 Input Accepted
DFS Data Index	0030_0017
Form	ALPHA.K3Z
Text	Variable
Form Buttons and IDs	• Alphanumeric keyboard (for alpha entry) • “Cancel” • “Backspace” • “Enter”
Terminal Buttons Allowed	All

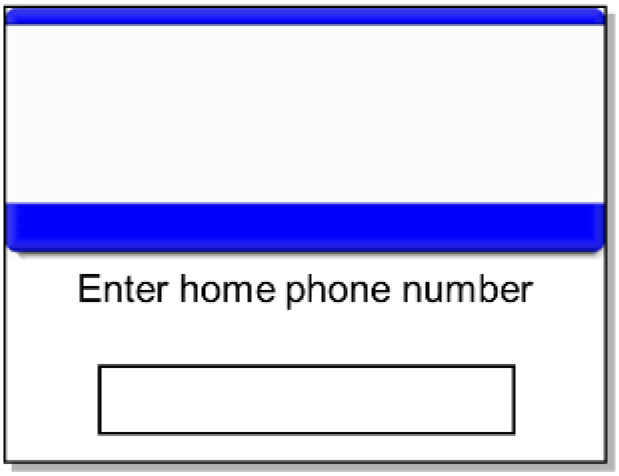
10.7.7.8.2 Input Entry

iPP320



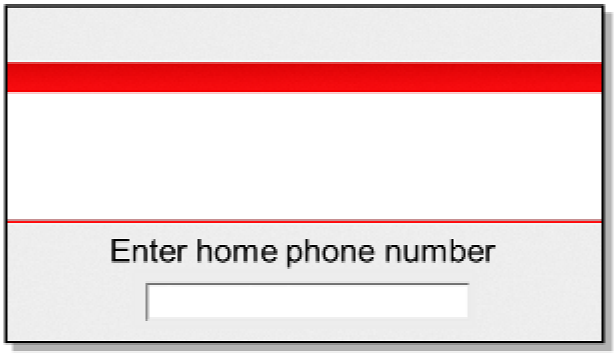
Enter home phone number

iPP350 & iWL250



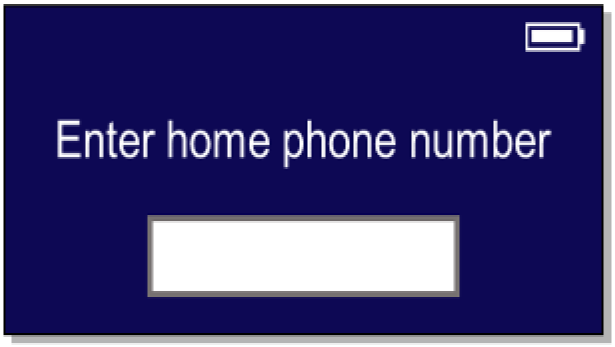
Enter home phone number

iSC250 & iSC350 & iSC480



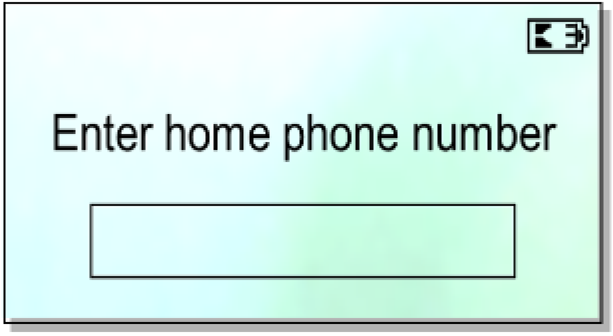
Enter home phone number

iSMP & iSMPc



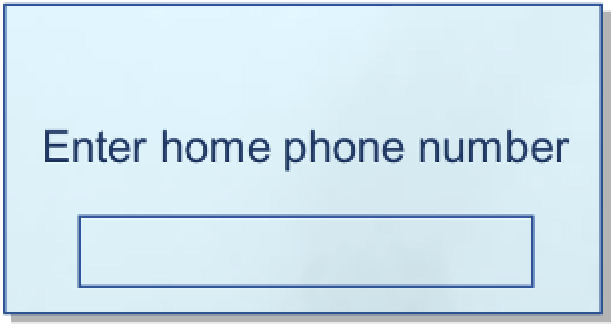
Enter home phone number

iCMP



Enter home phone number

iUP250



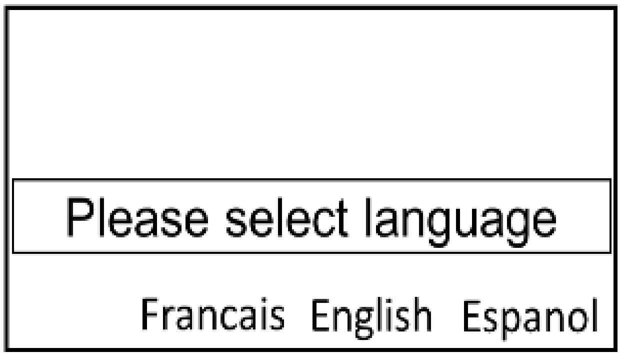
Enter home phone number

Description	Numeric input entry
Initiated By	21.x: Numeric Input Request Message (On-Demand)
Status Response upon display	11.12Input

Status Response after swipe	11.13Input Accepted
DFS Data Index	0030_0020
Form	INPUT.K3Z
Text	Variable
Form Buttons and IDs	N/A
Terminal Buttons Allowed	All

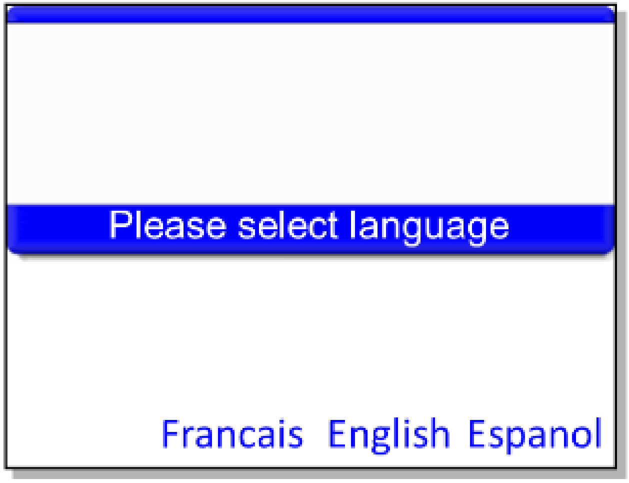
10.7.7.9 Language Selection

iPP320



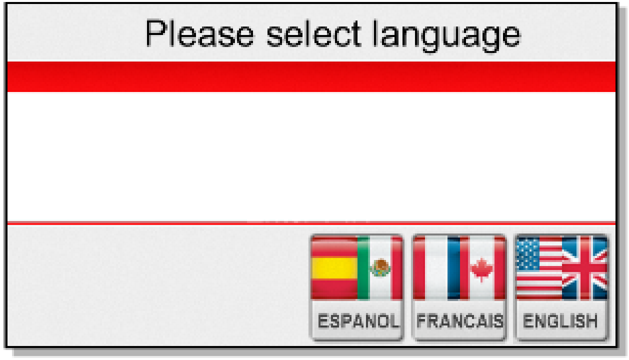
Please select language

iPP350 & iWL250



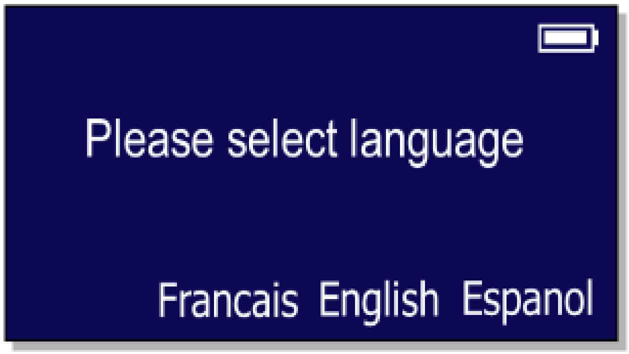
Please select language

iSC250 & iSC350 & iSC480



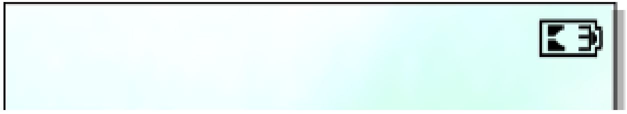
Please select language

iSMP & iSMPc

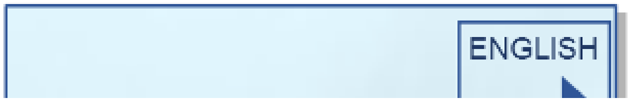


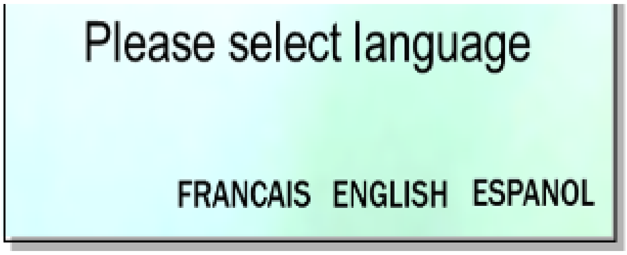
Please select language

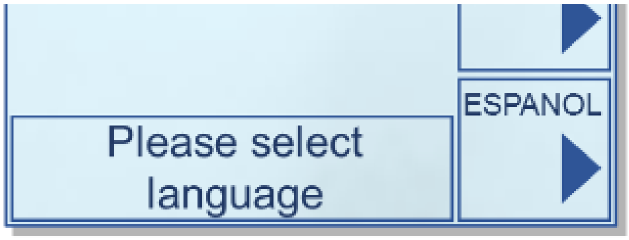
iCMP



iUP250



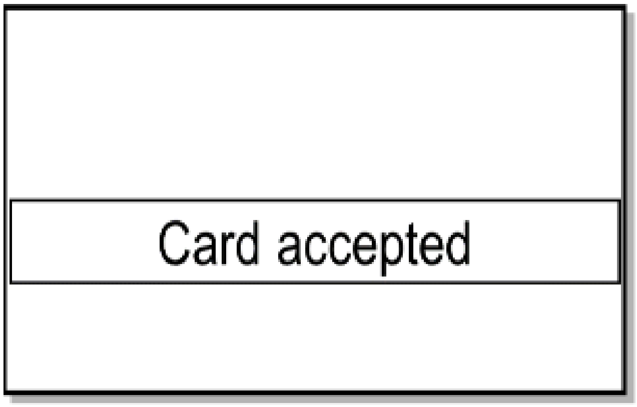

Please select language


Please select language

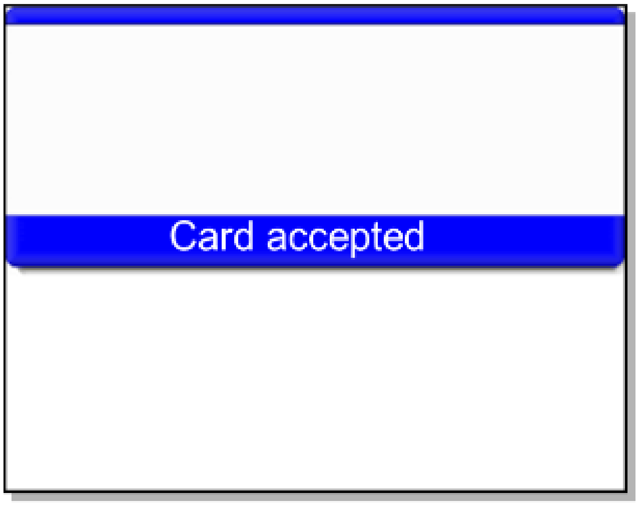
Description	Language selection
Initiated By	Flag in mainflow.dat
Status Response	
DFS Data Index	0030_0003
Form	LANG.K3Z
Text	“Please select language”
Form Buttons and IDs	• “French” or “Francais” – 3 • “English” – 1 • “Español” – 2
Terminal Buttons Allowed	Cancel

10.7.7.10 Message

iPP320


Card accepted

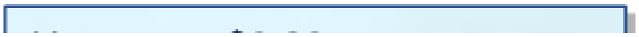
iPP350 & iWL250

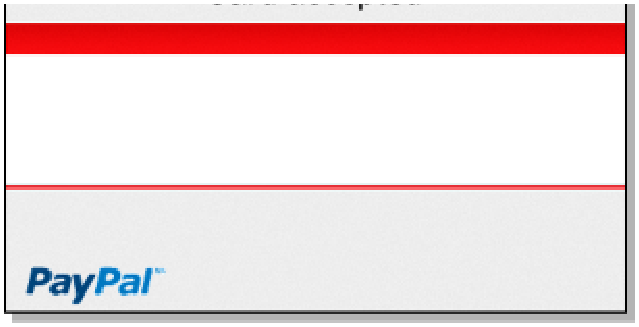

Card accepted

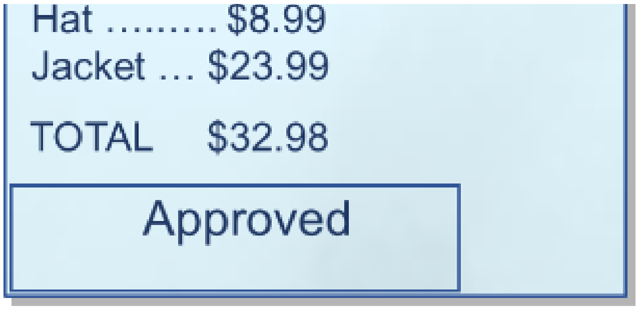
iSC250 & iSC350 & iSC480

Card accepted

iUP250

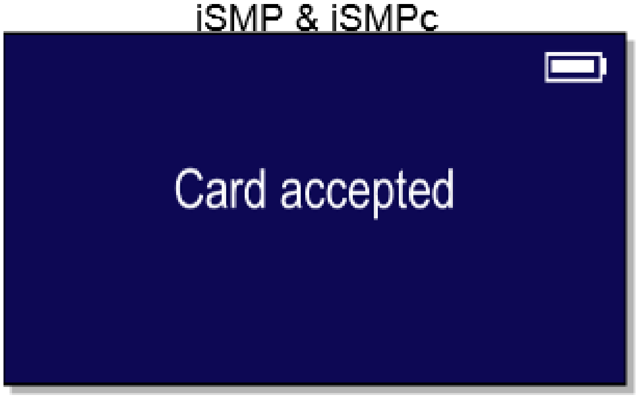






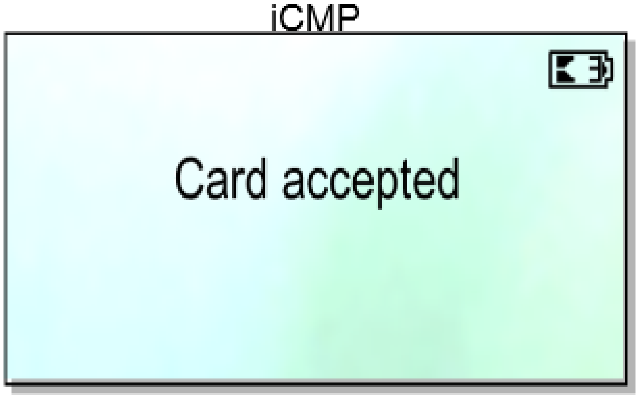
Hat $8.99
Jacket ... $23.99
TOTAL $32.98
Approved

iSMP & iSMPc



Card accepted

iCMP



Card accepted

Specification	Description
Description	Status response
Initiated By	After card swipe from 23.x Card Read Request
Status Response	11.13Input Accepted
DFS Data Index	0030_0002
Form	MSG.K3Z
Text	(variable) “insert your text here”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

Examples follow.

Description	Status response
Initiated By	After amount verification, waiting on host response

Status Response	11.05Processing
DFS Data Index	0030_0002
Form	MSG.K3Z
Text	“Processing …Please wait”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

Description	Status response
Initiated By	Payment selected, Terminal waiting for amount
Status Response	11.04Please Wait
DFS Data Index	0030_0002
Form	MSG.K3Z
Text	“Please wait for the cashier”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

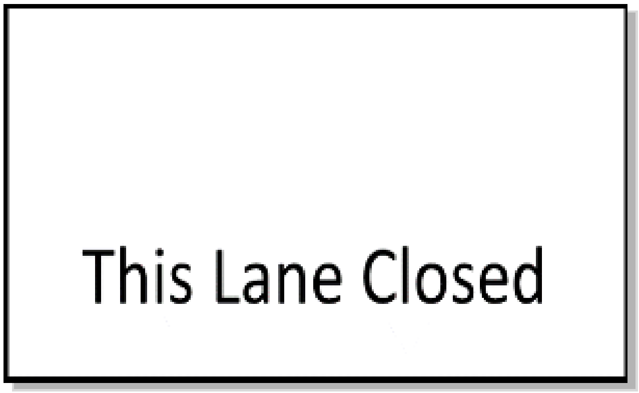
Description	Host response
Initiated By	0x and 50.x
Status Response	11.06Approved
DFS Data Index	0030_0002
Form	MSG.K3Z
Text	“Approved”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

Description	Signature accepted

Initiated By	After OK press on Signature screen
Status Response	11.11Signature Accepted
DFS Data Index	0030_0002
Form	MSG.K3Z
Text	“Signature accepted”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

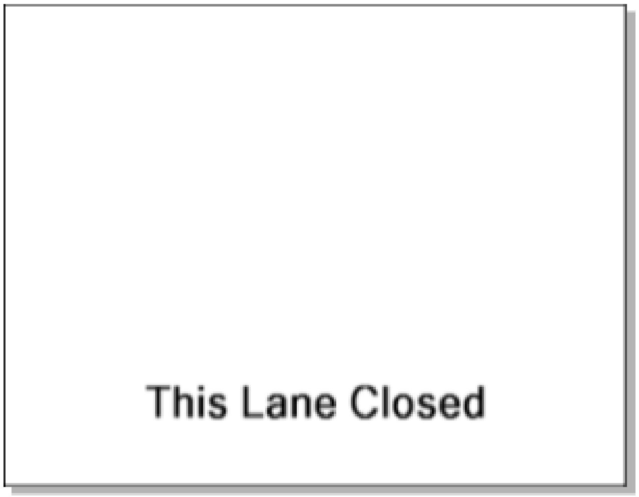
10.7.7.11 Offline

iPP320



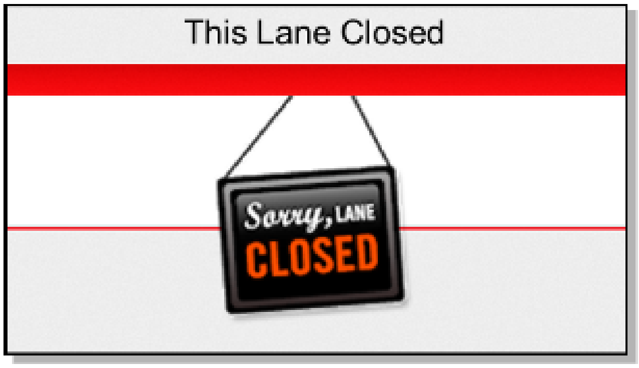
This Lane Closed

iPP350 & iWL250



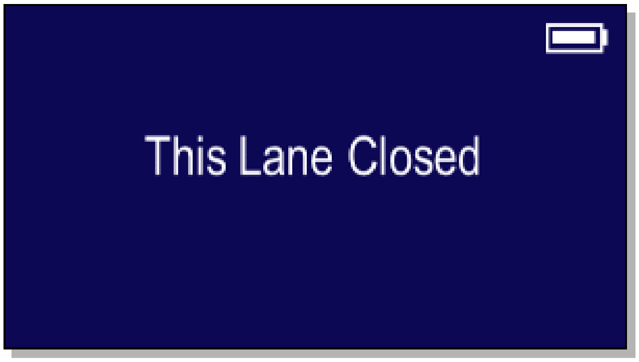
This Lane Closed

iSC250 & iSC350 & iSC480



This Lane Closed

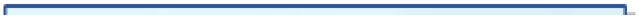
iSMP & iSMPc

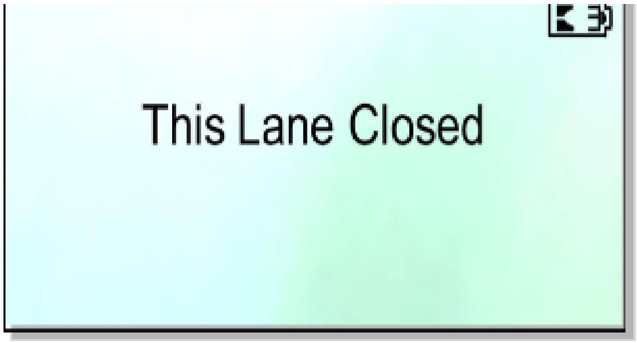


This Lane Closed

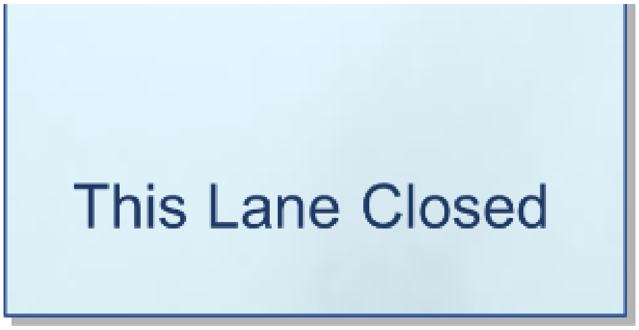
iCMP

iUP250





This Lane Closed



This Lane Closed

Description	Offline screen
Initiated By	00.x
Status Response	11.00Lane Closed
DFS Data Index	0030_0001
Form	OFFLINE.K3Z
Text	“This Lane Closed”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

Info

The PayPal version of this form is `PPOFFLINE.K3Z`. Note that the PayPal form is not available for iPP320 and iPP350 terminals.

