DIV350779 Rev 9 Telium Retail Base Application (RBA) Developer's Guide

Telium Terminals (iPP320, iPP350, iSC Touch 250, iSC350, iSC Touch 480, iCMP, iSMP Companion, iSMP, iUN2xx, and iWL250 Terminals)

Ingenico Inc. - 3025 Windward Plaza, Suite 600 - Alpharetta, GA 30005 Tel: (678) 456-1200 - Fax: (678) 456-1201 - www.ingenico.com

Telium Retail Base Application (RBA) Developer's Guide
Part Number DIV350779 Rev. 9
Released August, 2015
Copyright © 2015, Ingenico Corp. All rights reserved.

Customer Service Centers:

Ingenico Inc.
3025 Windward Plaza, Suite 600
Alpharetta, GA 30005
Tel: 678.456.1200
Fax: 678.456.1201
www.ingenico-us.com

Ingenico Canada Ltd.
79 Torbarrie Road, Toronto, Ontario
Canada M3L 1G5
Tel: 416.245.6700
Fax: 416.245.6701
www.ingenico-us.com

North American Customer Support
Tel: 888.900.8221
Fax: 905.795.9343
Email: customersfirst.us@ingenico.com

Customer Service Centers:

In the U.S.A.
3025 Windward Plaza, Suite 600
Alpharetta, GA 30005

Canada
6520 Gottardo Court
Mississauga, Ontario, L5T 2A2

No part of this publication may be copied, distributed, stored in a retrieval system, translated into any human or computer language, transmitted, in any form or by any means, without the prior written consent of Ingenico. Ingenico and the Ingenico logo are registered trademarks of Ingenico Corp. All other brand names and trademarks appearing in this guide are the property of their respective holders.

Information in this document is subject to change without notice.

The information contained herein is considered an intellectual property of Ingenico and as such should be treated as confidential information to be reviewed only by authorized employees covered under the executed Mutual Non-Disclosure signed between our companies. Ingenico Corp © 2015. All rights reserved.

ingenico GROUP

Table of Contents

1_Introduction 20

1_1 Terminals Covered in this Manual 20

1_2 About This Manual 22

1_3 Definitions 23

1_4 RBA Integration Kit Contents 24

1_4_1 Ingenico Documentation 24

1_4_2 RBA Application Package 25

1_4_3 RBA-Related Utilities 29

1_5 RBA Content Locations on Terminal 29

1_6 Minimum Requirements for RBA Customization 30

1_7 Terminal Profiles 30

1_7_1 Mock-up 30

1_7_2 Production 30

1_8 LCD Display Preservation for Telium Terminals 30

1_9 Signing 30

2_Terminal Process Flow 31

2_1 EFT Overview 31

2_1_1 Assumptions 32

2_1_2 Environment 32

2_1_3 Dependencies 32

2_2 Standard Process Flow 33

2_3 On-Demand 35

2_4 Spin the BIN - BIN Lookup Process 35

2_4_1 How to Enable BIN Checking 36

2_4_2 Internal BIN Range Checking 36

2_4_3 External BIN Range Checking 36

2_5 No or Cancel Process 37

2_5_1 Amount Verification 39

2_5_2 Cash Back 39

2_5_3 PIN Entry 39

2_5_4 Signature 40

2_5_5 Transaction Cancelled 40

2_5_6 Cancel after amount received 40

2_5_7 Cancel no amount 40

2_6 Transaction End Process 40

2_6_1 Configuring 42

2_6_2 Response Messages 42

2_7 Telium Terminal Information 42

2_7_1 VID and PID Settings for HID and CDC Communications 42

ingenico GROUP

2_7_2 Communication Settings . . . 43
2_7_2_1 SSL/TLS for Ethernet . . . 44
2_7_2_2 iCMP and iSMPc Bluetooth Pairing and Unpairing . . . 47
2_7_2_3 Communications Supported per Terminal Model . . . 52
3_Functions . . . 53
3_1 Advertising Support . . . 53
3_2 Card Swipe Function Details . . . 53
3_2_1 Standard ISO Cards . . . 53
3_2_2 Non-Standard Cards . . . 53
3_2_3 Non-Payment Cards . . . 54
3_2_4 On-Demand Requirements . . . 54
3_3 Scrolling (Digital) Receipt . . . 55
3_3_1 Clearing Line Items . . . 55
3_3_2 Compression . . . 55
3_4 Amount Function . . . 55
3_4_1 Configuring Amount Function . . . 56
3_5 Signature Retrieval . . . 56
3_5_1 Configuring Signature Retrieval . . . 57
3_5_1_1 Standard . . . 57
3_5_1_2 On-Demand . . . 57
3_5_2 Retrieval Using Get Variable . . . 57
3_5_3 Signature Format . . . 59
3_5_4 Disabling Electronic Signature on iSC Series Terminals . . . 59
3_5_4_1 Overview . . . 59
3_5_4_2 Configuring for Paper Signature . . . 60
3_6 MSR Encryption . . . 60
3_6_1 Supported Encryption Methods . . . 60
3_6_2 Encryption Requirements . . . 62
3_6_3 Enabling MSR Encryption in RBA . . . 63
3_6_4 EPS (Element Payment Systems) P2PE Encryption . . . 64
3_6_4_1 EPS P2PE Card Swipe Examples . . . 65
3_6_4_2 EPS P2PE Encryption Processing Examples . . . 67
3_6_5 Generic TDES DUKPT Encryption . . . 68
3_6_5_1 TDES DUKPT Configuration . . . 68
3_6_5_2 Usage . . . 69
3_6_6 Magtek Encryption . . . 72
3_6_6_1 Overview of Magtek Encryption . . . 72
3_6_6_2 Data to be Encrypted . . . 73
3_6_6_3 Configuring Mod-10 and Account Number Check Digit . . . 74
3_6_6_4 Configuring for Manual Entry – 23.x Message . . . 75
3_6_7 Mercury Encryption . . . 76
3_6_7_1 Configuration Parameters (in config.dfs) . . . 76
3_6_7_2 Data to be Encrypted . . . 77