10.7.7.12 Payment Forms

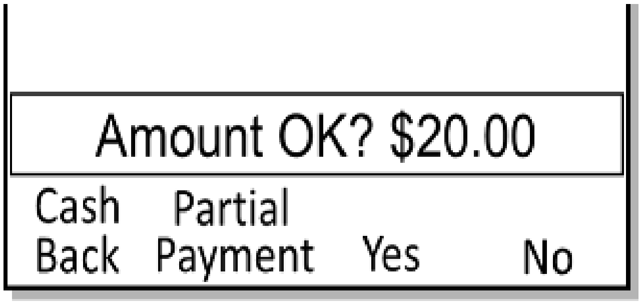
This section includes the following forms:

- Payment Selection
- Amount Verification

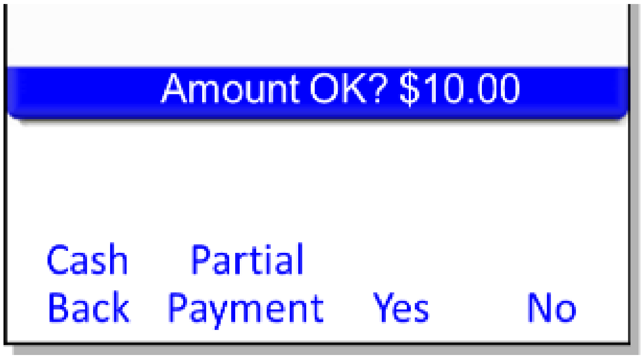
10.7.7.12.1 Amount Verification

iPP320

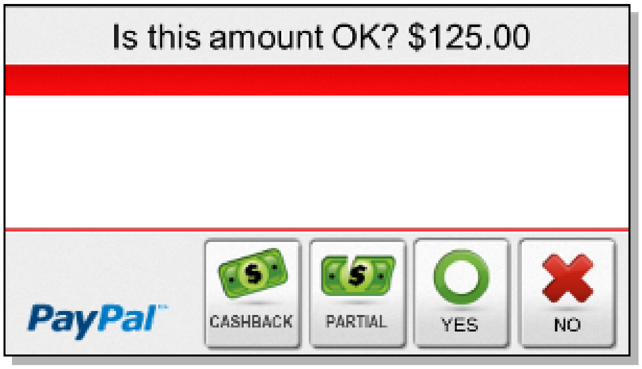
iPP350 & iWL250


Amount OK? $20.00

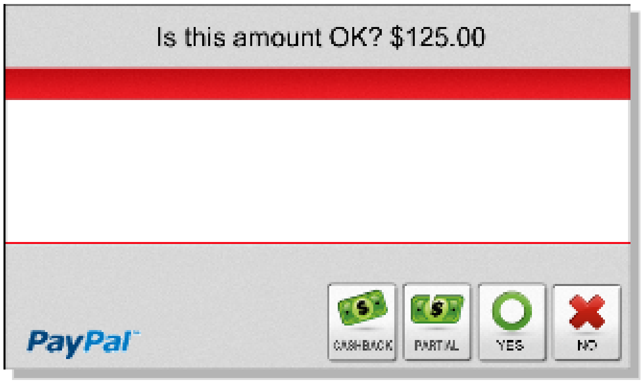
iSC250 & iSC350


Amount OK? $10.00

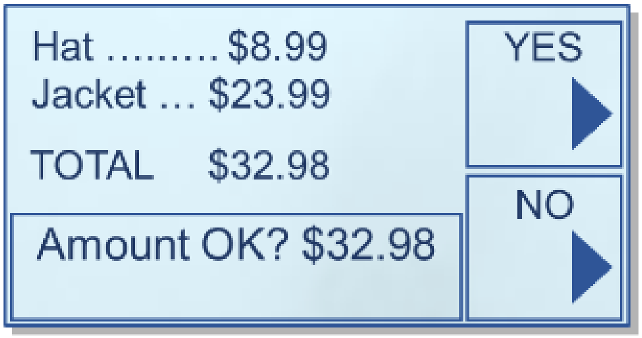
iSC480


Is this amount OK? $125.00

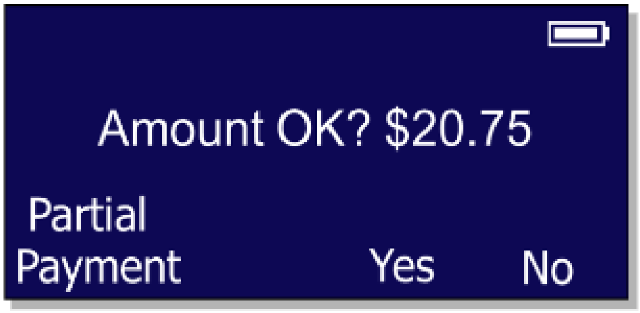
iUP250

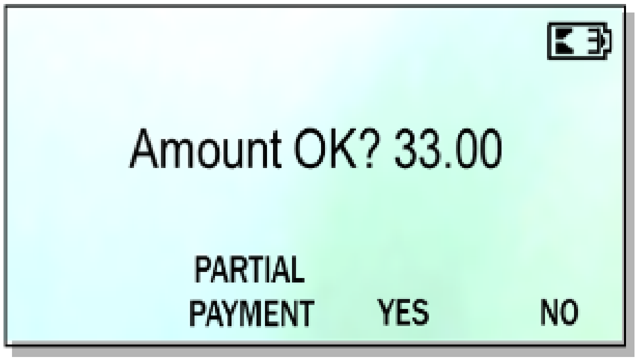

Is this amount OK? $125.00

iSMP & iSMPc


Hat $8.99
Jacket ... $23.99
TOTAL $32.98
Amount OK? $32.98

iCMP


Amount OK? $20.75


Amount OK? 33.00

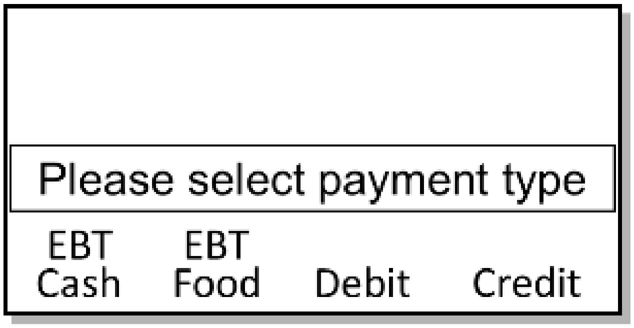
Description	Amount verification
Initiated By	After payment selection
Status Response	11.04Please Wait
DFS Data Index	0030_0010
Form	AMTV.K3Z
Text	“Amount OK? $xx.xx”
Form Buttons and IDs	• “Partial Payment” – P • “Cashback” – C • “Yes” – Y • “No” – N
Terminal Buttons Allowed	Cancel

10.7.7.12.2 Payment Selection

Info

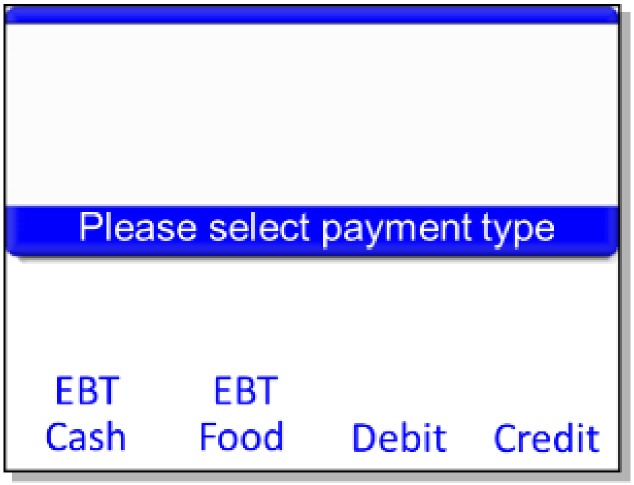
The form format discussed below applies to the default payment selection screen (`pay1.K3Z`) as well as any custom payment screens that may be added (`pay2.K3Z` through `pay9.K3Z`).

iPP320



Please select payment type

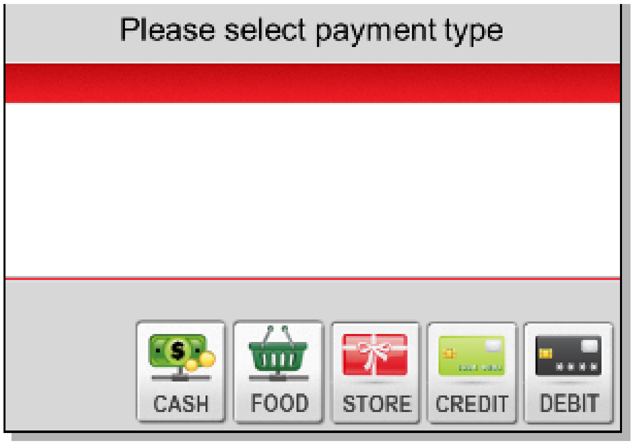
iPP350 & iWL250

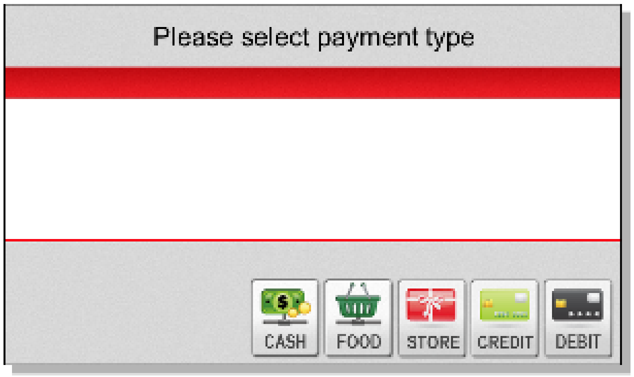


Please select payment type

iSC250 & iSC350

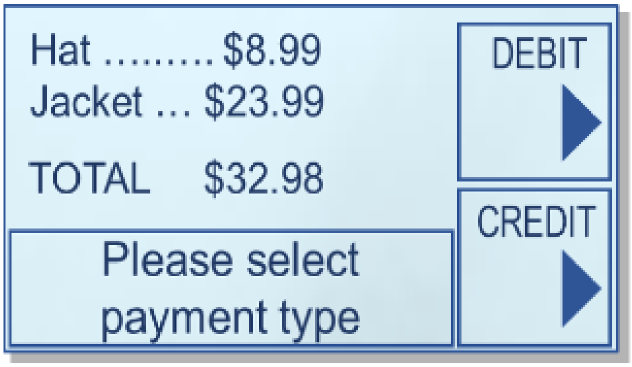
iSC480

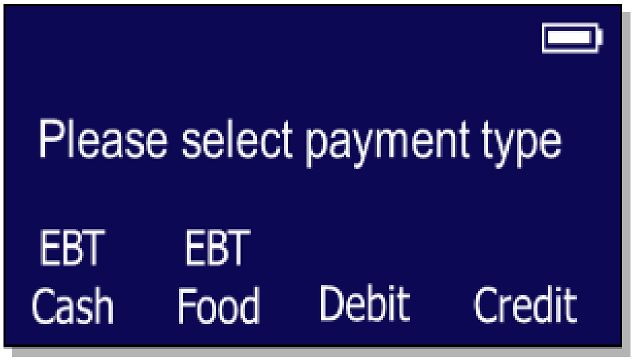

Please select payment type


Please select payment type

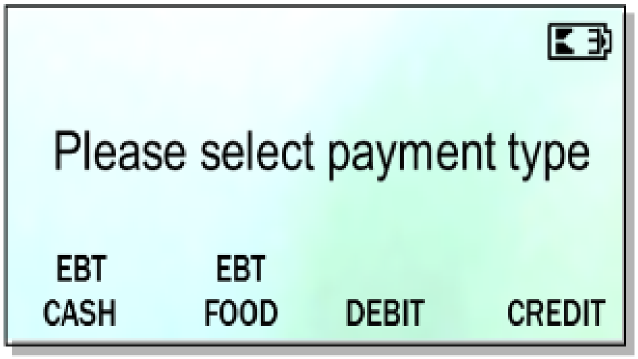
iUP250

iSMP & iSMPc


Hat $8.99
Jacket ... $23.99
TOTAL $32.98
Please select payment type


Please select payment type

iCMP


Please select payment type

Description	Select Payment Type
Initiated By	Card swipe
Status Response	11.02Select Payment
DFS Data Index	0030_0006

Form	PAY%d.K3Z (%d Replaced by numbers 1-9)
Text	“Please select payment type”
Form Buttons and IDs	iPP320/iPP350/iWL250 • “EBT Cash” – C • “EBT Food” – D • “Debit” – A • “Credit” – B iUP250 • “Debit” – A • “Credit” – B iSC250/iSC350/iSC480 • iPP320/iPP350/iWL250 • “EBT Cash” – C • “EBT Food” – D • “Store” – E • “Debit” – A • “Credit” – B
Terminal Buttons Allowed	Cancel

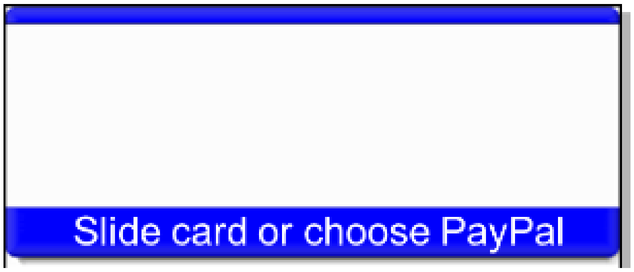
10.7.7.13 PayPal Forms

This section includes the following forms:

- Card Swipe with PayPal
- Card Swipe with PayPal and Language Selection
- Contactless Card Swipe with PayPal Selection
- Contactless Card Swipe with PayPal and Language Selection
- PayPal Data Input (On-Demand form)
- PayPal PIN Entry
- PayPal Please Wait

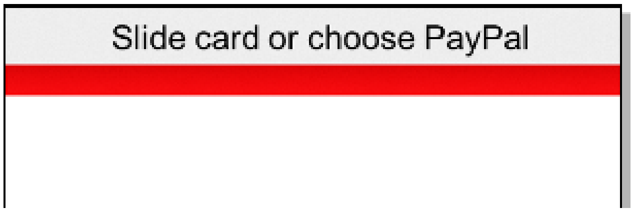
10.7.7.13.1 Card Swipe with PayPal

iPP350

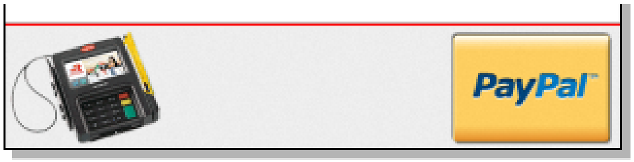


Slide card or choose PayPal

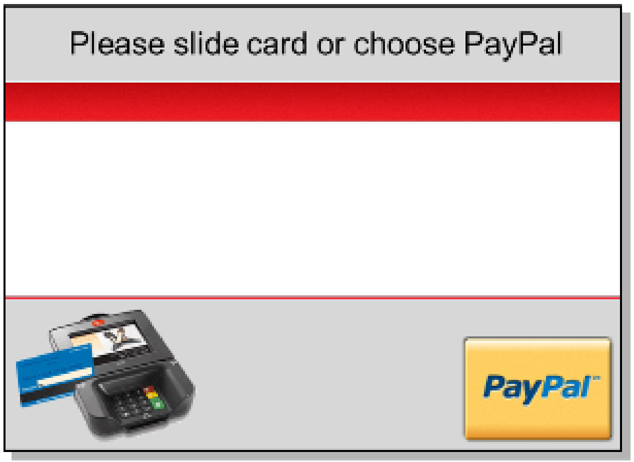
iSC250



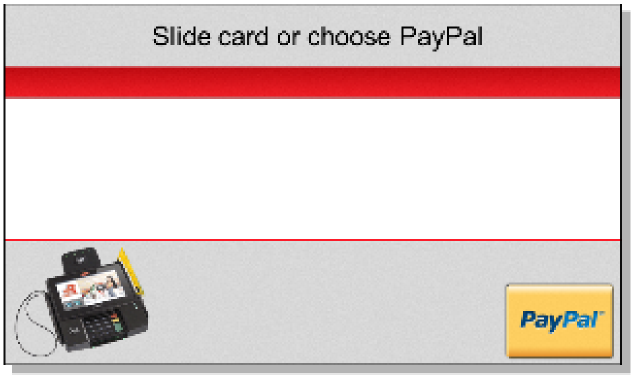
Slide card or choose PayPal



iSC350


Please slide card or choose PayPal

iSC480

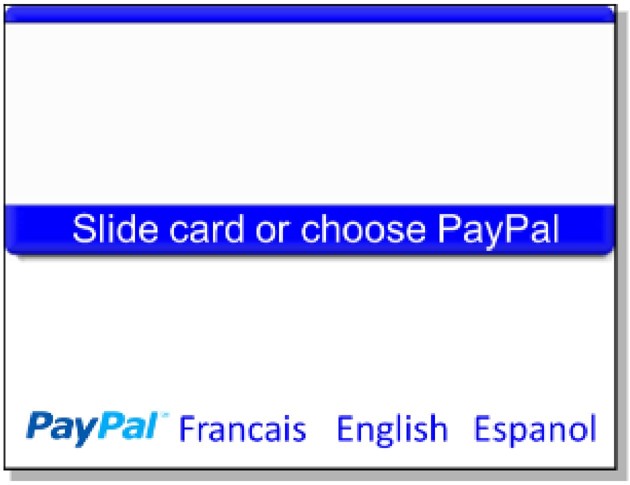

Slide card or choose PayPal

This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

<table>
<tr><td>Description</td><td>Card swipe with PayPal. Settings are as follows:<table><tr><th>Parameter</th><th>Setting</th><th>Description</th></tr><tr><td>0040_0006</td><td>1</td><td>PayPal support enabled.</td></tr></table></td></tr>
<tr><td>Initiated By</td><td>01.x Online message and language disabled.</td></tr>
<tr><td>Status Response</td><td>11.01 Slide/Insert/Tap Card</td></tr>
<tr><td>DFS Data Index</td><td>0030_0004</td></tr>
<tr><td>Form</td><td>PSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Slide card or choose PayPal”</td></tr>
<tr><td>Form Buttons and IDs</td><td>• "PayPal" - P</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

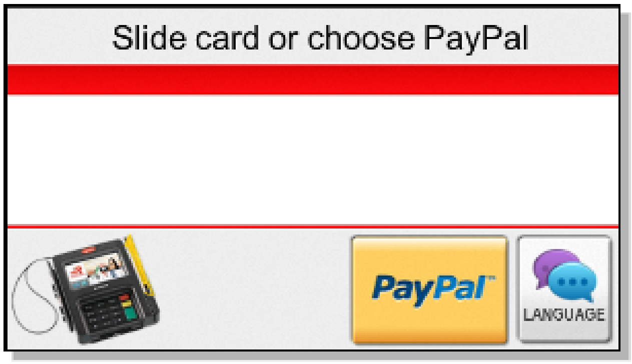
10.7.7.13.2 Card Swipe with PayPal and Language Selection

iPP350



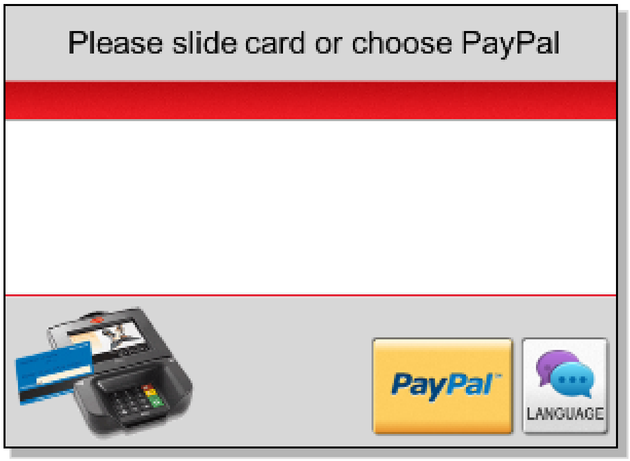
Slide card or choose PayPal

iSC250



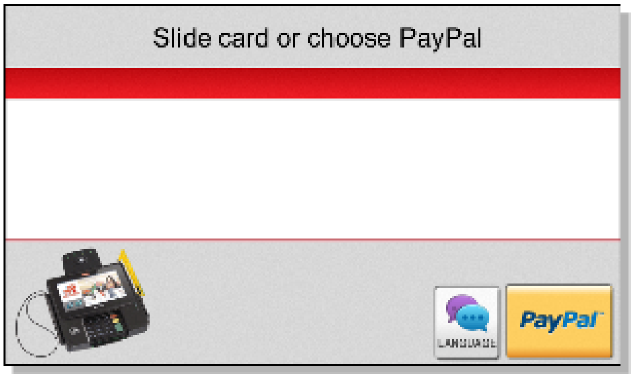
Slide card or choose PayPal

iSC350



Please slide card or choose PayPal

iSC480



Slide card or choose PayPal

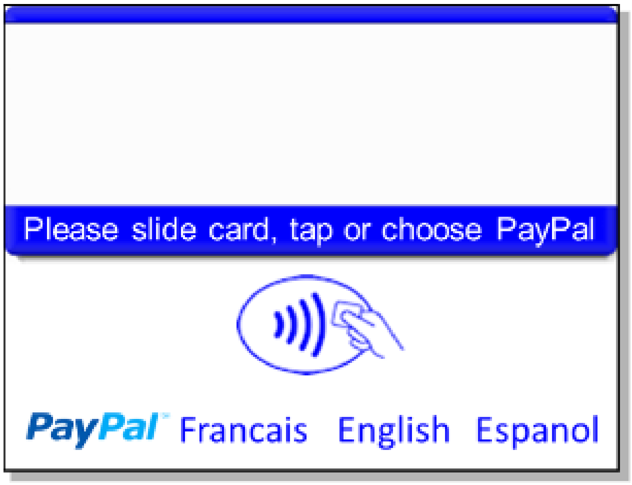
This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

<table>
<tr><td>Description</td><td>Card swipe with PayPal and language. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0040_0006</td><td>1</td><td>PayPal support enabled.</td></tr>
<tr><td>0007_0004</td><td>1</td><td>Combine language with swipe screen.</td></tr>
</table>
</td></tr>
</table>

Initiated By	01.x Online message disabled.
Status Response	11.01 = Slide/Insert/Tap Card
DFS Data Index	0030_0005
Form	PLSWIPE.K3Z
Text	“Slide card or choose PayPal"
Form Buttons and IDs	iSC250/iSC350/iSC480 • "PayPal" - P • "Language" iPP350 • "PayPal" - P • "Francais" - 3 • "English" - 1 • "Espanol" - 2
Terminal Buttons Allowed	Cancel

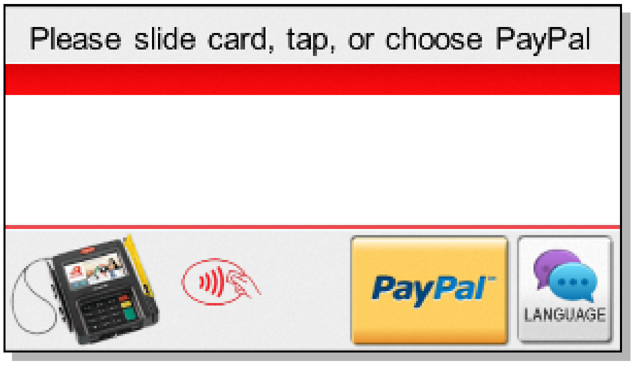
10.7.7.13.3 Contactless Card Swipe with PayPal and Language Selection

iPP350



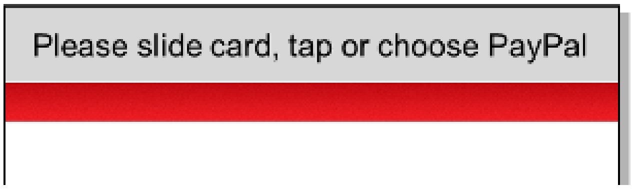
Please slide card, tap or choose PayPal

iSC250



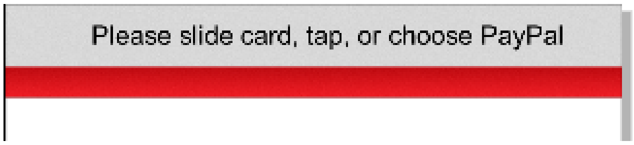
Please slide card, tap, or choose PayPal

iSC350

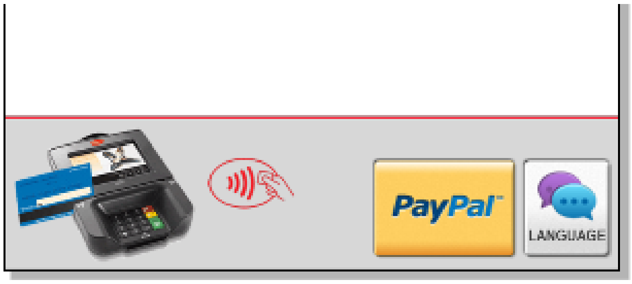


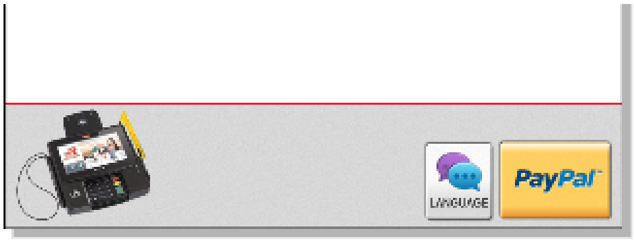
Please slide card, tap or choose PayPal

iSC480



Please slide card, tap, or choose PayPal





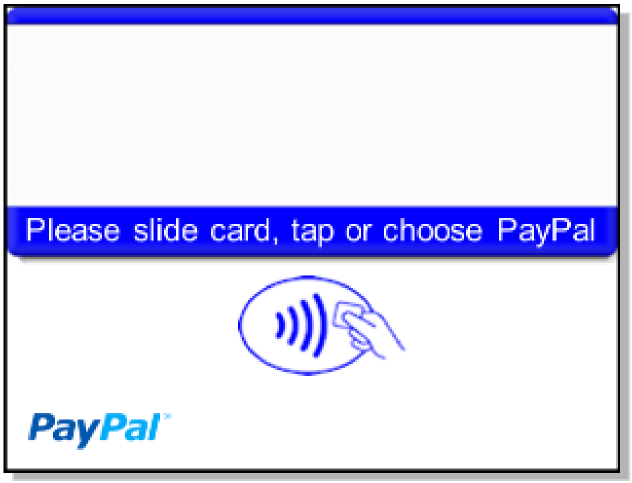
This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

<table>
<tr><td>Description</td><td>Contactless card swipe with PayPal and language. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0008_0001</td><td>1</td><td>Contactless enabled.</td></tr>
<tr><td>0007_0004</td><td>1</td><td>Combine language with swipe screen.</td></tr>
<tr><td>0040_0006</td><td>1</td><td>PayPal support enabled.</td></tr>
</table></td></tr>
<tr><td>Initiated By</td><td>01.x Online message and TBD</td></tr>
<tr><td>Status Response</td><td>11.01 Slide/Insert/Tap Card</td></tr>
<tr><td>DFS Data Index</td><td>0030_0022</td></tr>
<tr><td>Form</td><td>CPLSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Please slide card, tap or choose PayPal”</td></tr>
<tr><td>Form Buttons and IDs</td><td>iPP350
• "PayPal" - P
• "Francais" - 3
• "English" - 1
• "Espanol" - 2
iSC250/iSC350/iSC480
• "PayPal" - P
• "LANGUAGE"</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

10.7.7.13.4 Contactless Card Swipe with PayPal Selection

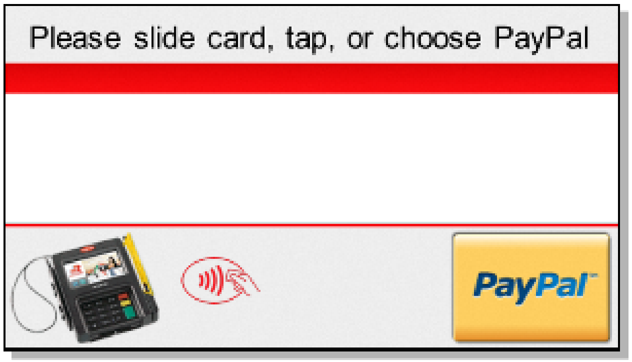
iPP350 iSC250

iPP350



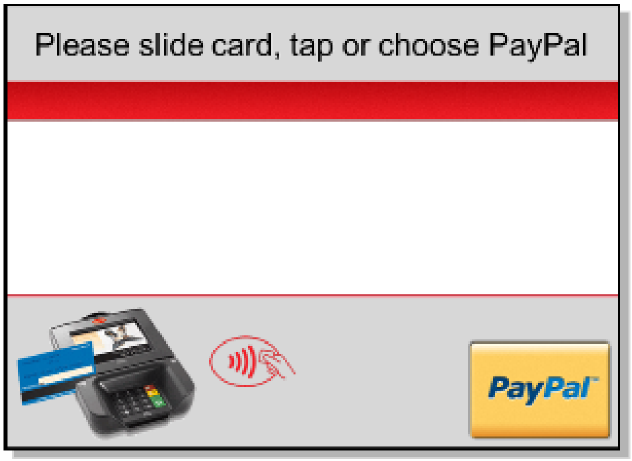
Please slide card, tap or choose PayPal

iSC250



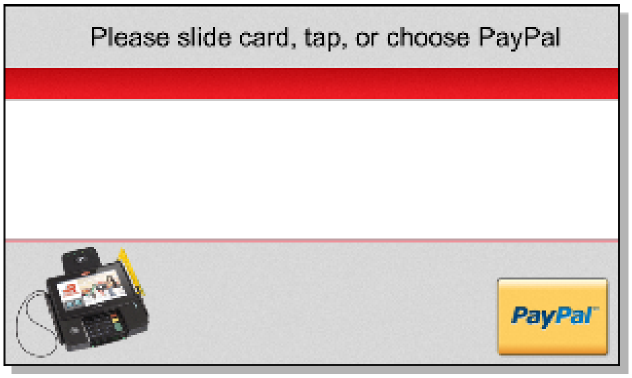
Please slide card, tap, or choose PayPal

iSC350



Please slide card, tap or choose PayPal

iSC480



Please slide card, tap, or choose PayPal

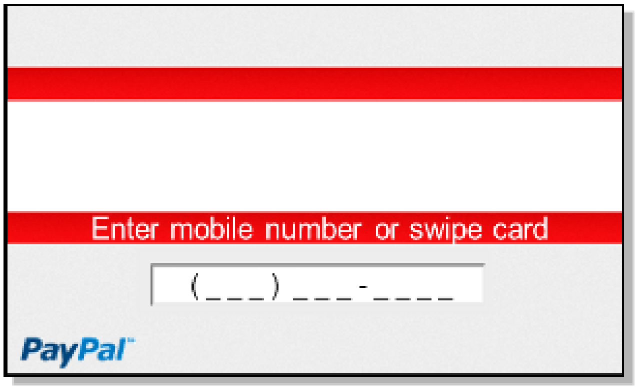
This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

<table>
<tr><td>Description</td><td>Contactless card swipe with PayPal. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0008_0001</td><td>1</td><td>Enable contactless.</td></tr>
<tr><td>0040_0006</td><td>1</td><td>PayPal support enabled.</td></tr>
</table>
</td></tr>
<tr><td>Initiated By</td><td>01.x Online message and language disabled</td></tr>
</table>

Status Response	11.01 Slide/Insert/Tap Card
DFS Data Index	0030_0021
Form	CPSWIPE.K3Z
Text	“Please slide card, tap or choose PayPal”
Form Buttons and IDs	"PayPal" - P
Terminal Buttons Allowed	Cancel

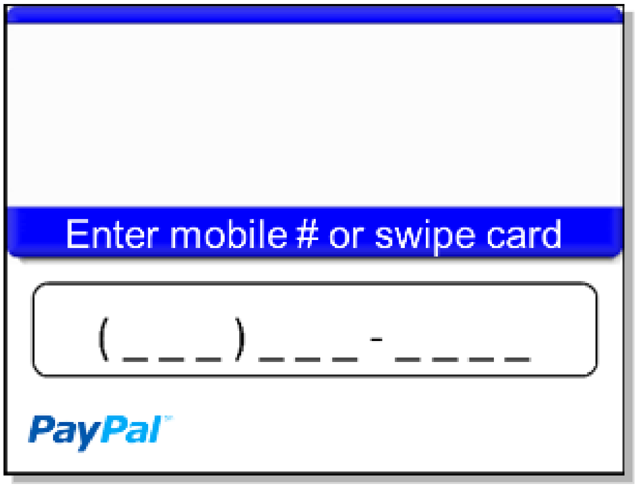
10.7.7.13.5 PayPal Data Input (On-Demand form)

iSC250 & iSC350 & iSC480



Enter mobile number or swipe card

iPP350



Enter mobile # or swipe card

This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

Description	PayPal data input form (zip code or phone number, for instance)
Initiated By	21.x message
Status Response	11.12 Input Accepted/Declined
DFS Data Index	0030_0027
Form	PPALINP.HTM

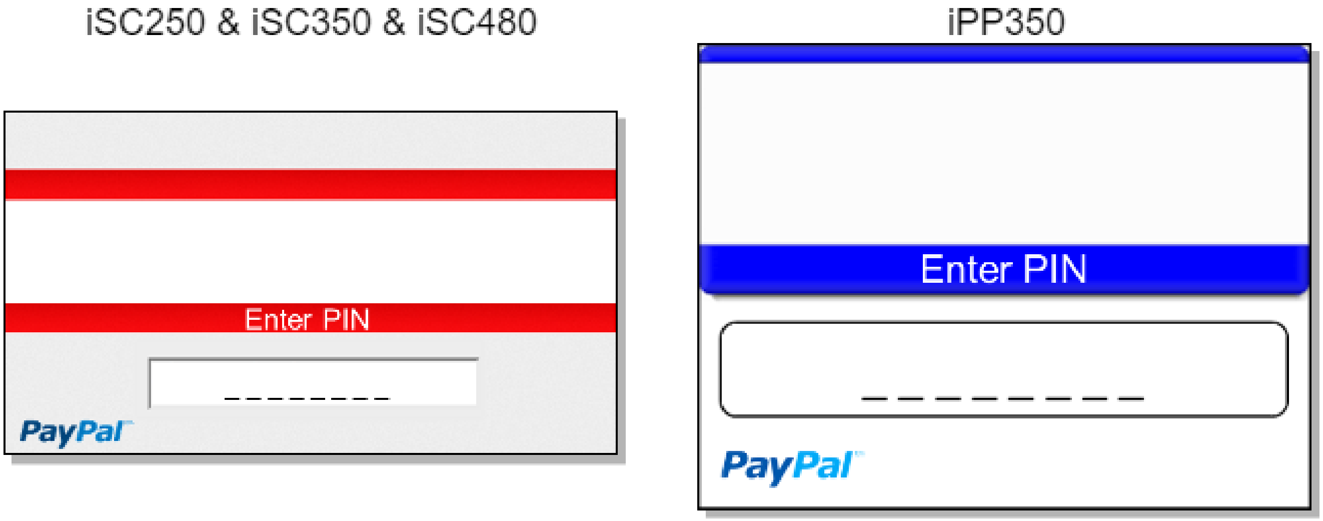
Text	“Enter mobile number or swipe card"
Form Buttons and IDs	N/A
Terminal Buttons Allowed	All

The form file type is .HTM (not .K3Z).

Other form characteristics:

- This form is used with the 21.x message (on-demand).
- The Enter key will end input and will return entered data in a 21.x response message.

10.7.7.13.6 PayPal PIN Entry



This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

Description	PayPal PIN entry form
Initiated By	27.x message

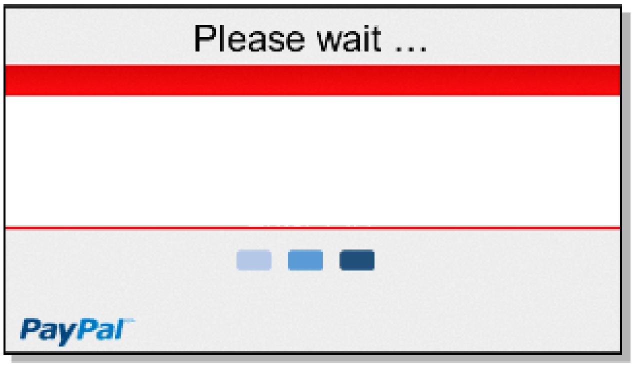
Status Response upon form display	11.03 Enter PIN
Status Response after data entry	11.04 Amount OK?
DFS Data Index	0030_0028
Form	PPALPCAN.HTM
Text	“Enter PIN”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	All

The form file type is .HTM (not .K3Z).

This form can also be used as an on demand form using the 31.x PIN Entry on demand message.

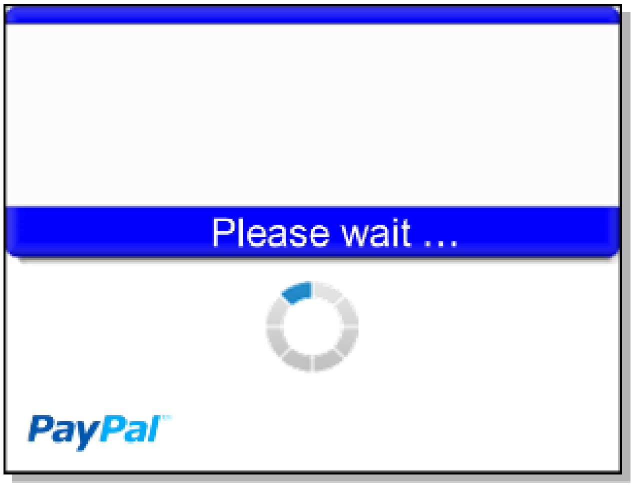
10.7.7.13.7 PayPal Please Wait

iSC250 & iSC350 & iSC480



Please wait …

iPP350



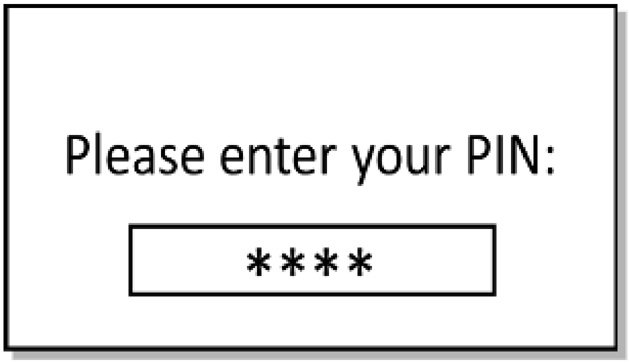
Please wait …

This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

Description	PayPal Please Wait form
Initiated By	27.x message
Status Response upon form display	11.04 Amount OK?
Status Response after data entry	11.06 Approved/Denied
DFS Data Index	0030_0029
Form	PPWAIT.K3Z
Text	“Processing… please wait”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	All

10.7.7.14 PIN Entry

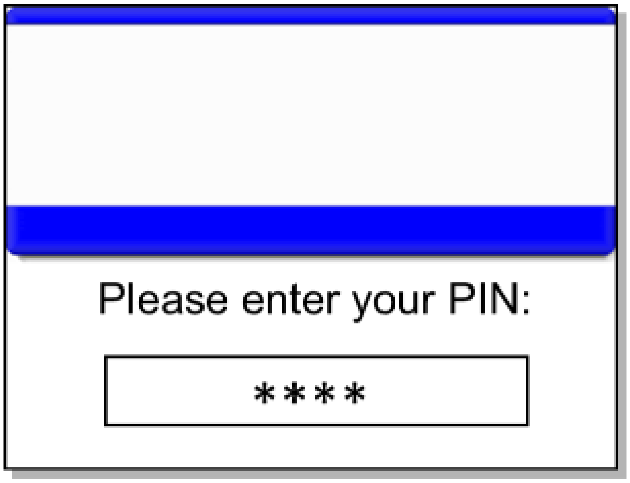
iPP320



Please enter your PIN:

****

iPP350 & iWL250

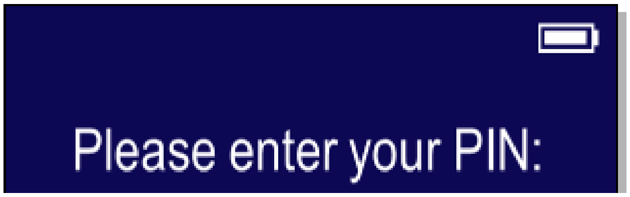


Please enter your PIN:

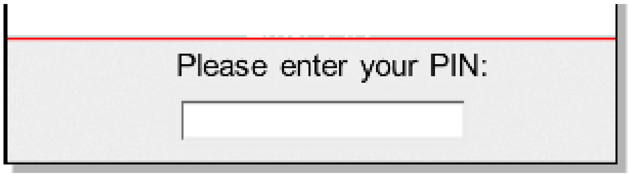
****

iSC250 & iSC350 & iSC480

iSMP & iSMPc



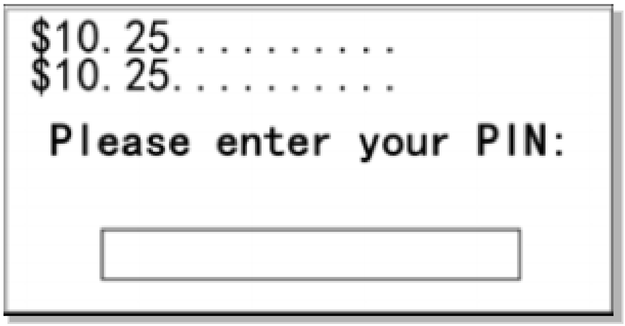
Please enter your PIN:



Please enter your PIN:

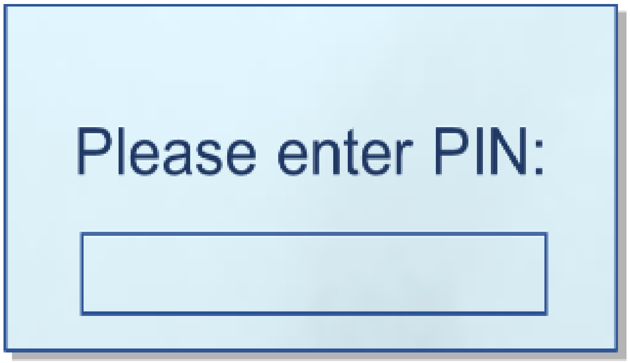
iUP250

iCMP



$10. 25. . . . .
$10. 25. . . . .

Please enter your PIN:



Please enter PIN:

Description	PIN entry
Initiated By	Debit or EBT selection
Status Response	11.03Enter PIN
DFS Data Index	0030_0015
Form	PIN%c.K3Z
Text	“Please enter your PIN:”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	All

If a virtual key is pressed with buttonID:

- 1-9 causes the terminal to display the respective payment type selection screen
- A-P sets the respective payment type and proceeds with the transaction.

10.7.7.14.1 Using the Credit Soft Key on PIN.K3Z Form for PIN Bypass

The PIN.K3Z form supports a "Credit" Button to allow PIN bypass during debit transactions.

Configuration Requirements

To bypass PIN entry, the following flags must be set:

- 0006_0013 = 1 to allow zero-length PIN entry.

- 0006_0013 = 1 for MSR PIN bypass.
- 0011_0001 "On PIN entry cancel" bit = "B" to convert a debit transaction to a credit transaction.

Additionally, the following flags must be set to enable PIN bypass during EMV transactions:

- 0019_0001 = 1 to enable EMV
- 0011_0008 (card type H), which is a copy of the corresponding 0011_0001 parameter set above to be used by EMV debit transactions.
- 0021_000x (Debit AID) Pin Bypass bit = 1

Form Requirements

The PIN entry form loaded to the terminal should contain and display a Credit key that functions as the Enter key as follows:

- For MSR transactions, PIN.K3Z.
- For EMV transactions, PINH.K3Z for example (called when Payment type is set to unused payment type H) or another dedicated payment type.

> Standard flow uses the same configuration parameters as on-demand; however, each payment type can use a custom form. For example, Payment type A uses PINA.K3Z, Payment type B uses PINB.K3Z, and so on.

10.7.7.15 Signature Forms

This section includes the following forms:

- Post-Sign
- Pre-Sign
- Signature (On-Demand)

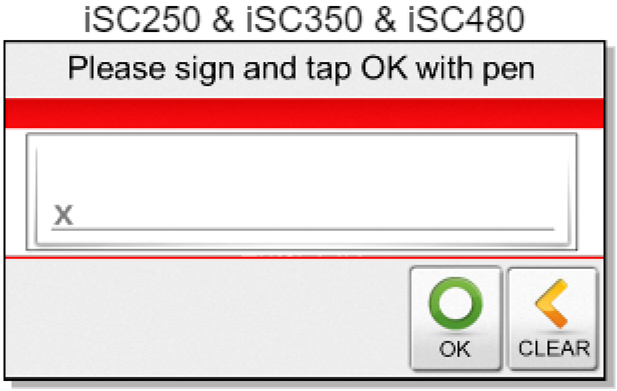
10.7.7.15.1 Post-Sign

Signature capture for the iSC250, iSC350 and iSC480 is now implemented using a generic Pre-Sign form (`PRESIGN.K3Z`). This form displays the "Cancel," "OK" and "Clear" buttons which are fully functional prior to initiating the signature. Once the signature is initiated, this form is replaced by the Post-Sign form (`POSTSIGN.K3Z`) which does not provide the "Cancel" option. The "Cancel" button is cleared from the screen and the "Cancel' key on the terminal keypad will then be processed as a "Clear" key (which will erase the signature). Refer to the below table for a summary of the "Cancel" button and key function for Pre-Sign and Post-Sign forms.

	Button / Keypad Key		**Pre-Sign Form**	**Post-Sign Form**
Signature Request	☒ Cancel	Screen "**Cancel**" Button	Processed as "CANCEL"	Cleared from screen

	Button / Keypad Key		**Pre-Sign Form**	**Post-Sign Form**
		Keypad "**CANCEL**" button	Processed as "CANCEL"	Processed as "CLEAR"

On-Demand signature request does not use the Pre-Sign and Post-Sign forms. Instead, a signature form (`SIGN.K3Z`) is displayed throughout the transaction, and the "Cancel" button and keypad key continue to function as "Cancel."


iSC250 & iSC350 & iSC480

Please sign and tap OK with pen

This form is not available for the iCMP, iSMP, iSMPc, iPP320, iPP350 or iWL250.

Description	Post-sign signature entry.
Initiated By	20.x message or after host approval
Status Response before signature	11.10 "Please sign and tap OK with pen"
Status Response after signature	11.11 "Signature Accepted"
DFS Data Index	0030_0011
Form	POSTSIGN.K3Z
Text	“Please sign and tap OK with pen”
Form Buttons and IDs	- “OK" - "Clear

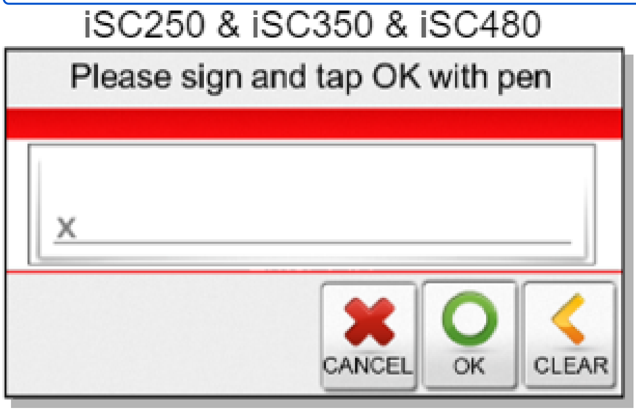
Terminal Buttons Allowed	Enter and Clear.

10.7.7.15.2 Pre-Sign

Signature capture for the iSC250, iSC350 and iSC480 is now implemented using a generic Pre-Sign form (`PRESIGN.K3Z`). This form displays the "Cancel," "OK" and "Clear" buttons which are fully functional prior to initiating the signature. Once the signature is initiated, this form is replaced by the Post-Sign form (`POSTSIGN.K3Z`) which does not provide the "Cancel" option. The "Cancel" button is cleared from the screen and the "Cancel' key on the terminal keypad will then be processed as a "Clear" key (which will erase the signature). Refer to the below table for a summary of the "Cancel" button and key function for Pre-Sign and Post-Sign forms.

	Button / Keypad Key		**Pre-Sign Form**	**Post-Sign Form**
Signature Request		Screen "**Cancel**" Button	Processed as "CANCEL"	Cleared from screen
		Keypad "**CANCEL**" button	Processed as "CANCEL"	Processed as "CLEAR"

On-Demand signature request does not use the Pre-Sign and Post-Sign forms. Instead, a signature form (`SIGN.K3Z`) is displayed throughout the transaction, and the "Cancel" button and keypad key continue to function as "Cancel.



This form is not available for the iCMP, iSMP, iSMPc, iPP320, iPP350 or iWL250.

Description	Signature entry - generic pre-sign form.
Initiated By	20.x message or after host approval
Status Response before signature	11.10 "Please sign and tap OK with pen"
Status Response after signature	11.11 "Signature Accepted"
DFS Data Index	0030_0040
Form	PRESIGN.K3Z
Text	“Please sign and tap OK with pen”
Form Buttons and IDs	• “OK" • "Clear" • "Cancel" The "CANCEL" button is displayed until the signature has started. Once the signature is initiated, the Pre-Sign Form is replaced by the Post-Sign form and the "CANCEL" button is no longer displayed. This does not apply to On-Demand signature request, where the "Cancel" button will continue to be displayed and functional during the signature process.
Terminal Buttons Allowed	Cancel, Clear, and Enter as stated above.

10.7.7.15.3 Signature (On-Demand)

Signature capture for the iSC250, iSC350 and iSC480 is implemented using a generic signature form `(SIGN.K3Z).` This form's font size has been changed to support a variable number of prompt lines. The default font size differs by model to accommodate their individual screen sizes. By default, for example, the iSC350 supports 4 prompt lines. Changing the font size in form `SIGN.K3Z` changes the number of prompts lines that can be displayed.

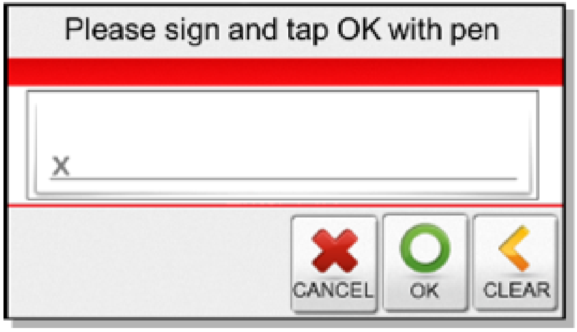
This form initially displays the "Cancel," "OK" and "Clear" buttons which are fully functional prior to initiating the signature. Once the signature is initiated and a predetermined amount of data from the signature has been captured, the "Cancel" button is cleared (hidden) from the screen and the "Cancel" key on the terminal keypad will then be processed as a "Clear" key (which will erase the signature). The following table illustrates this process. Note that this does not apply to On-Demand signature. For On-Demand signature, all buttons remain visible and functional throughout the signature capture process.

Cancel Button and Physical Key Function During Signature Capture Process

	Button / Keypad Key		**Pre-Signature**	**Signature Initiated**
Signature Capture		Screen "**Cancel**" Button	Processed as "CANCEL"	The “Cancel” button will become hidden after signature is started.
		Keypad "**CANCEL**" button	Processed as "CANCEL"	Current unsubmitted signature image is cleared from screen as if the physical "CLEAR" button were pressed.

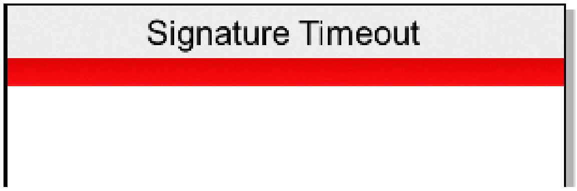
For On-Demand signature, the "Cancel" button continues to be displayed during the signature process. It is not hidden, and it remains fully functional.

iSC250 & iSC350 & iSC480
Signature Form

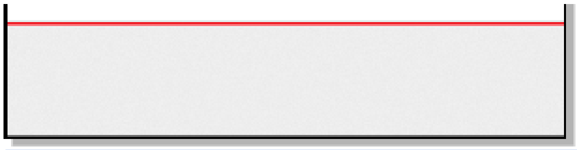


Please sign and tap OK with pen

Signature Timeout Form



Signature Timeout



> This form is not available for the iCMP, iSMP, iSMPc, iPP320, iPP350 or iWL250.

Description	Signature entry - generic pre-sign form.
Initiated By	20.x message or after host approval
Status Response before signature	11.10 "Please sign and tap OK with pen"
Status Response after signature	11.11 "Signature Accepted"
DFS Data Index	0030_0041
Form	SIGN.K3Z
Text	“Please sign and tap OK with pen” is initially displayed. "Signature Timeout" is displayed upon signature start timeout if parameter '0009_0004' is set to a value other than '0'.
Form Buttons and IDs	• “OK" • "Clear" • "Cancel" The "CANCEL" button is displayed until the signature has started. Once the signature is initiated, this button is then hidden.
Terminal Buttons Allowed	Cancel, Clear, and Enter as stated above.

10.7.7.16 Smart Card (SMC) and EMV Forms

This section includes the following forms:

- Contactless Smart Card (EMV) and Swipe
- Contactless Smart Card (EMV) and Swipe with Language Selection

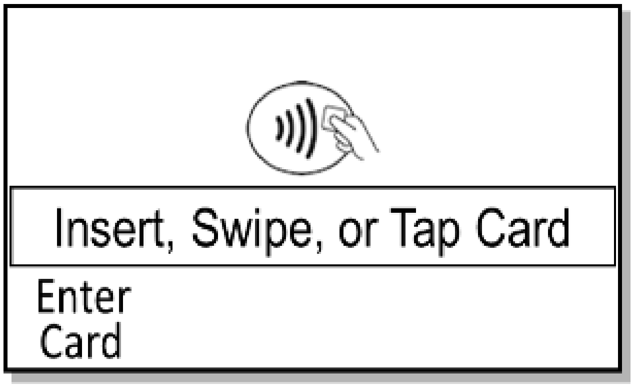
- Smart Card (EMV) and Swipe
- Smart Card (EMV) and Swipe with Language Selection
- Smart Card (EMV) Please Wait
- Smart Card (EMV) Confirm Application
- Smart Card (EMV) Confirm Amount
- Smart Card (EMV) Application Selection
- Smart Card (EMV) Language Selection

The term "Smart Card" is synonymous with the term "EMV Card." These terms refer to embedded chip cards which can be used in contact or contactless EMV transactions. Smart cards are inserted in the terminal through a separate card reader instead of being swiped. When inserted, a connection is made through a set of contacts visible on the front of the card which powers the embedded chip and enables it to communicate directly with the terminal. Hence the term "insert" applies to smart cards.

A contactless smart card features an antenna embedded in the card. When the card is tapped, the RF field generated by the contactless card reader powers the embedded chip and enables communications with the terminal.

10.7.7.16.1 Contactless Smart Card (EMV) and Swipe

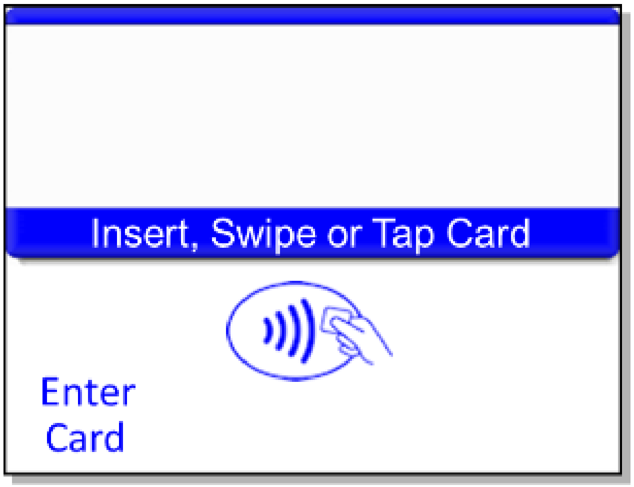
iPP320



Insert, Swipe, or Tap Card

Enter
Card

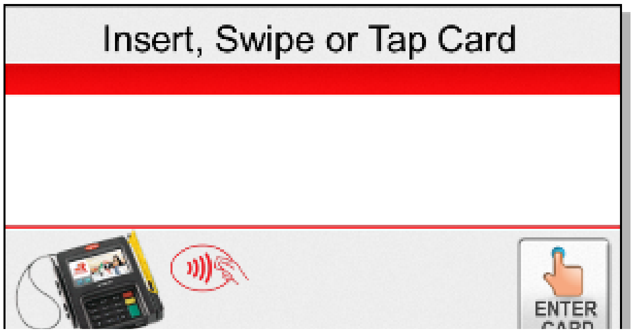
iPP350 & iWL250



Insert, Swipe or Tap Card

Enter
Card

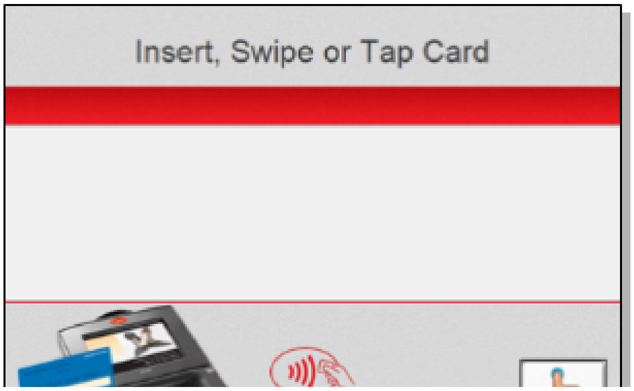
iSC250



Insert, Swipe or Tap Card

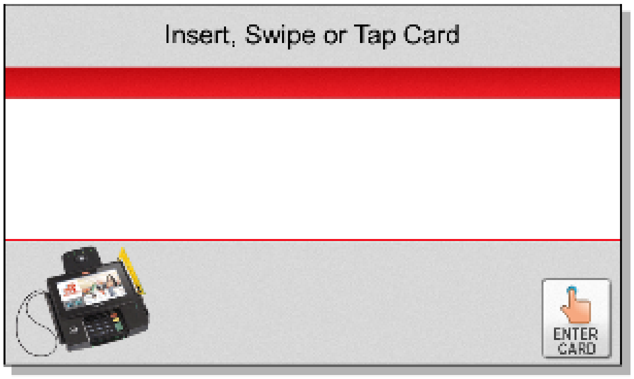
ENTER
CARD

iSC350

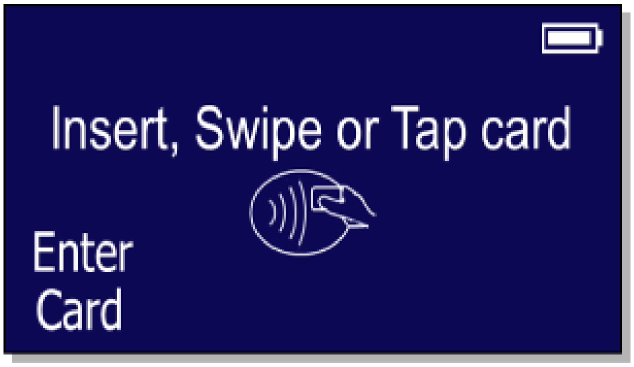


Insert, Swipe or Tap Card

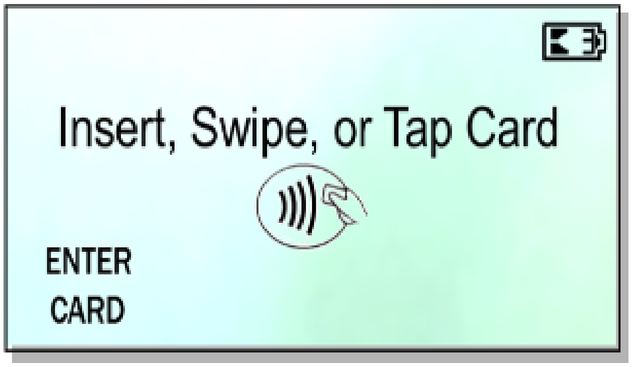
iSC480


Insert, Swipe or Tap Card

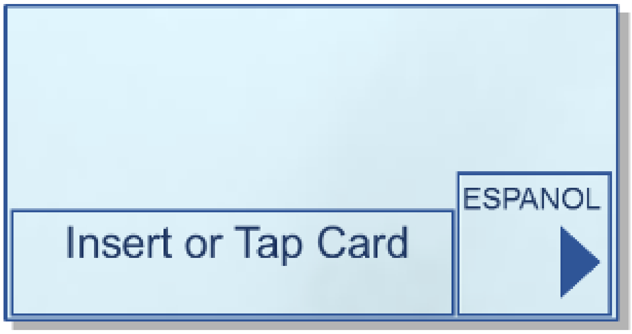
iSMP & iSMPc


Insert, Swipe or Tap card

iCMP


Insert, Swipe, or Tap Card

iUP250


Insert or Tap Card

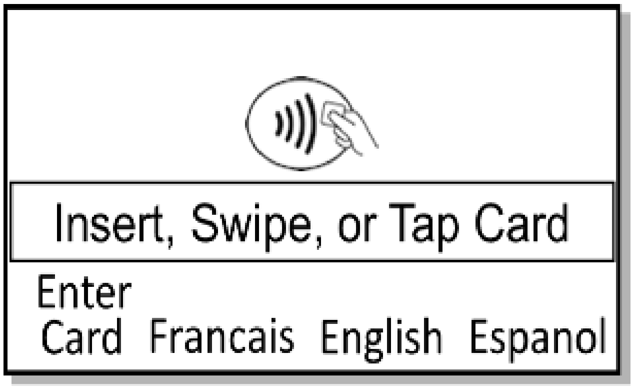
<table>
<tr><td>Description</td><td>Contactless EMV and swipe. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0019_0001</td><td>1</td><td>Enable EMV transactions.</td></tr>
<tr><td>0007_0004</td><td>1</td><td>Combine language with swipe screen.</td></tr>
<tr><td>0007_0005</td><td>0</td><td>Swipe card first, then select payment type.</td></tr>
<tr><td>0007_0029</td><td>1</td><td>Display "Enter Card" button and prompt for card number, expiration date and CVV.</td></tr>
<tr><td>0008_0001</td><td>9</td><td>Enable contactless for EMV.</td></tr>
</table>

This parameter controls the display of the "Enter Card" button on the card swipe screen and enables cardholder prompts for manual entry. When not set to '0', the "Enter Card" button will be displayed with prompt options. Additional settings for parameter '0007_0029' used during manual entry include:
<ul>
<li>2 = Display "Enter Card" button and prompt for card number and expiration date (no CVV).</li>
<li>3 = Display "Enter Card" button and prompt for card number and CVV (no expiration date).</li>
<li>4 = Display "Enter Card" button and prompt for card number (no expiration date or CVV).</li>
</ul>
</td></tr>
<tr><td>Initiated By</td><td>01.x Online message and language disabled</td></tr>
<tr><td>Status Response</td><td>11.01 Slide/Insert/Tap Card</td></tr>
<tr><td>DFS Data Index</td><td>0030_0038</td></tr>
<tr><td>Form</td><td>CESWIPE.K3Z</td></tr>
<tr><td>Text</td><td>
<ul>
<li>"Insert or Tap Card", iUP250 only.</li>
<li>“Insert, Swipe or Tap Card”, all other terminals.</li>
</ul>
</td></tr>
</table>

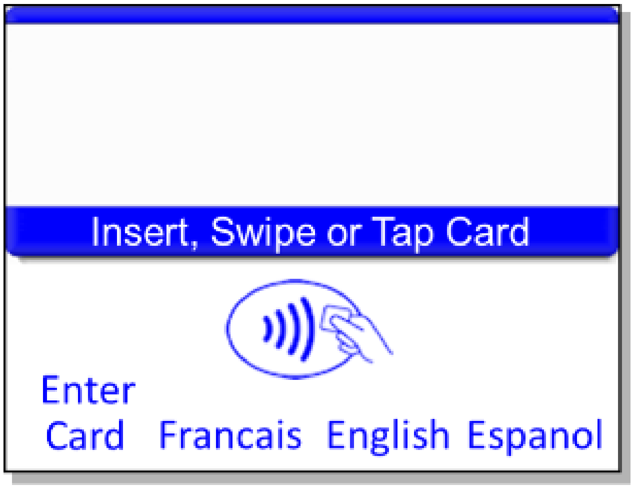
Form Buttons and IDs	• "ESPANOL", iUP250 only. • “Enter Card” – M, all other terminals.

10.7.7.16.2 Contactless Smart Card (EMV) and Swipe with Language Selection

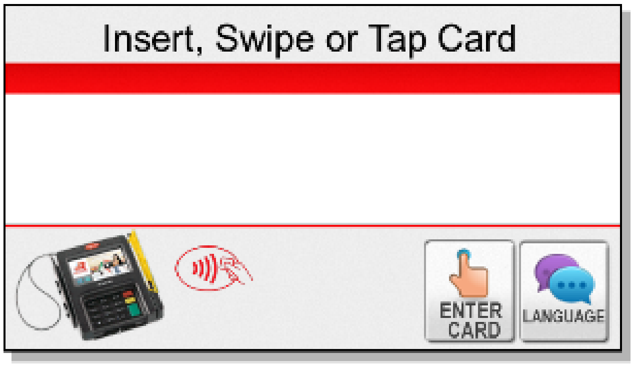
iPP320


Insert, Swipe, or Tap Card

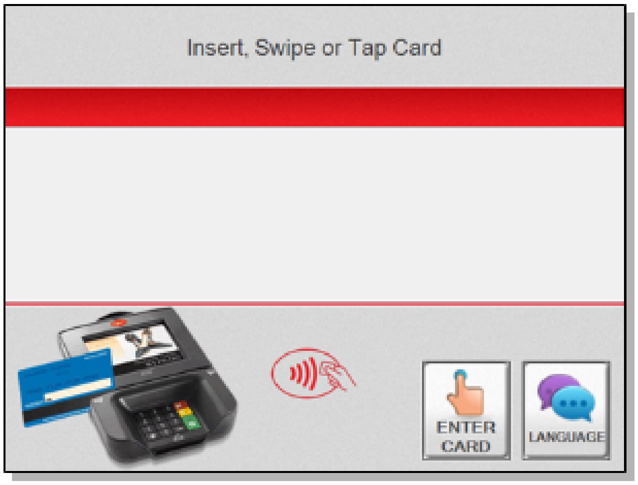
iPP350 & iWL250


Insert, Swipe or Tap Card

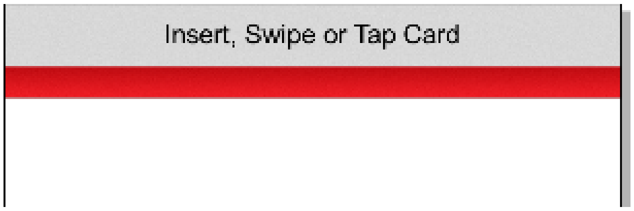
iSC250


Insert, Swipe or Tap Card

iSC350

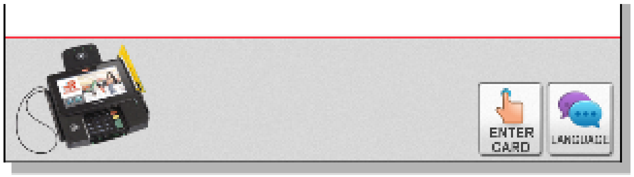

Insert, Swipe or Tap Card

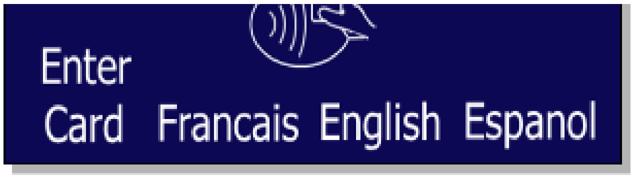
iSC480


Insert, Swipe or Tap Card

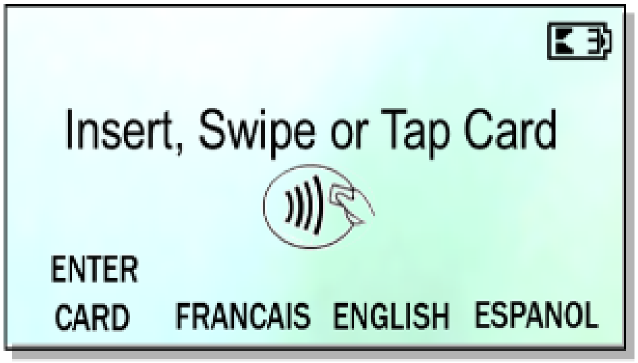
iSMP & iSMPc

Insert, Swipe or Tap card





iCMP

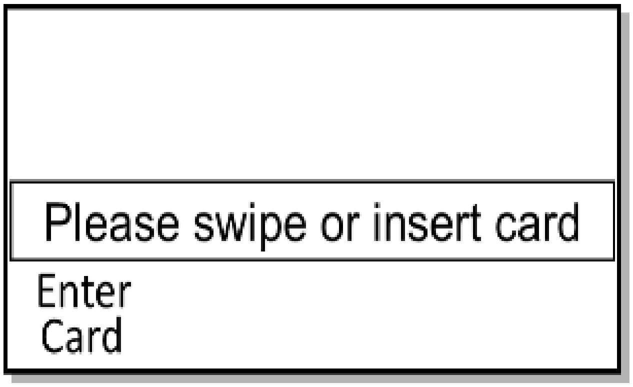

Insert, Swipe or Tap Card

<table>
<tr><td>Description</td><td>Contactless EMV and Swipe with Language. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0019_0001</td><td>1</td><td>Enable EMV transactions.</td></tr>
<tr><td>0008_0001</td><td>1</td><td>Enable contactless.</td></tr>
<tr><td>0007_0004</td><td>1</td><td>Combine language with swipe screen.</td></tr>
<tr><td>0007_0005</td><td>0</td><td>Swipe card first, then select payment type.</td></tr>
<tr><td>0007_0029</td><td>1</td><td>Display "Enter Card" button and prompt for card number, expiration date and CVV.</td></tr>
<tr><td>0008_0001</td><td>9</td><td>Enable contactless for EMV.</td></tr>
</table>
This parameter controls the display of the "Enter Card" button on the card swipe screen and enables cardholder prompts for manual entry. When not set to '0', the "Enter Card" button will be displayed with prompt options. Additional settings for parameter '0007_0029' used during manual entry include:
• 2 = Display "Enter Card" button and prompt for card number and expiration date (no CVV).
• 3 = Display "Enter Card" button and prompt for card number and CVV (no expiration date).
• 4 = Display "Enter Card" button and prompt for card number (no expiration date or CVV).</td></tr>
<tr><td>Initiated By</td><td>01.x Online message</td></tr>
<tr><td>Status Response</td><td>11.01 Slide/Insert/Tap Card</td></tr>
<tr><td>DFS Data Index</td><td>0030_0039</td></tr>
<tr><td>Form</td><td>CELSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Insert, Swipe or Tap Card”</td></tr>
</table>

Form Buttons and IDs	iPP320/iPP350/iWL250/iMP350/iMP352 • “Enter Card” – M • "Francais" • "English" • "Espanol" iSC250/iSC350/iSC480 • "ENTER CARD" - M • "LANGUAGE" - L

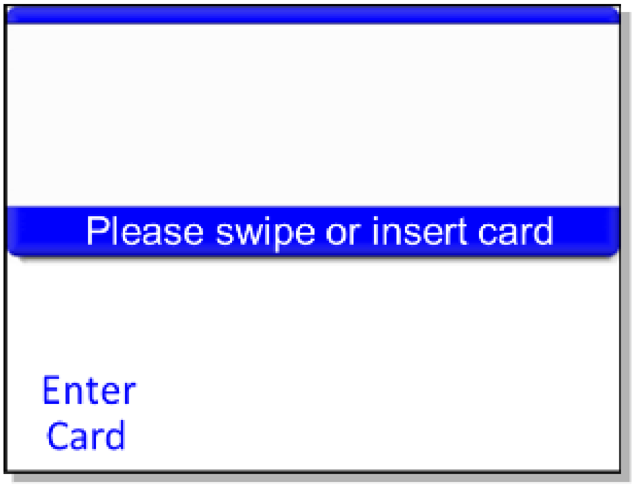
10.7.7.16.3 Smart Card (EMV) and Swipe

iPP320



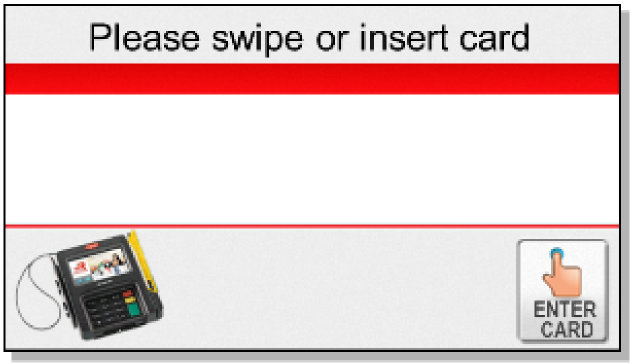
Please swipe or insert card

iPP350 & iWL250



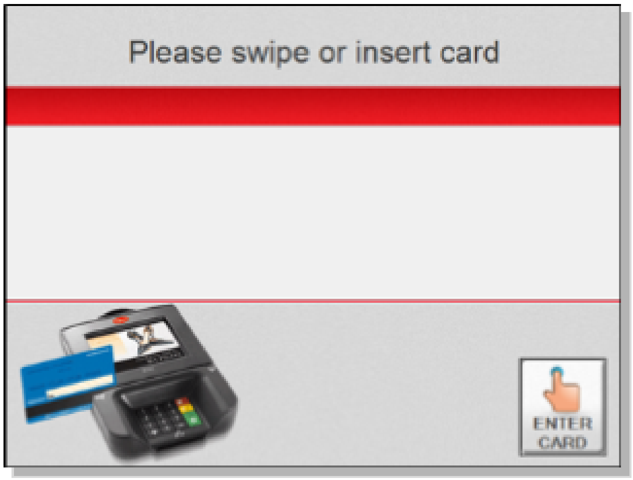
Please swipe or insert card

iSC250



Please swipe or insert card

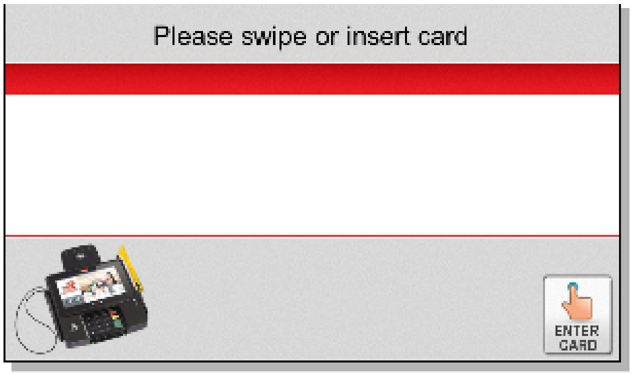
iSC350



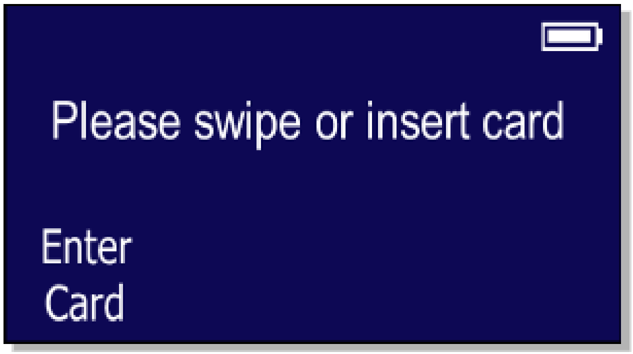
Please swipe or insert card

iSC480

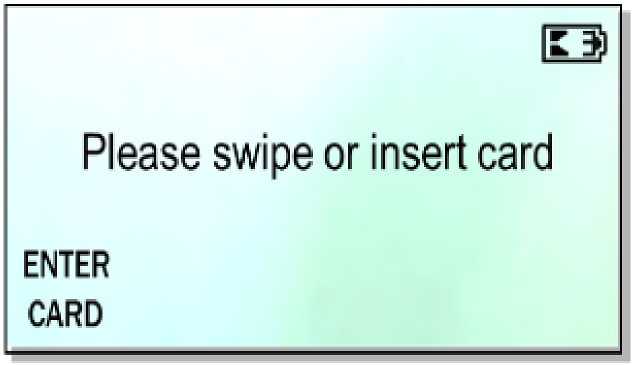
iSMP & iSMPc

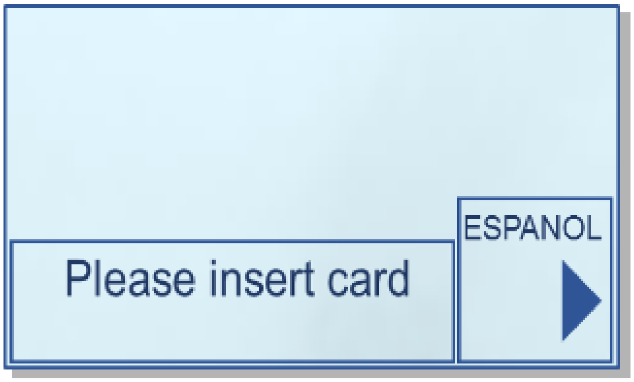

Please swipe or insert card

iCMP


Please swipe or insert card

iUP250


Please swipe or insert card

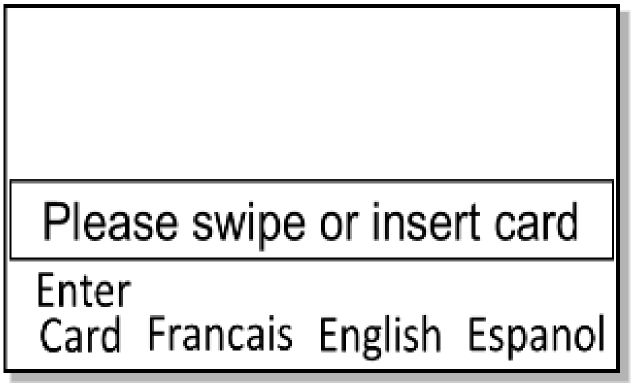

Please insert card

<table>
<tr><td>Description</td><td>EMV card and swipe. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0019_0001</td><td>1</td><td>Enable EMV transactions.</td></tr>
<tr><td>0007_0005</td><td>0</td><td>Swipe card first, then select payment type.</td></tr>
<tr><td>0007_0029</td><td>1</td><td>Display "Enter Card" button and prompt for card number, expiration date and CVV.
This parameter controls the display of the "Enter Card" button on the card swipe screen and enables cardholder prompts for manual entry. When not set to '0', the "Enter Card" button will be displayed with prompt options. Additional settings for parameter '0007_0029' used during manual entry include:<ul><li>2 = Display "Enter Card" button and prompt for card number and expiration date (no CVV).</li><li>3 = Display "Enter Card" button and prompt for card number and CVV (no expiration date).</li><li>4 = Display "Enter Card" button and prompt for card number (no expiration date or CVV).</li></ul></td></tr>
</table></td></tr>
<tr><td>Initiated By</td><td>01.x Online message and language disabled</td></tr>
<tr><td>Status Response</td><td>11.01 Slide/Insert/Tap Card</td></tr>
<tr><td>DFS Data Index</td><td>0030_0036</td></tr>
<tr><td>Form</td><td>ESWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Please swipe or insert card"</td></tr>
</table>

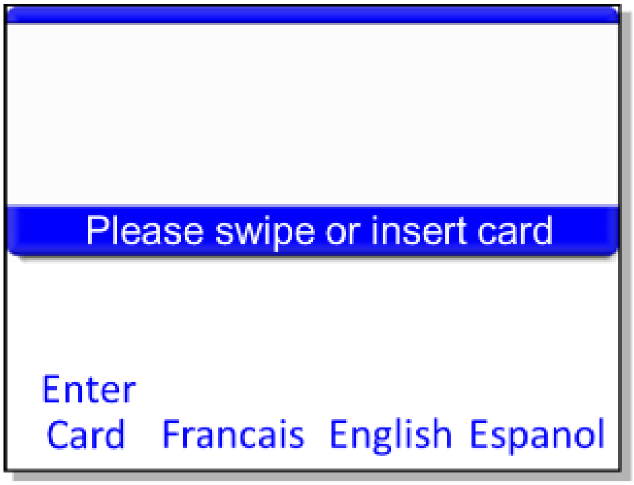
Form Buttons and IDs	"Enter Card" – M
Terminal Buttons Allowed	Cancel

10.7.7.16.4 Smart Card (EMV) and Swipe with Language Selection

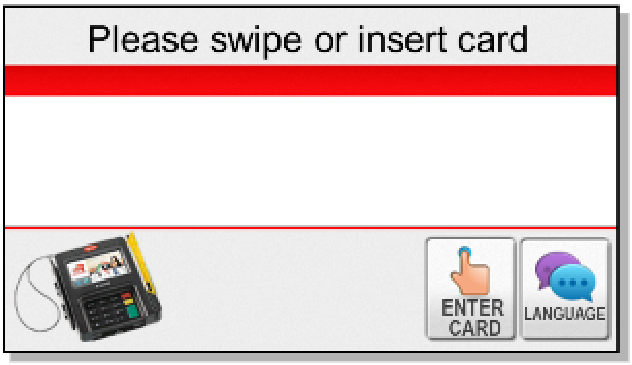
iPP320


Please swipe or insert card

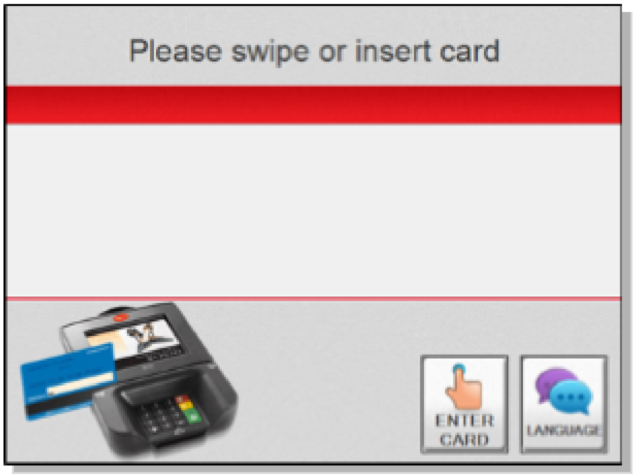
iPP350 & iWL250


Please swipe or insert card

iSC250

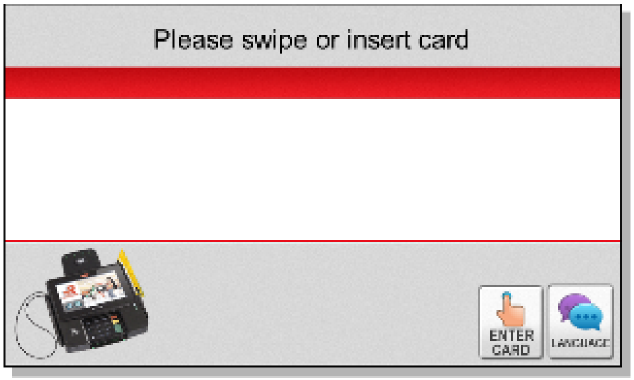

Please swipe or insert card

iSC350


Please swipe or insert card

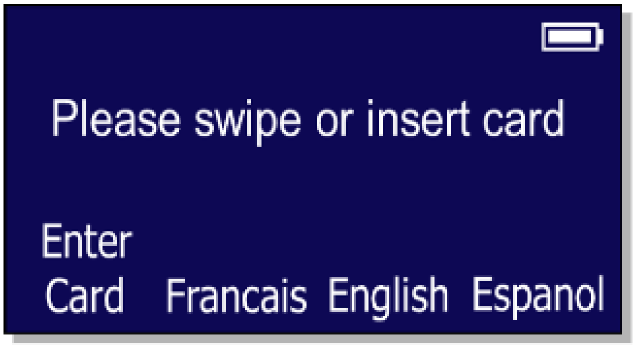
iSC480

iSMP & iSMPc



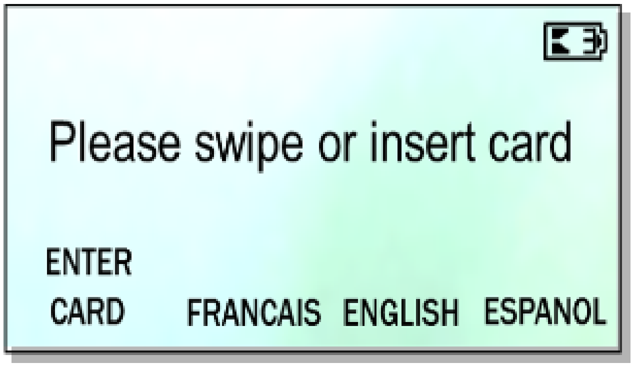
Please swipe or insert card

iCMP

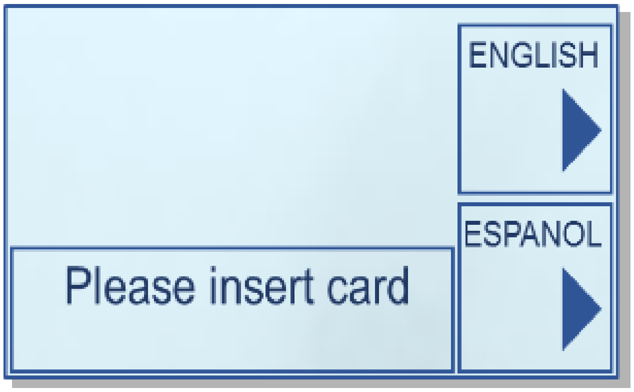


Please swipe or insert card

iUP250



Please swipe or insert card



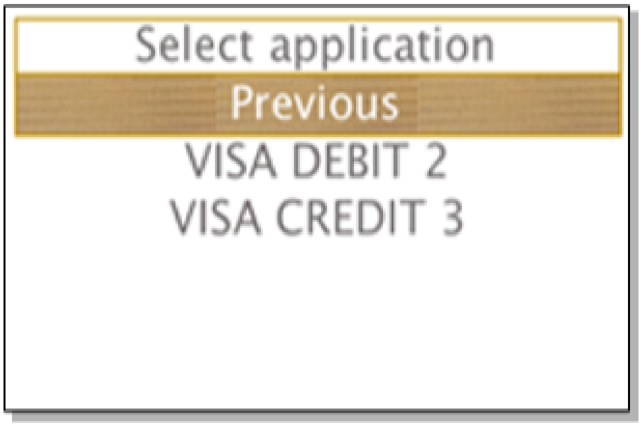
Please insert card

<table>
<tr><td>Description</td><td>EMV card swipe with language buttons. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0019_0001</td><td>1</td><td>Enable EMV transactions.</td></tr>
<tr><td>0007_0004</td><td>1</td><td>Combine language with swipe screen.</td></tr>
<tr><td>0007_0005</td><td>0</td><td>Swipe card first, then select payment type.</td></tr>
<tr><td>0007_0029</td><td>1</td><td>Display "Enter Card" button and prompt for card number, expiration date and CVV.</td></tr>
</table>
<p>This parameter controls the display of the "Enter Card" button on the card swipe screen and enables cardholder prompts for manual entry. When not set to '0', the "Enter Card" button will be displayed with prompt options. Additional settings for parameter '0007_0029' used during manual entry include:</p>
<ul>
<li>2 = Display "Enter Card" button and prompt for card number and expiration date (no CVV).</li>
<li>3 = Display "Enter Card" button and prompt for card number and CVV (no expiration date).</li>
<li>4 = Display "Enter Card" button and prompt for card number (no expiration date or CVV).</li>
</ul>
</td></tr>
<tr><td>Initiated By</td><td>01.x Online message</td></tr>
<tr><td>Status Response</td><td>11.01 = Slide/Insert/Tap Card</td></tr>
<tr><td>DFS Data Index</td><td>0030_0037</td></tr>
<tr><td>Form</td><td>ELSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Please swipe or inert card"</td></tr>
</table>

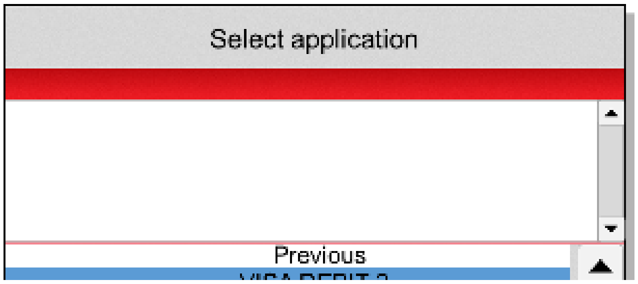
Form Buttons and IDs	iPP320/iPP350/iWL250 • “Enter Card” – M • "Francais" • "English" • "Espanol" iUP250 • "English" • "Espanol" iSC250/iSC350/iSC480 • "ENTER CARD" • "LANGUAGE"
Terminal Buttons Allowed	Cancel

10.7.7.16.5 Smart Card (EMV) Application Selection

iPP320



iSC250 & iSC350 & iSC480



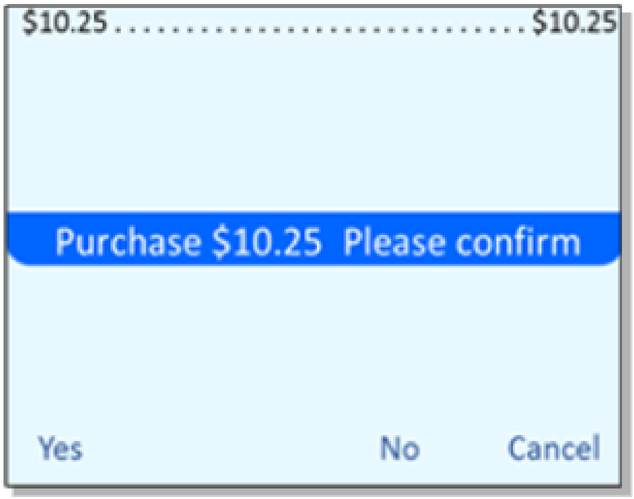
MENU Form Description

Specification	Description
DFS Data Index	0030_0034
Form	MENU.K3Z The MENU.K3Z form is used for menu selection when parameter '0019_0003' = '1' or '3'.
Text	<?ivEMV_DISPLAY_MESSAGE?>
Form Buttons	Scroll bar is shown on screen when number of available applications exceeds page display capacity. • For iSC250, iSC350, and iSC480,the3 cardholder can use the soft buttons to scroll through the list of available applications. • For all other terminals, the Function keys (F2 and F3) are used to scroll through the list of available applications.

10.7.7.16.6 Smart Card (EMV) Confirm Amount

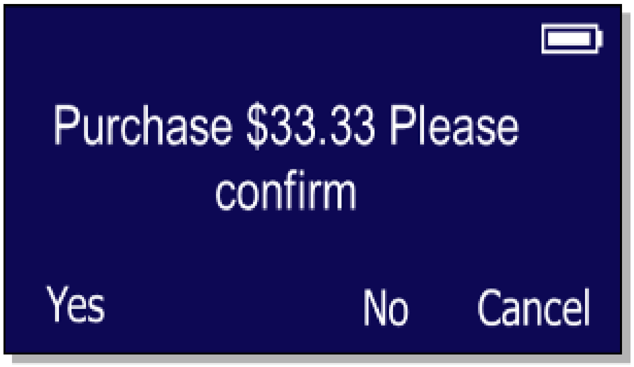
The EMVAMT form prompts the cardholder to confirm the transaction amount. The following table illustrates the form display for iPP series and iSC series Ingenico devices.

iPP350 & iWL250



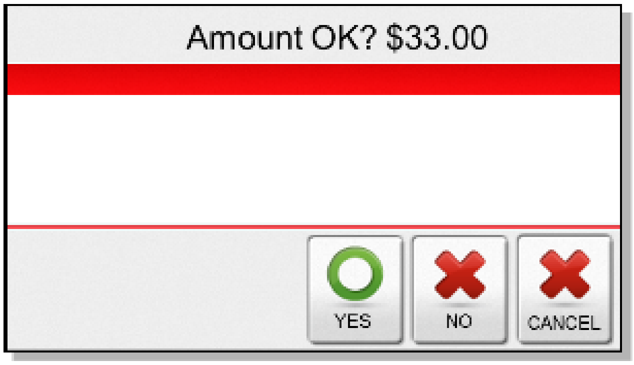
Purchase $10.25 Please confirm

iSMP & iSMPc



Purchase $33.33 Please confirm

iSC250 & iSC350 & iSC480


Amount OK? $33.00

The following table provides a description of the EMVAMT form.

EMVAMT Form Description

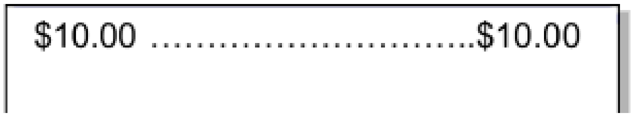
Specification	Description
DFS Data Index	0030_0033
Form	ECONFIRM.K3Z
Text	<?ivEMV_DISPLAY_MESSAGE?>
Form Buttons	**iPP320 • iPP350 • iWL250** Button / Button ID Yes / btn1 No / btn2 Cancel / btn3 **iSC250 • iSC350 • iSC480** Button / Button ID Yes / btn1 No / btn2 Cancel / btn3

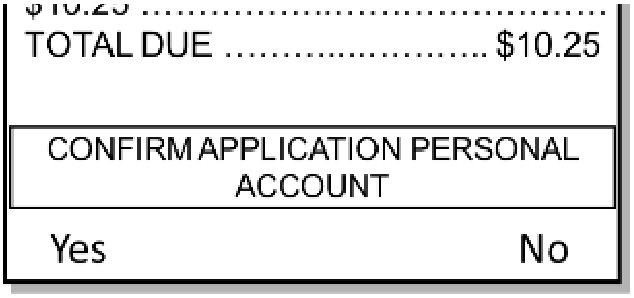
10.7.7.16.7 Smart Card (EMV) Confirm Application

iPP320

$10.25

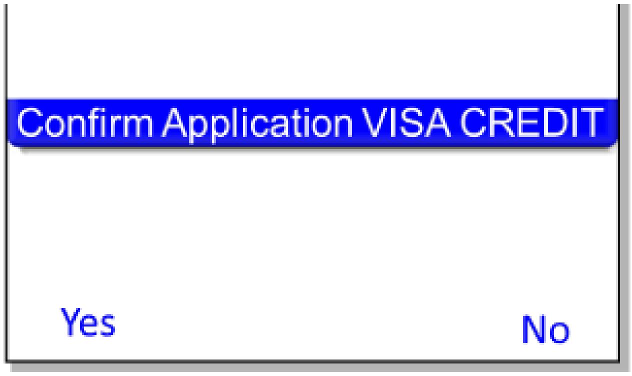
iPP350 & iWL250


$10.00 ....$10.00



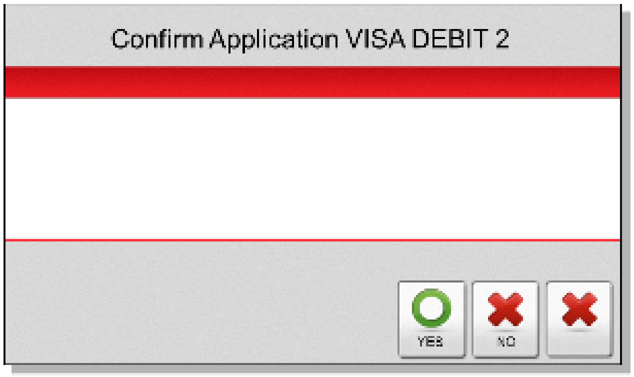
TOTAL DUE $10.25

CONFIRM APPLICATION PERSONAL ACCOUNT



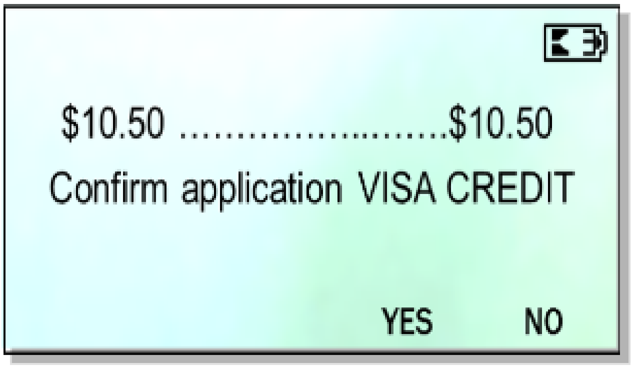
Confirm Application VISA CREDIT

iSC250 & iSC350 & iSC480



Confirm Application VISA DEBIT 2

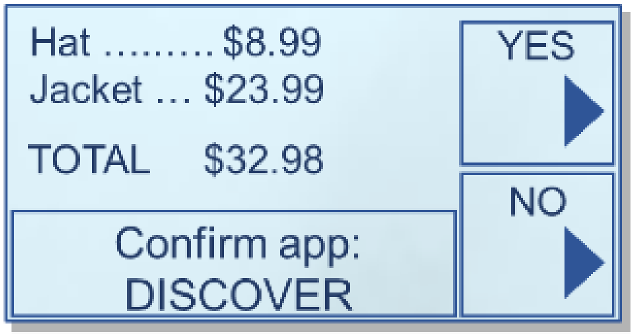
iCMP



$10.50 ....$10.50

Confirm application VISA CREDIT

iUP250



Hat $8.99
Jacket ... $23.99

TOTAL $32.98

Confirm app: DISCOVER

The following table provides a description of the EMVAPP form.

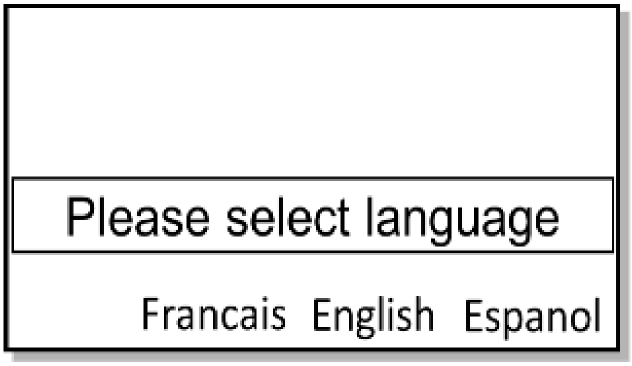
EMVAPP Form Description

Specification	Description
DFS Data Index	0030_0033

<table>
<tr><th>Specification</th><th>Description</th></tr>
<tr><td>Form</td><td>ECONFIRM.K3Z
The ECONFIRM.K3Z form is used for confirming the amount as well as the application <b>when</b> parameter '0019_0003' = '1' or '3' for confirmation select, <b>and</b> also to confirm the application <b>after</b> menu selection when parameter '0019_0003' = '0'. The MENU.K3Z form (parameter '0030_0034') is used for menu selection when parameter '0019_0003' = '0' or '2'.</td></tr>
<tr><td>Text</td><td>"Confirm Application %s"</td></tr>
<tr><td>Form Buttons</td><td><b>iPP320 • iPP350 • iWL250 • iUP250</b>
<table>
<tr><th>Button</th><th>Button ID</th></tr>
<tr><td>YES</td><td>btn1</td></tr>
<tr><td>NO</td><td>btn2</td></tr>
</table>
<b>iSC250 • iSC350 • iSC480</b>
<table>
<tr><th>Button</th><th>Button ID</th></tr>
<tr><td>Yes</td><td>btn1</td></tr>
<tr><td>No</td><td>btn2</td></tr>
<tr><td>Cancel</td><td>btn3</td></tr>
</table></td></tr>
</table>

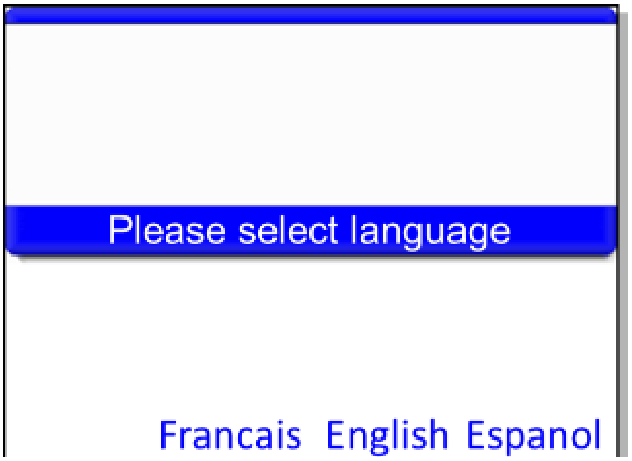
10.7.7.16.8 Smart Card (EMV) Language Selection

iPP320



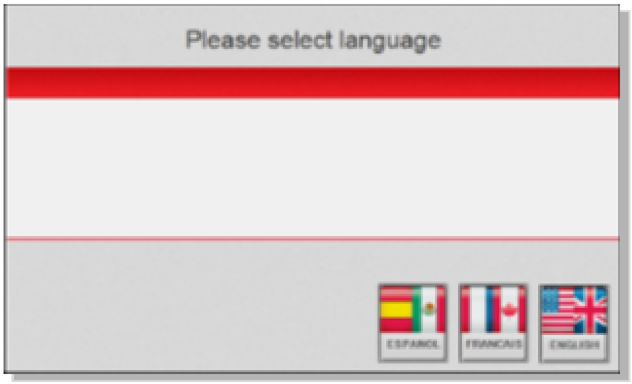
Please select language

iPP350 & iWL250



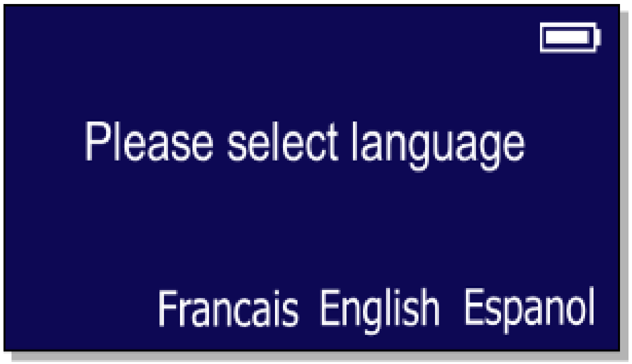
Please select language

iSC250 & iSC350 & iSC480



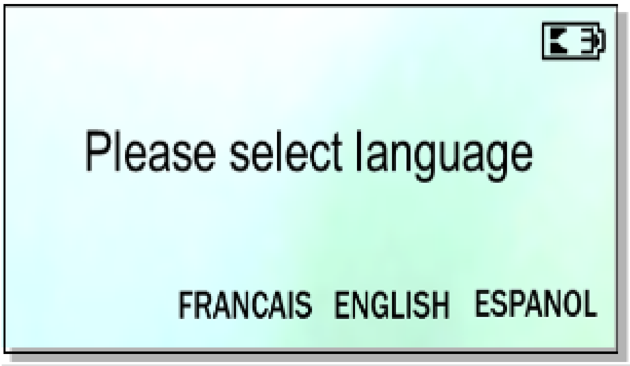
Please select language

iSMP & iSMPc



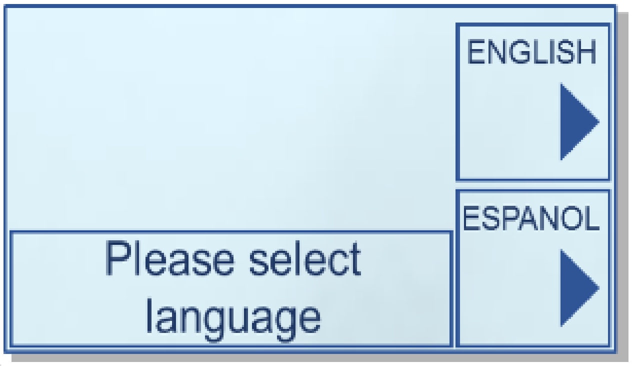
Please select language

iCMP



Please select language

iUP250

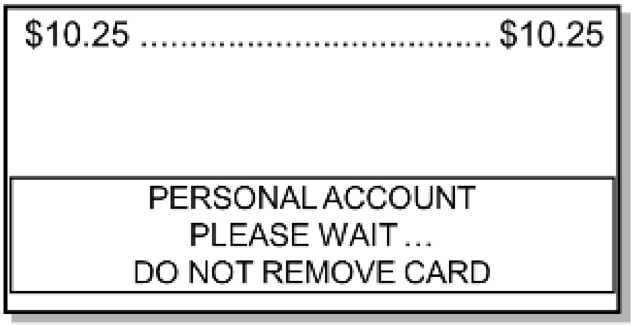


Please select language

Specification	Description
Description	Language selection
Initiated By	Flag in emv.dat
Status Response	
DFS Data Index	0030_0032
Form	ELANG.K3Z
Text	“Please select language”
Form Buttons and IDs	• “French” or “Francais” – 3 • “English” – 1 • “Español” – 2
Terminal Buttons Allowed	Cancel

10.7.7.16.9 Smart Card (EMV) Please Wait

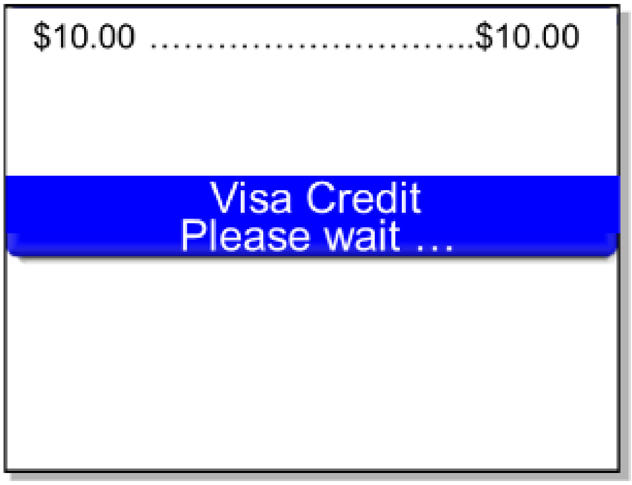
iPP320



$10.25 .... $10.25

PERSONAL ACCOUNT
PLEASE WAIT ...
DO NOT REMOVE CARD

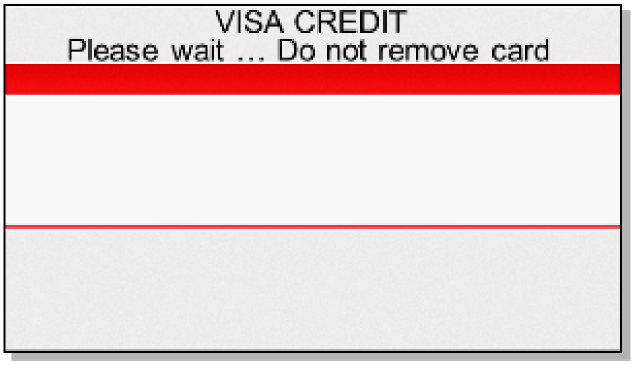
iPP350 & iWL250



$10.00 ....$10.00

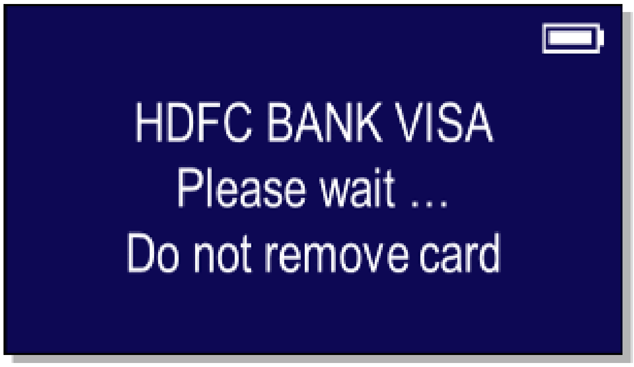
Visa Credit
Please wait ...

iSC250 & iSC350 & iSC480



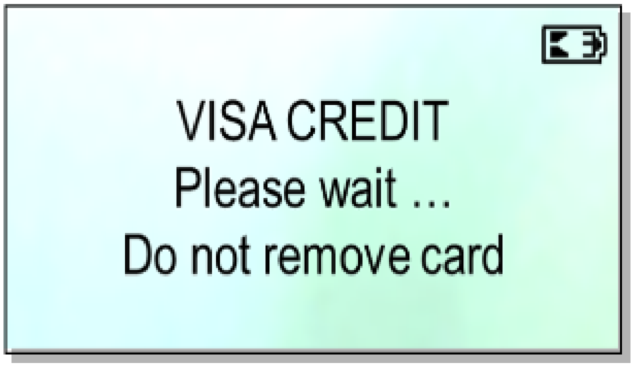
VISA CREDIT
Please wait ... Do not remove card

iSMP & iSMPc



HDFC BANK VISA
Please wait ...
Do not remove card

iCMP



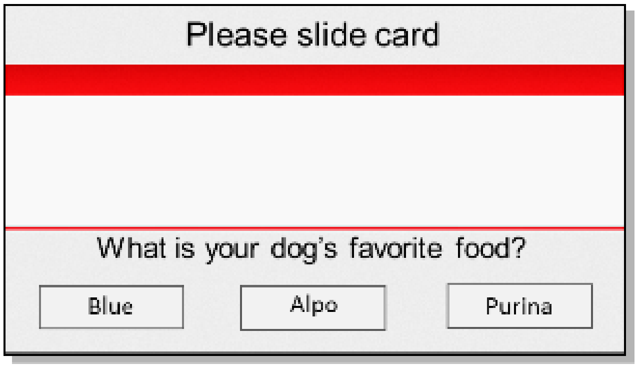
VISA CREDIT
Please wait ...
Do not remove card

EMV Contactless Transactions: Once an application has been selected and confirmed during an EMV Contact transaction, the application name is displayed on the "Please wait ..." form which follows. For EMV Contactless transactions, however, the application name will not be displayed on this form.

Specification	Description
DFS Data Index	0030_0035
Form	MSGTHICK.K3Z
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A
Notes	EMV transactions primarily uses this form when displaying two-line prompts.

10.7.7.17 Survey Swipe

iSC250 & iSC350 & iSC480



Please slide card

What is your dog's favorite food?

This form is not available for the iCMP, iSMP, iSMPc, iPP320, iPP350, or iWL250.

Description	Survey swipe screen. Customer may either slide card for payment, or participate in the survey (and slide card later).
Initiated By	40.x Survey Question Request (collects question and button text), then, 40.0 Survey Request (sends question and button text to terminal display)
Status Response before (survey) button press or card swipe	40.10 Survey Question Response (where '1' represents language #1)
Status Response after (survey) button press	40.2 Survey Request Response (this is the response for having selected the second button)

Status Response after card swipe	11.01 Slide Card
DFS Data Index	0030_0026
Form	SURSWIPE.K3Z
Text	• In blue strip: "Please Slide Card" • Survey question: Variable (see 40.x Survey Messages)
Form Buttons and IDs	Variable. Minimum 1 button, maximum 3 buttons. • Button 1 text • Button 2 text • Button 3 text
Terminal Buttons Allowed	N/A

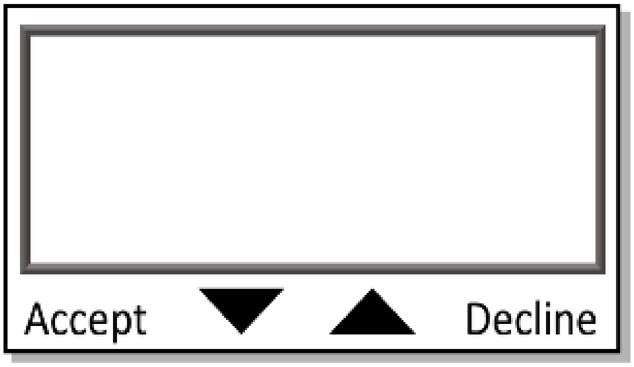
10.7.7.18 Terms and Conditions Forms

This section includes the following forms:

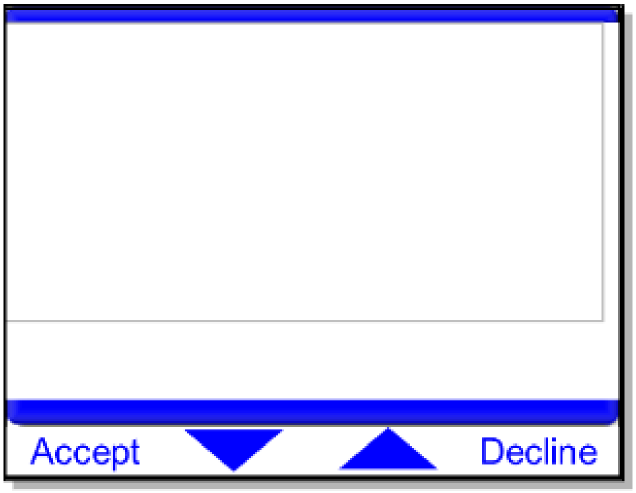
- Terms and Conditions
- Terms and Conditions Signature

10.7.7.18.1 Terms and Conditions

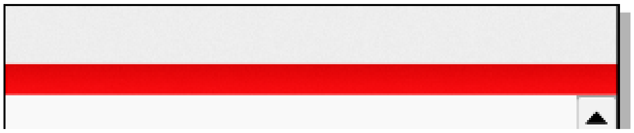
iPP320



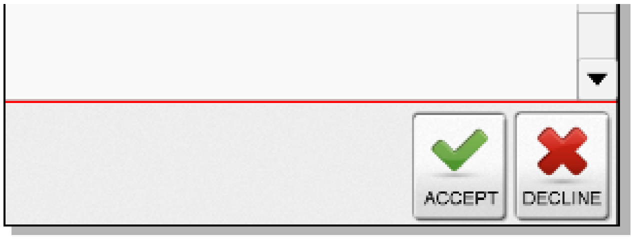
iPP350 & iWL250

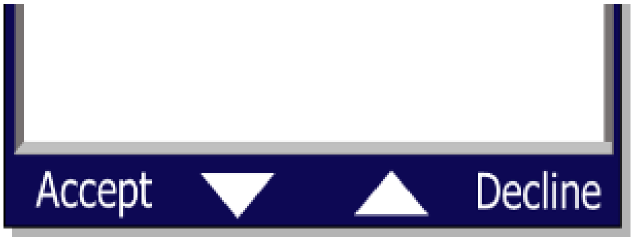


iSC250 & iSC350 & iSC480

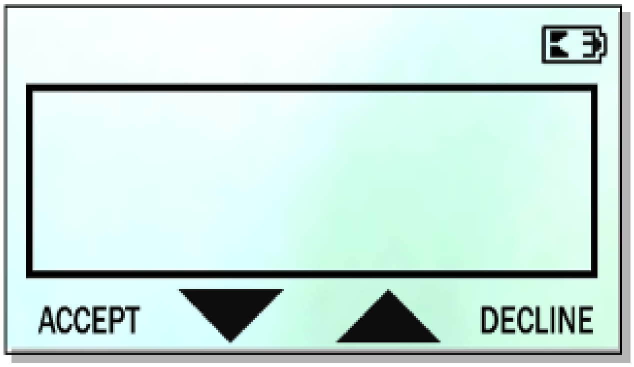


iSMP & iSMPc

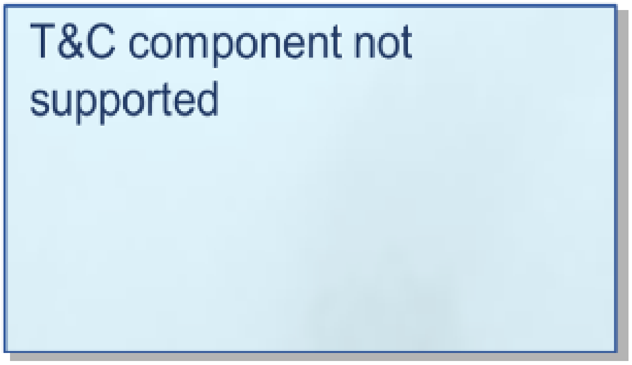




iCMP



iUP250

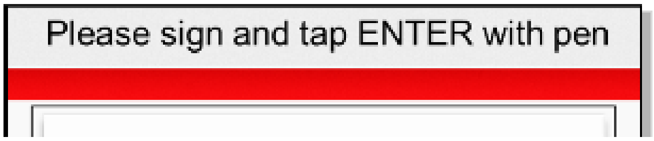


T&C component not supported

Description	Terms and conditions
Initiated By	25.x
Status Response	25.0Y or 25.0N
DFS Data Index	0030_0012
Form	TC.K3Z
Text	Variable
Form Buttons and IDs	• "Accept" • "Decline" • Scroll arrow up • Scroll arrow down
Terminal Buttons Allowed	N/A

10.7.7.18.2 Terms and Conditions Signature

iSC250 - iSC350 - iSC480



Please sign and tap ENTER with pen

This form is not available for the iCMP, iSMP, iSMPc, iPP320, iPP350 and iWL250.

Description	Terms and conditions signature
Initiated By	After Accept or Decline on tc.K3Z
Status Response	
DFS Data Index	0030_0013
Form	TCSIGN.K3Z
Text	Variable
Form Buttons and IDs	• “OK” – Y • “Clear” – N
Terminal Buttons Allowed	N/A

10.7.8 Form Customization

There are a number of ways forms can be tailored to suit the needs of any particular user- typically done by altering the value of any of the Form Variables. This section covers the variables that can be used to meet those needs.

10.7.8.1 Bluetooth Status Icon

A Bluetooth status icon can be added to forms on the iSMP, iSMP Companion, iCMP and iWL250 devices. The <BluetoothStatus.../> element may be added to .K3Z forms on these devices.

10.7.8.1.1 Defining the Bluetooth Object

The Bluetooth icon K3Z object is defined as:

```
<BluetoothStatus id='BLUETOOTHSTATUS' x='xxx' y='yyy' width='www' height='hhh'
BTActiveImage='image file name' BTInActiveImage ='image file name'/>
```

where:

- xxx = the location of the left edge of the status image.
- yyy = the top edge of the status image.
- www = the width of the status image.
- hhh = the height of the status image.
- image file name = the path to the image to display (depending on the status).

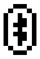
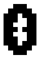
The status icon will only be displayed if the communication mode is set to Bluetooth, so no new forms need be created for Bluetooth or other communication modes.

The ID parameter must be set to 'BLUETOOTHSTATUS' for the icon to appear.

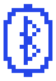
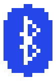
10.7.8.1.2 Icons Displayed

The icons in the two tables below have been doubled in size for the sake of visibility.

iSMP and iCMP Bluetooth Icons

Icon	Description
	Bluetooth is enabled, but not connected.
	Bluetooth is both enabled and connected.

iWL250 Bluetooth Icons

Icon	Description
	Bluetooth is enabled, but not connected.
	Bluetooth is both enabled and connected.

10.7.8.2 Form Variables

Certain variables can be displayed on a form as described in the following table. These variables are always shown in text preceded by "<?iv" and succeeded by ">". For example: <?ivIP_ADDR_PORT?>.

Variable ID	Description
AMOUNT	Purchase amount in dollars and cents.
CASHBACK	Cashback amount in dollars and cents.
CB_INC	Cashback increment in dollars and cents.
CB_MAX	Cashback maximum in cents.

Variable ID	Description
CB_MAX_TEXT	Cashback maximum in dollars and cents.
CB1	Quick cashback 1 amount in dollars and cents.
CB1$	Quick cashback 1 amount in dollars.
CB2	Quick cashback 2 amount in dollars and cents.
CB2$	Quick cashback 2 amount in dollars.
CB3	Quick cashback 3 amount in dollars and cents.
CB3$	Quick cashback 3 amount in dollars.
CB4	Quick cashback 4 amount in dollars and cents.
CB4$	Quick cashback 4 amount in dollars.
EMVAPPNAME	Name of the EMV application selected from a chip card.
IP_ADDR_PORT	IP address & port "x.x.x.x:p" format, see IP Address and Port Display Variable for TCP/IP.
LANG_COUNT	Number of languages supported.
PARTIALPAY	Partial payment enabled (1 = yes, 0 = no).
PROMPT%d	Text prompts (replace %d with prompt index).
RBAADIMAGES	Image to display in `ad.htm`.
RBATRANSAD	Image to display in transaction advertising.
SURV_BTN1	Button for survey option 1.
SURV_BTN2	Button for survey option 2.
SURV_BTN3	Button for survey option 3.
SURV_QUES	Survey Question.
TERMINAL	Terminal number.

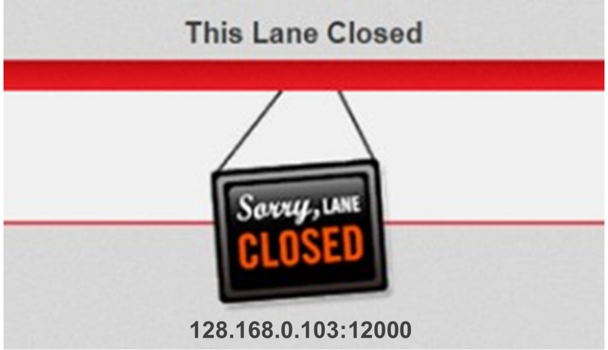
Variable ID	Description
textID	Allows text parameter substitution for labels.
TOTAL	Total amount in dollars and cents.
TYPE	Payment description.
VAR%d	User variables (wherein %d is replaced with the variable number).
WIFISSID	iWL258, iWL228 and iSMP4 only. Displays the SSID of the access point the terminal is connected to, if any. Variable is empty on unsupported terminals or if not connected via Wi-Fi.

10.7.8.2.1 IP Address and Port Display Variable for TCP/IP

IP Address Display Variable

Terminals can display their IP address without having to reboot. The form variable <?ivIP_ADDR_PORT?> displays whether a TCP/IP (Ethernet) connection is enabled. Default forms, such as This Lane Closed, can be configured using this variable to display the IP address.

The form displays the address in the format www.xxx.yyy.zzz:ppppp:

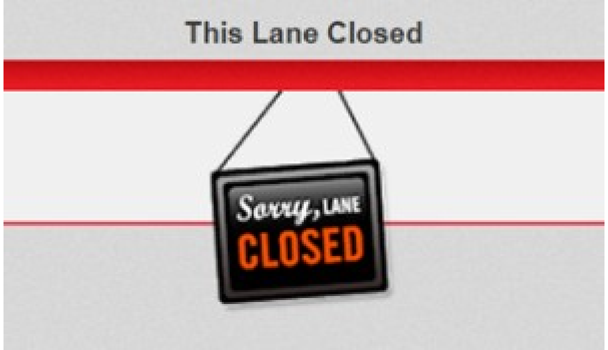


This Lane Closed

128.168.0.103:12000

Lane Closed Form Displaying IP Address

If the variable is not set as active, the IP address is not displayed:



This Lane Closed

Lane Closed Form Default

Using the Variable

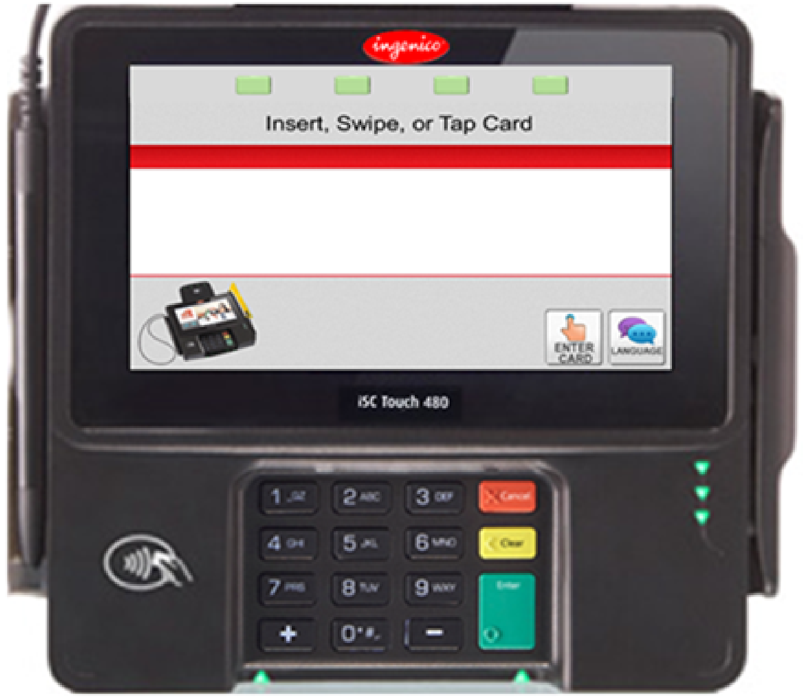
The following example shows how the variable is used in the code for the This Lane Closed form:

```
<Form x='0' y='0' width='480' height='272' template='TEMPLLD.HTM'
 backgroundcolor='D7D7D7'
timeout='0' enterenabled='false' entertone='0' clearenabled='false' cleartone='0'
cancelenabled='false' canceltone='0' tonetype='0' f1enabled='false' f2enabled='false'
f3enabled='false' f4enabled='false' />
<Image image='BG-IngBase1.png' hostimage='.\BG-IngBase1.png' id='image1' x='0' y='0'
width='480' height='272' />
<Image image='LnDisPrompt.png' hostimage='.\LnDisPrompt.png' id='image2' x='0' y='46'
width='480' height='130' />
<Image image='lane-clsd.png' hostimage='.\lane-clsd.png' id='image3' x='164' y='71'
width='152' height='174' />
<Label id='lbl1' textsource='custom' text='This Lane Closed' x='0' y='12' width='480'
height='30' border='false' bordercolor='000000' textcolor='3C3C3C' fontsize='17px'
fontweight='bold' fontfamily='userfont1' align='center' background='false'
backgroundcolor='FFFFFF' />
<Label id='IP_ADDR_PORT' textsource='custom' text='&lt;?ivIP_ADDR_PORT?&gt;' x='0'
 y='240'
width='480' height='30' border='false' bordercolor='000000' textcolor='3C3C3C'
fontsize='17px' fontweight='bold' fontfamily='userfont1' align='center'
 background='false'
backgroundcolor='FFFFFF' />
```

10.7.9 Using the iSC480 Terminal Screen to Display Contactless Status

10.7.9.1 Overview

The iSC480 terminal can be configured with an internal contactless reader or external contactless reader. Because there are no built-in LEDs for the internal contactless reader, they can be emulated using the terminal display. Just as the external contactless reader has four green LEDs that are illuminated when the contactless card enters the RF field, four green LEDs are displayed in a similar manner at the top of the terminal screen as illustrated:



iSC480 terminal with Simulated Contactless Status LEDs

10.7.9.2 Implementation

To implement this feature, contactless must be enabled. The application identifies whether the contactless reader is internal or external. If an internal contactless reader is detected, forms are shifted down to display the simulated contactless status LEDs.

10.7.10 Button IDs and Images

Text for dynamically generated buttons is specified in the `PROMPT.XML` file in the prompts directory. A separate `PROMPT.XML` is provided for each terminal.

Some customers use Coupons.com support with the application, which involves displaying a button on the main screen to trigger a sign-in/sign-up process. RBA returns button IDs that are not recognized to enable the sign-up/sign-in process. using the 24.x Form Entry Request (On-Demand) message. During the standard application flow, any button press that is not handled by the application results in a 24.x message being sent to the POS. The message format is the same as that of the response message sent when a button is pressed in response to a form display request. This feature is not supported by on-demand messages, which have their own response to return key presses.

Set a custom button ID in the Telium Form Builder using one of the following value types:

- Hexadecimal
- Character (supported)
- Decimal

However use only the Character type. Refer to the Form Contents and Descriptions section for a description and images of forms.

There are also BmpButton IDs associated with each form, which are reserved and should not be used for custom buttons. Refer to Reserved Form Buttons for a list of reserved BmpButton IDs for each form.

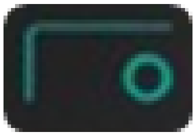
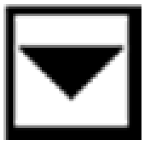
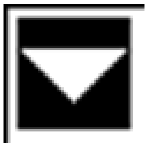
Refer to the following sections for buttons and button IDs specific to each terminal:

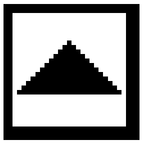
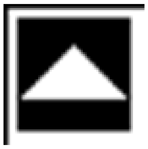
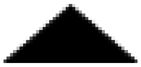
- iSMP and iSMPc Button IDs and Images
- iPP320 Button IDs and Images
- iPP350 Button IDs and Images
- iSC250 Button IDs and Images
- iSC350 Button IDs and Images
- iSC480 Button IDs and Images

Also refer to Mobile Terminal Battery Level Icons.

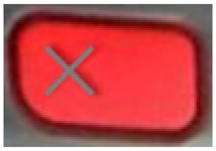
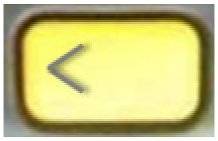
10.7.10.1 iCMP Button IDs and Images

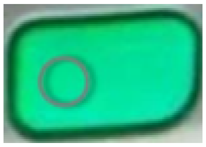
Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	20 (cash back amt 1)	143	1
Dynamically generated text	40 (cash back amt 2)	147	2
Dynamically generated text	80 (cash back amt 3)	155	3
Dynamically generated text	Accept	101	Y
Dynamically generated text	Clear	103	0x08 [ESC]
Dynamically generated text	Credit	104	66
Dynamically generated text	Debit	105	65
Dynamically generated text	Decline	106	N
Dynamically generated text	EBT Cash	107	67
Dynamically generated text	EBT Food	108	68
Dynamically generated text	English	109	1
Dynamically generated text	Enter Card	110	M
Dynamically generated text	Español	111	2
Dynamically generated text	No	113	N

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	Ok	114	0x0D [CR]
Dynamically generated text	Other	115	O
Dynamically generated text	Partial Payment	116	
Dynamically generated text	Store	117	69
Dynamically generated text	Yes	118	Y
	CANCEL		0x1B
	CLEAR		N/A
	Down		N/A
	ENTER		0x0D [CR]
	FUNCTION		
 Not Selected  Selected	Scroll Down		N/A

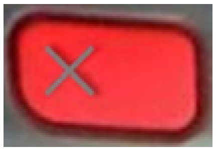
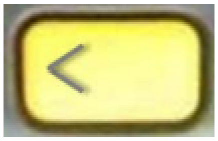
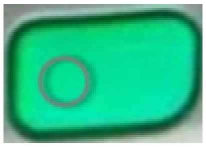
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Scroll Up		N/A
	Up		N/A

10.7.10.2 iPP320 Button IDs and Images

Button Images	Button Name	Button ID	Key Press Value
Dynamically generated text	20 (cash back amt 1)	143	1
Dynamically generated text	40 (cash back amt 2)	147	2
Dynamically generated text	80 (cash back amt 3)	155	3
Dynamically generated text	Accept	101	Y
	CANCEL		0x1B
	CLEAR	N/A	
Dynamically generated text	Clear	103	0x08 [ESC]
Dynamically generated text	Credit	104	66
Dynamically generated text	Debit	105	65
Dynamically generated text	Decline	106	N

Button Images	Button Name	Button ID	Key Press Value
▼	Down		N/A
Dynamically generated text	EBT Cash	107	67
Dynamically generated text	EBT Food	108	68
Dynamically generated text	English	109	1
	ENTER		0x0D [CR]
Dynamically generated text	Enter Card	110	M
Dynamically generated text	Español	111	2
Dynamically generated text	No	113	N
Dynamically generated text	Ok	114	0x0D [CR]
Dynamically generated text	Other	115	O
Dynamically generated text	Partial Payment	116	
Dynamically generated text	PayPal		P
🔽	Scroll Down		N/A
☰	Scroll Thumb		N/A
🔼	Scroll Up		N/A
Dynamically generated text	Store	117	69
▲	Up		N/A
Dynamically generated text	Yes	118	Y

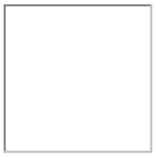
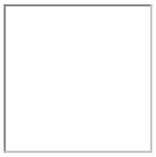
10.7.10.3 iPP350 Button IDs and Images

Button Image	**Button Name**	**Button ID**	**Key Press Value**
Dynamically generated text	20 (cash back amt 1)		1
Dynamically generated text	40 (cash back amt 2)		2
Dynamically generated text	80 (cash back amt 3)		3
Dynamically generated text	Accept	101	Y
	CANCEL		0x1B
	CLEAR		N/A
Dynamically generated text	Clear	103	0x08 [ESC]
Dynamically generated text	Credit	104	66
Dynamically generated text	Debit	105	65
Dynamically generated text	Decline	106	N
▼	Down		N/A
Dynamically generated text	EBT Cash	107	67
Dynamically generated text	EBT Food	108	68
Dynamically generated text	English	109	1
	ENTER		0x0D [CR]
Dynamically generated text	Enter Card	110	M

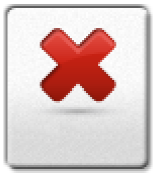
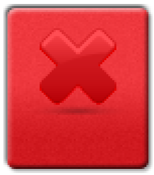
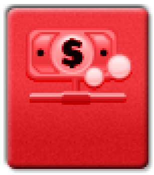
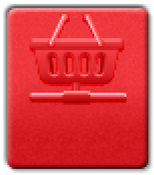
Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	Español	111	2
Dynamically generated text	No	113	N
Dynamically generated text	Ok	114	0x0D [CR]
Dynamically generated text	Other	115	O
Dynamically generated text	Partial Payment	116	
PayPal	PayPal		P
▽ Not Selected ▼ Selected	Scroll Down		N/A
▭	Scroll Thumb		N/A
△ Not Selected ▲ Selected	Scroll Up		N/A
Dynamically generated text	Store	117	69
▲	Up		N/A
Dynamically generated text	Yes	118	Y

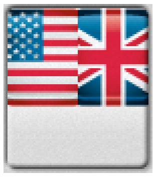
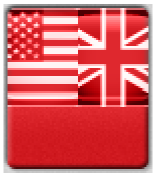
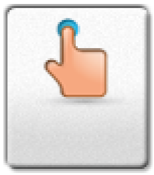
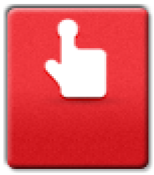
10.7.10.4 iSC250 Button IDs and Images

Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	20 (cash back amt 1)	143	1
Not selected Selected	40 (cash back amt 2)	147	2
Not Selected Selected	80 (cash back amt 3)	155	3

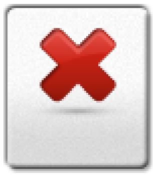
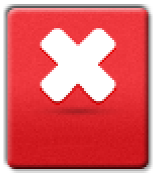
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	100 (cash back amt 4)	159	4
Not Selected Selected	Accept	101	Y
	CANCEL (*physical*)		0x1B
	CLEAR (*physical*)		N/A

Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	Clear	103	0x08 [ESC]
Not Selected Selected	Credit	104	66
Not Selected Selected	Debit	105	65

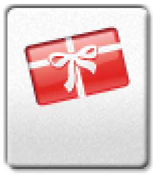
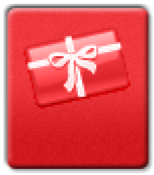
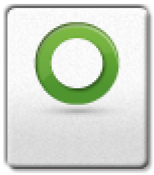
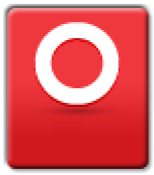
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Decline	106	N
Not Selected  Selected	EBT Cash	107	67
Not Selected  Selected	EBT Food	108	68

Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	English	109	1
	ENTER (*physical*)		0x0D [CR]
 Not Selected  Selected	Enter Card	110	M

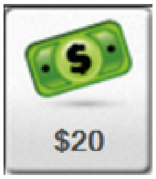
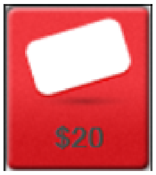
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	Español	111	2
Not Selected Selected	Français	112	3
Not Selected Selected	LANGUAGE	177	L

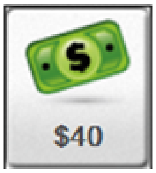
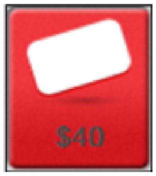
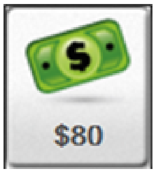
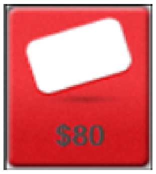
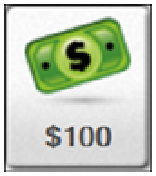
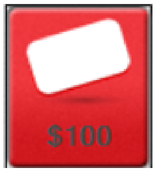
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	No	113	N
Dynamically generated	Ok	114	0x0D [CR]
Not Selected Selected	Other	115	O
Not Selected Selected	PARTIAL PAYMENT	116	

Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	btnp		P
Not Selected Selected	Scroll Down		N/A
	Scroll Thumb		N/A
Not Selected Selected	Scroll Up		N/A

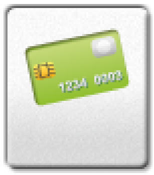
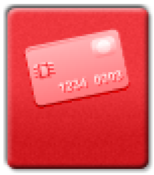
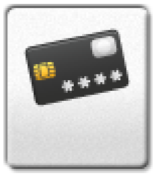
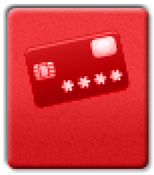
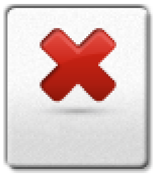
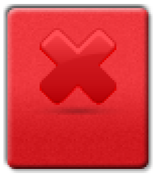
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Store	117	69
 Not Selected  Selected	Yes	118	Y

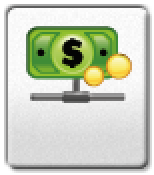
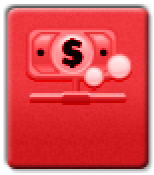
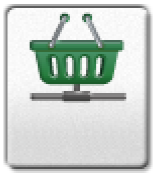
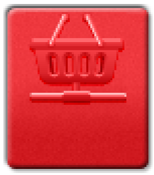
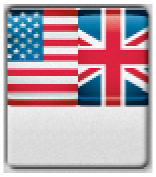
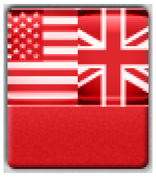
10.7.10.5 iSC350 Button IDs and Images

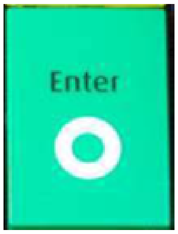
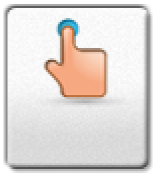
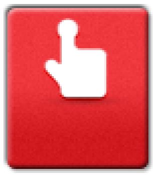
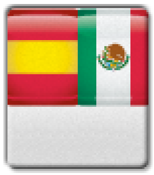
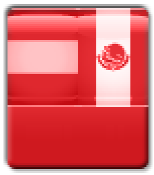
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	20 (cash back amt 1)	143	1

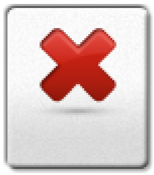
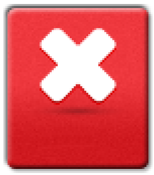
Button Image	Button Name	Button ID	Key Press Value
 Not selected  Selected	40 (cash back amt 2)	147	2
 Not Selected  Selected	80 (cash back amt 3)	155	3
 Not Selected  Selected	100 (cash back amt 4)	159	4

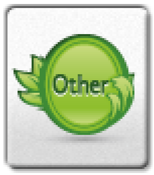
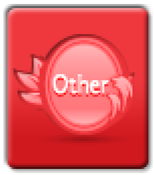
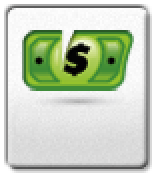
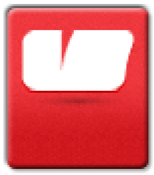
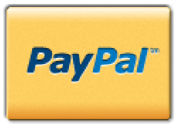
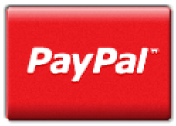
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	Accept	101	Y
	CANCEL (*physical*)		0x1B
	CLEAR (*physical*)		N/A
Not Selected Selected	Clear	103	0x08 [ESC]

Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Credit	104	66
 Not Selected  Selected	Debit	105	65
 Not Selected  Selected	Decline	106	N

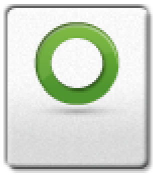
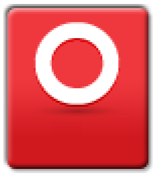
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	EBT Cash	107	67
 Not Selected  Selected	EBT Food	108	68
 Not Selected  Selected	English	109	1

Button Image	Button Name	Button ID	Key Press Value
 	ENTER (*physical*)	114	0x0D [CR]
 Not Selected  Selected	Enter Card	110	M
 Not Selected  Selected	Español	111	2

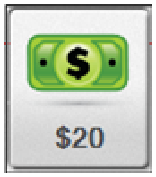
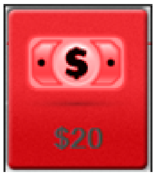
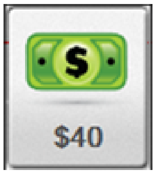
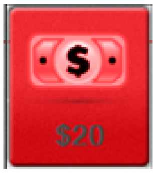
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	Français	112	3
Not Selected Selected	LANGUAGE	177	L
 Not Selected  Selected	No	113	N
Dynamically generated	Ok	167	0x0D [CR]

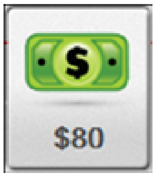
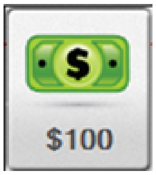
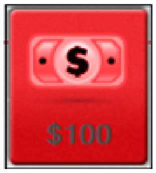
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Other	115	O
 Not Selected  Selected	PARTIAL PAYMENT	116	
 Not Selected  Selected	btnp		P

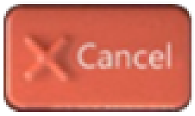
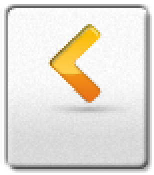
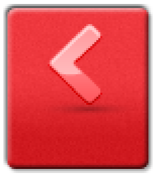
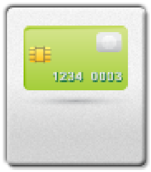
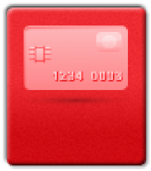
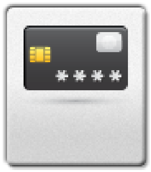
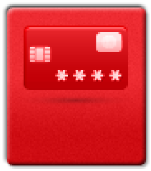
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	Scroll Down		N/A
	Scroll Thumb		N/A
Not Selected Selected	Scroll Up		N/A
Not Selected Selected	Store	117	69

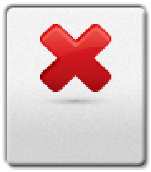
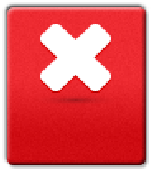
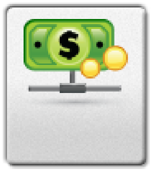
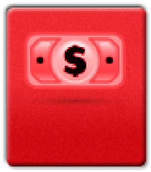
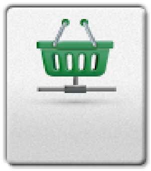
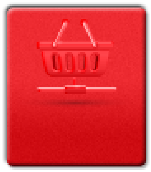
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Yes	118	Y

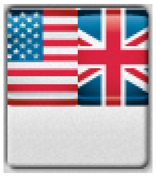
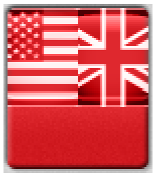
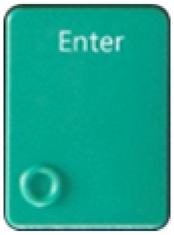
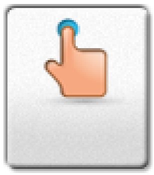
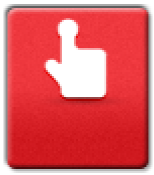
10.7.10.6 iSC480 Button IDs and Images

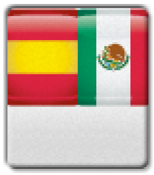
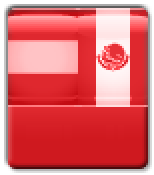
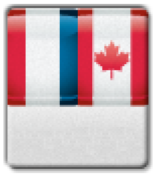
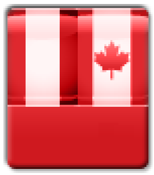
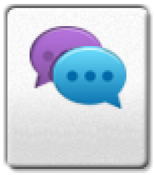
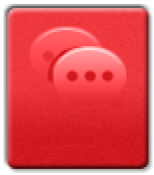
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	20 (cash back amt 1)	143	1
 Not Selected  Selected	40 (cash back amt 2)	147	2

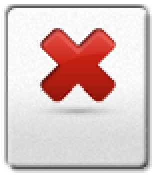
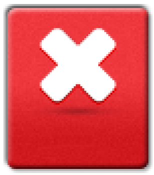
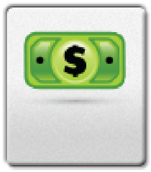
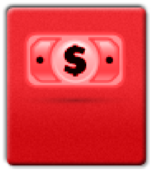
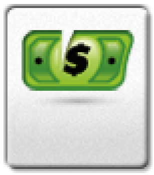
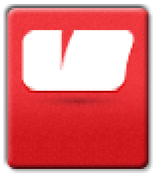
Button Image	Button Name	Button ID	Key Press Value
 Not Selected Selected	80 (cash back amt 3)	155	3
 Not Selected  Selected	100 (cash back amt 4)	159	4
Not Selected Selected	Accept	101	Y
	CANCEL (*physical*)		0x1B

Button Image	Button Name	Button ID	Key Press Value
	CLEAR (*physical*)		N/A
 Not Selected  Selected	Clear	103	0x08 [ESC]
 Not Selected  Selected	Credit	104	66
 Not Selected  Selected	Debit	105	65

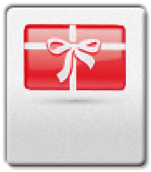
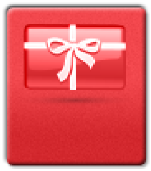
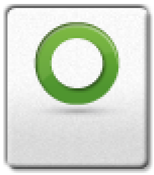
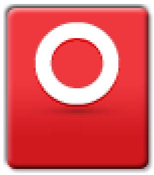
Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Decline	106	N
 Not Selected  Selected	EBT Cash	107	67
 Not Selected  Selected	EBT Food	108	68

Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	English	109	1
	ENTER (*physical*)		0x0D [CR]
 Not Selected  Selected	Enter Card	110	M

Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Español	111	2
 Not Selected  Selected	Français	112	3
 Not Selected  Selected	LANGUAGE	177	L

Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	No	113	N
Dynamically generated	Ok	114	0x0D [CR]
 Not Selected  Selected	Other	115	O
 Not Selected  Selected	PARTIAL PAYMENT	116	

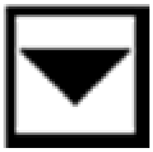
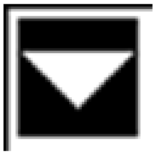
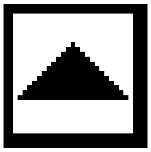
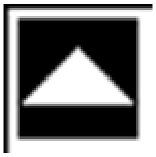
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	btnp		P
Not Selected Selected	Scroll Down		N/A
	Scroll Thumb		N/A
Not Selected Selected	Scroll Up		N/A

Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Store	117	69
 Not Selected  Selected	Yes	118	Y

10.7.10.7 iSMP and iSMPc Button IDs and Images

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	20 (cash back amt 1)	143	1
Dynamically generated text	40 (cash back amt 2)	147	2
Dynamically generated text	80 (cash back amt 3)	155	3
Dynamically generated text	Accept	101	Y
Dynamically generated text	Clear	103	0x08 [ESC]
Dynamically generated text	Credit	104	66
Dynamically generated text	Debit	105	65

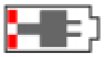
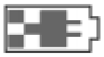
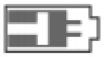
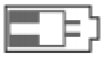
Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	Decline	106	N
Dynamically generated text	EBT Cash	107	67
Dynamically generated text	EBT Food	108	68
Dynamically generated text	English	109	1
Dynamically generated text	Enter Card	110	M
Dynamically generated text	Español	111	2
Dynamically generated text	No	113	N
Dynamically generated text	Ok	114	0x0D [CR]
Dynamically generated text	Other	115	O
Dynamically generated text	Partial Payment	116	
Dynamically generated text	Store	117	69
Dynamically generated text	Yes	118	Y
✕ cancel	CANCEL		0x1B
‹ clear	CLEAR		N/A
▼	Down		N/A
enter	ENTER		0x0D [CR]
F	FUNCTION		

Button Image	Button Name	Button ID	Key Press Value
 Not Selected  Selected	Scroll Down		N/A
 Not Selected  Selected	Scroll Up		N/A
	Up		N/A

10.7.10.8 *Mobile Terminal Battery Level Icons*

Battery-powered mobile terminal (e.g., iSMP) include a battery icon in the upper right corner of the display. At any one time, one of 12 icons will display depending upon the battery's charge level and state. The following images are displayed constantly without blinking, and appear much smaller on the terminal than shown here.

Battery Charging State Icons

Icon Image	FileName	File Size
	BATTC010.PNG	3kB
	BATTC020.PNG	3kB
	BATTC040.PNG	3kB
	BATTC060.PNG	3kB

Icon Image	FileName	File Size
🔋	BATTC080.PNG	3kB
🔋	BATTC100.PNG	3kB

Battery Discharging State Icons

Icon Image	FileName	File Size
🔋	BATTD010.PNG	2kB
🔋	BATTD020.PNG	2kB
🔋	BATTD040.PNG	2kB
🔋	BATTD060.PNG	3kB
🔋	BATTD080.PNG	3kB
🔋	BATTD100.PNG	3kB

Icon images may be customized by replacing the images with another set, maintaining the same file names as listed above. Each icon image is 11 pixels tall x 18 pixels wide.

Info
The charging state (charging or discharging) and numerals corresponding to the Mobile Terminal Battery Level (0-100%) may also be found using the 07.x: Unit Data Request message.

Info
The battery level icons appear on non-input forms only.

10.7.10.9 Reserved Form Buttons

The following table lists the BmpButton IDs which are reserved for each form. These BmpButton IDs have implicit or explicit functions in the form and should not be used for custom buttons.

Reserved Buttons by Form

File Names	BmpButton IDs
LSWIPE.K3Z	btnm, btnl, btn1, btn2, btn3, btnp
AMTV.K3Z	btnc, btnn, btnp, btny

File Names	BmpButton IDs
CASHB.K3Z	btn1, btn2, btn3, btn4, btnn, btny
CASHBA.K3Z	btn1, btn2, btn3, btn4, btnn
CASHBV.K3Z	btnn, btny
COD.K3Z/CCOD.K3Z	btnm
CELSWIPE.K3Z	btnm, btnl, btn1, btn2, btn3, btnp
CESWIPE.K3Z	btnm, btni, btng, btnp
CLSWIPE.K3Z	btnm, btnl, btn1, btn2, btn3, btnp
CPLSWIPE.K3Z	btnm, btnl, btn1, btn2, btn3, btnp
CPSWIPE.K3Z	btnm, btni, btng, btnp
CSWIPE.K3Z	btnm, btni, btng
EACCOUNT.K3Z	btn4, btn5, btn6
ECONFIRM.K3Z	btn4, btn5, btn6
ELANG.K3Z	btn4, btn5, btn6
ELSWIPE.K3Z	btnm, btnl, btn1, btn2, btn3, btnp
ESWIPE.K3Z	btnm, btni, btng
LANG.K3Z	btn1, btn2, btn3
MENU.K3Z	btn2, btn3
PAY1.K3Z	btna, btnb, btnc, btnd, btne
PLSWIPE.K3Z	btnm, btnl, btn1, btn2, btn3, btnp
PRESIGN.K3Z	btnc, btnn, btny
PSWIPE.K3Z	btnm, btni, btng, btnp
SIGN.K3Z	btnn, btny
SURSWIPE.K3Z	btn1, btn2, btn3
SWIPE.K3Z	btnm, btni, btng, btnp
TC.K3Z	btn1, btn2, btn3, btn4
TCSIGN.K3Z	btnn, btny

10.7.10.10 Ethernet Status Icon

10.7.10.10.1 Overview

The RBA supports an Ethernet status icon for iSC250 and iPP320 payment terminals. The icon can be added to any .K3Z form to provide Ethernet status via the terminal screen. To do so, use the following control:

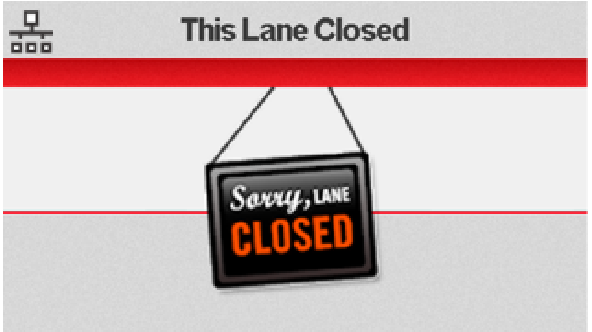
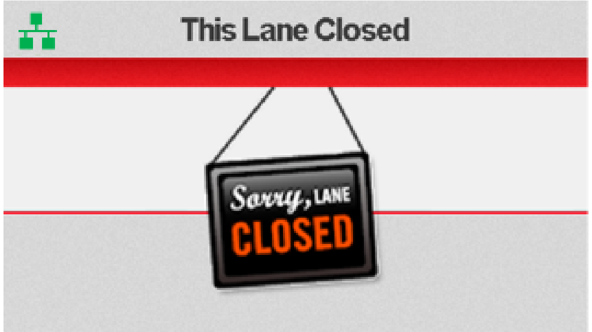
```
<EthStatus id='ETHERNETSTATUS' x='5' y='13' width='30' height='30'
 statusactive='ethactive.png'
statustrying='ethtrying.png' statusidle='ethidle.png' statusfailed='ethfailed.png'
 updateperiod='1000' />
```

where:

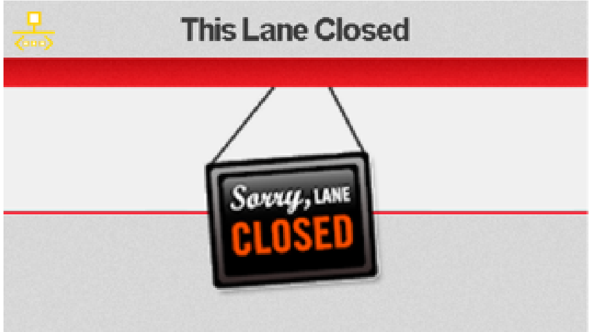
- EthStatus = Control type.
- id = Control ID. "ETHERNETSTATUS" id is mandatory.
- x = Control x position.
- y = Control y position.
- width = Icon width.
- height = Icon height.
- statusactive = Icon resource should be shown when Ethernet is connected and active.
- statustrying = Icon resource should be shown when Ethernet is trying to connect.
- statusidle = Icon resource should be shown when Ethernet is idle (temporary status after failed connection attempt).
- statusfailed = Icon resource should be shown when Ethernet failed to connect.
- updateperiod = How often RBA should update Ethernet status in milliseconds. This field is optional. The default value is 2 seconds.

The following tables illustrate the Ethernet icon for iSC250 and iPP320 terminals.

10.7.10.10.2 **iSC250 Ethernet Icons**

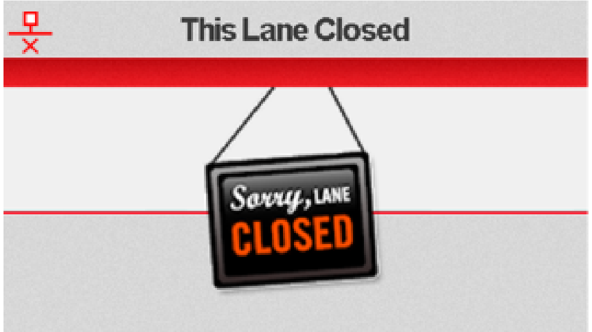
Ethernet Idle	Ethernet Active
	
Ethernet Trying	Ethernet Failed

Ethernet Trying



This Lane Closed

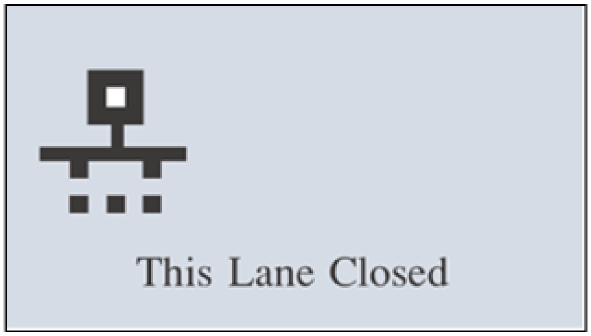
Ethernet Failed



This Lane Closed

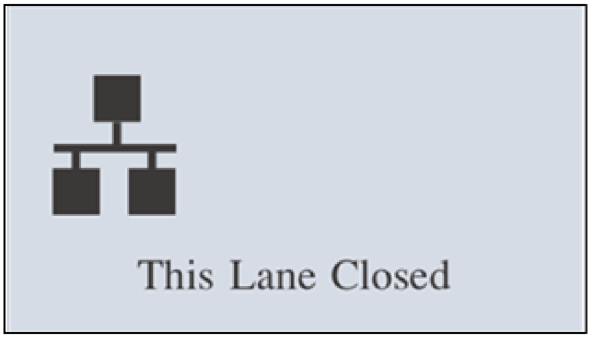
10.7.10.10.3 **iPP320 Ethernet Icons**

Ethernet Idle



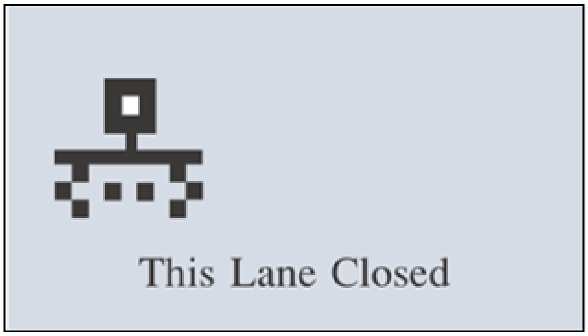
This Lane Closed

Ethernet Active



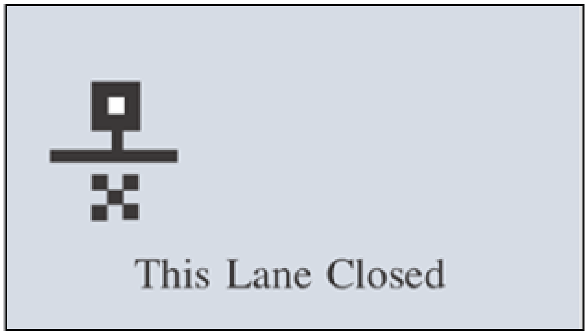
This Lane Closed

Ethernet Trying



This Lane Closed

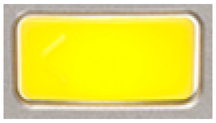
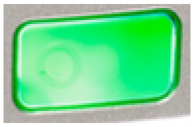
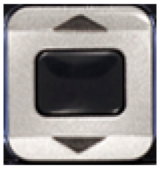
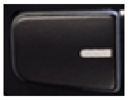
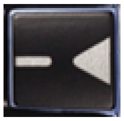
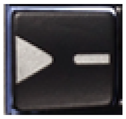
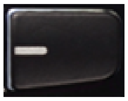
Ethernet Failed



This Lane Closed

10.7.10.11 iWL250 Button IDs and Images

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	20 (cash back amt 1)	143	1
Dynamically generated text	40 (cash back amt 2)	147	2
Dynamically generated text	80 (cash back amt 3)	155	3
Dynamically generated text	Accept	101	Y
Dynamically generated text	Clear	103	0x08 [ESC]
Dynamically generated text	Credit	104	66
Dynamically generated text	Debit	105	65
Dynamically generated text	Decline	106	N
Dynamically generated text	EBT Cash	107	67
Dynamically generated text	EBT Food	108	68
Dynamically generated text	English	109	1
Dynamically generated text	Enter Card	110	M
Dynamically generated text	Español	111	2
Dynamically generated text	No	113	N
Dynamically generated text	Ok	114	0x0D [CR]
Dynamically generated text	Other	115	O
Dynamically generated text	Partial Payment	116	
Dynamically generated text	Store	117	69
Dynamically generated text	Yes	118	Y
	CANCEL		0x1B

Button Image	Button Name	Button ID	Key Press Value
	CLEAR		N/A
	ENTER		0x0D [CR]
	Navigate Up/Down, Left/Right		
	Function key F1		N/A
	Function Key F2		N/A
	Function Key F3		N/A
	Function Key F4		N/A

10.7.10.12 iUC250 Button IDs and Images

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	20 (cash back amt 1)	143	1
Dynamically generated text	40 (cash back amt 2)	147	2
Dynamically generated text	Accept	101	Y

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	Clear	103	0x08 [ESC]
Dynamically generated text	Credit	104	66
Dynamically generated text	Debit	105	65
Dynamically generated text	Decline	106	N
Dynamically generated text	EBT Cash	107	67
Dynamically generated text	EBT Food	108	68
Dynamically generated text	English	109	1
Dynamically generated text	Enter Card	110	M
Dynamically generated text	Español	111	2
Dynamically generated text	No	113	N
Dynamically generated text	Ok	114	0x0D [CR]
Dynamically generated text	Other	115	O
Dynamically generated text	Partial Payment	116	
Dynamically generated text	Store	117	69
Dynamically generated text	Yes	118	Y
	CANCEL		0x1B
	CLEAR		N/A

Button Image	Button Name	Button ID	Key Press Value
	ENTER		0x0D [CR]
	Navigate Up/Down		

10.7.10.13 iUP250 Button IDs and Images

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	20 (cash back amt 1)	143	1
Dynamically generated text	40 (cash back amt 2)	147	2
Dynamically generated text	80 (cash back amt 3)	155	3
Dynamically generated text	Accept	101	Y
Dynamically generated text	Clear	103	0x08 [ESC]
Dynamically generated text	Credit	104	66
Dynamically generated text	Debit	105	65
Dynamically generated text	Decline	106	N
Dynamically generated text	EBT Cash	107	67
Dynamically generated text	EBT Food	108	68
Dynamically generated text	English	109	1
Dynamically generated text	Enter Card	110	M
Dynamically generated text	Español	111	2

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	No	113	N
Dynamically generated text	Ok	114	0x0D [CR]
Dynamically generated text	Other	115	O
Dynamically generated text	Partial Payment	116	
Dynamically generated text	Store	117	69
Dynamically generated text	Yes	118	Y
	CANCEL		0x1B
	CLEAR		N/A
	ENTER		0x0D [CR]
	Navigate Up/Down		

10.8 Appendix H: The Estate Manager

The Estate Manager is a server-based enterprise terminal management system that enables a customer to view the status of the installed footprint and perform remote software updates. This section describes setting up Telium terminals to communicate with The Estate Manager.

Terminals always initiate calls to Estate Manager via TCP/IP; therefore, the terminal must be able to reach the Estate Manager server via a TCP/IP connection using Ethernet or Wi-Fi.

In a typical call, the terminal checks with Estate Manager to see if there are any software updates or other packages to be downloaded, and then proceeds with the appropriate downloads. After a call, the terminal reboots.

You can schedule calls in The Estate Manager to update software on a terminal. For more information about how to configure The Estate Manager, please refer to its documentation suite.

10.8.1 Estate Manager Heartbeat

Estate Manager Heartbeat is an optional feature that informs the customer of any terminals that are disconnected or experiencing problems. In a Heartbeat call, the terminal calls The Estate Manager simply to report that it is alive and optionally, to provide some basic operational statistics. Heartbeat calls are performed more frequently than typical calls and never result in a download.

The following information about a terminal is provided by the Heartbeat function in The Estate Manager:

- Application version
- Device name
- Flash memory size
- IP address
- Number of MSR swipes
- Number of reboots
- OS version
- Pen status
- RAM size
- Security library version
- Serial number

For more information on Estate Manager Heartbeat, please refer to the Heartbeat Application User Guide, available on request.

10.8.2 Initiating a Call to The Estate Manager

There are three ways to initiate calls from a terminal running RBA to The Estate Manager:

1. The POS can initiate a call directly, by sending a specific online message: 01.0TMS0TMS.
2. The POS can set the value of RBA variable 511.
3. Calls can be scheduled on a periodic or one-time basis through a parameter file, `TMS.XML`.

Note: For Telium applications, the standard Estate Manager call scheduling feature cannot be used. Call scheduling requires the use of the `TMS.XML` file, which is described in this section.

The `TMS.XML` file is located in the HOST directory on the terminal. It can be created and edited using a standard text editor or XML editor. Merchants can load a `TMS.XML` file to a terminal using any standard download mechanism, such as the 62.x File Write message, or by including `TMS.XML` in an Estate Manager package.

The following section describes the format and contents of the TMS.XML file: Parameter File (TMS.XML)

10.8.3 Parameter File (TMS.XML)

The `TMS.XML` file contains the following parameters:

Item	Description
CALLENABLED	Set to 1 to enable call scheduling, 0 to disable.
CALLTYPE	Set to ONCE to make a single call, or PERIODIC to schedule regular calls.
LASTCALL	This parameter is used by the system to remember the time of the last call. The user should not change this value.
CALLDELAY	Time, in seconds, that the application must be inactive before it will make a scheduled call. With Standard Flow, RBA will only place a call when it is idle (no transaction in progress). With On-Demand messaging, RBA does not know when a transaction is in progress, so it will wait for a period of inactivity and then assume that no transaction in progress. This is not ideal, since the POS could still start a transaction just as RBA is preparing to make a call. For finer control over scheduled calls, the POS can use variables 510 and 511.
TIME	Time of day for the next call, in HHMMSS format.
WINDOW	Calling window in minutes. A call will be made at a random time within this period after the scheduled time. This allows the same TMS.XML to be used by multiple terminals, while preventing them all from calling the Estate Manager at the same time and possibly overloading the Estate Manager or the network.
DATE	Date for the next call, in MMDDYY format.
NOOFRETRY	Number of times that the system will retry a failed call. Note that the system reboots after each connection attempt, whether successful or not.
MAXDELAYBETWEENRETRIES	Maximum time delay, in minutes, between retries. The actual delay will be random, between 0 and this value.
TMSURL	URL of the Estate Manager server. Either the URL or the IP address should be specified, not both.
TMSIPADDRESS	IP address of the Estate Manager server. Either the URL or the IP address should be specified, not both.
TMSIPPORT	IP port of the Estate Manager server.

Item	Description
TMSSSLMODE	Indicates whether to use SSL to connect to the Estate Manager server: • 0 = Do not use SSL • 1 = SSLv2 • 2 = SSLv3 • 3 = TLSv1 • 4 = SSLv23 • 5 = TLSv1_1 Note: SSL requires the use of a certificate or a chain of certificates; these should be supplied by the administrator of the Estate Manager server, and must be loaded to the HOST directory on the terminal. Certificate files are named TMSSSL.CRT and TMSSSL.Cxx, where xx is 00 to the number of certificates minus 1. If there is only a TMSSSL.CRT file, then it assumed to be a self-signed certificate. If there are one or more TMSSSL.Cxx files, then TMSSSL.CRT is the certificate for the terminal, and TMSSSL.Cxx is the certificate chain up to the root. Certificates are added in incremental order starting at C00.
FREQUENCY	Determines the frequency of scheduled calls, if CALLTYPE is set to PERIODIC. Allowed values are: DAILY, WEEKLY, BI WEEKLY, MONTHLY, and QUARTERLY. Defaults to DAILY if not specified.

The following is an example `TMS.XML` parameter file.

```
<?xml version="1.0"?>
<TMS>
    <CallScheduling>
        <Item name="CALLENABLED" value="0" />
        <Item name="CALLTYPE" value="ONCE" />
        <Item name="LASTCALL" value="0" />
        <Item name="CALLDELAY" value="10" />
        <Item name="TIME" value="120000" />
        <Item name="WINDOW" value="90" />
        <Item name="DATE" value="121218" />
        <Item name="NOOFRETRY" value="00" />
        <Item name="MAXDELAYBETWEENRETRIES" value="00" />
    </CallScheduling>
    <TMSServer>
        <Item name="TMSURL" value="" />
        <Item name="TMSIPADDRESS" value="000.000.000.000" />
        <Item name="TMSIPPORT" value="" />
        <Item name="TMSSSLMODE" value="" />
    </TMSServer>
    <Configuration>
        <PERIODIC>
            <Item name="FREQUENCY" value="" />
        </PERIODIC>
    </Configuration>
</TMS>
```