3_6_7_3 Encryption Does Not Occur If... 78
3_6_7_4 Other Reasons 78
3_6_7_5 Encryption Data Returned to the POS 78
3_6_7_6 Initiating an Encrypted Swipe/Tap/Manual Entry 79
3_6_7_7 Configuring for Manual Entry Using the 23.x Message 79
3_6_7_8 On-Demand Transactions 79
3_6_7_9 Card Swipe and Contactless Transactions 80
3_6_8 Monetra CardShield Encryption 82
3_6_8_1 Introduction 82
3_6_8_2 Data to be Encrypted if Track 3 is Available 83
3_6_8_3 Data to be Encrypted if Track 3 is Unavailable 83
3_6_8_4 Manual Entry 83
3_6_8_5 Encryption Does Not Occur If... 83
3_6_8_6 Encryption Data Returned to the POS – Tracks 1 & 2 83
3_6_8_7 Configuring for Manual Entry using the 23.x Message 84
3_6_8_8 Initiating an Encrypted Swipe/Tap/Manual Entry 84
3_6_8_9 Monetra CardShield Encryption Examples 85
3_6_9 On-Guard Encryption 86
3_6_9_1 Overview 86
3_6_9_2 On-Guard Configuration 87
3_6_9_3 On-Guard Card Data Encryption Rules 90
3_6_9_4 New RBA Messages for On-Guard and KME Encryption 91
3_6_9_5 Handling of Existing RBA Messages 92
3_6_9_6 MSR Encryption Example 92
3_6_9_7 E2EE Card Data Encryption 94
3_6_10 RSA-OAEP and TransArmor Encryption 101
3_6_10_1 Configuration Parameters (in config.dfs) 101
3_6_10_2 Encryption Data Returned to the POS 103
3_6_10_3 Manual Entry Transactions 103
3_6_11 S1 Encryption 104
3_6_11_1 S1 Encryption Overview 104
3_6_11_2 S1 Encryption Process 104
3_6_11_3 Configuration Parameters (in config.dfs) 105
3_6_11_4 S1 and MOD-10 106
3_6_11_5 Load Whitelist 107
3_6_11_6 Card Swipes During the Standard Flow 107
3_6_11_7 Working with Sensitive Data 107
3_6_12 Voltage TEP1 and TEP2 Encryptions 107
3_6_12_1 Overview of Voltage TEP1 and TEP2 Encryption 107
3_6_12_2 PAN Encryption 107
3_6_12_3 Obsolete Voltage Parameters 114
3_6_12_4 PAN Encryption 114
3_6_12_5 Voltage TEP1 and TEP2 Encryption Examples 115

3_6_13 MSR Encryption Processing . . . 121
3_6_14 MSR Encryption Data Variables . . . 123
3_6_15 Retrieving MSR information using the 29.x (Get Variable) Message . . . 123
3_6_16 Requesting the PIN Block Using the Masked PAN . . . 125
3_6_17 Mod-10 Checking . . . 126
3_6_18 Loading Key Serial Number (KSN) Data . . . 126
3_6_19 Selecting Specific Cards to be Encrypted . . . 126
3_6_20 Signing Requirements for .DAT File Changes . . . 127
3_7 Call Scheduling . . . 129
3_8 Contactless Key Card Support . . . 129
3_8_1 Introduction to Contactless Key Card Support . . . 129
3_8_2 RBA Low-Level Contactless Key Card Support . . . 129
3_8_2_1 Contactless Key Card General Flow . . . 130
3_8_2_2 Setting Contactless Mode (60.x/28.x) . . . 130
3_8_2_3 Enabling Contactless and Requesting Card Tap (01.x/23.x) . . . 131
3_9 Google Wallet Implementation . . . 131
3_9_1 Overview of Google Wallet . . . 131
3_9_2 Google Wallet Host Interface Messages . . . 132
3_9_3 General Payment Flow . . . 132
3_9_4 Supported Google Wallet Modes . . . 132
3_10 Softcard SmartTap Implementation . . . 132
3_10_1 Overview of Softcard SmartTap . . . 132
3_10_2 Configuring the Terminal and Sending Softcard SmartTap Data to the POS . . . 132
3_10_3 Supported Contactless Modes . . . 133
3_10_4 Erroneous Card Tap . . . 133
3_11 Offline Remote Key Injection (RKI) Support . . . 134
3_11_1 Overview . . . 134
3_11_2 Prerequisites/Requirements . . . 134
3_11_3 Enabling/Disabling Remote Key Injection . . . 134
3_11_4 Offline RKI Process . . . 135
3_11_5 Initiating Offline RKI through the TSA . . . 136
3_11_6 Log Files . . . 136
3_11_7 Functional Limitations . . . 137
3_12 Dynamic Update of RSA-OAEP Public Keys . . . 137
3_12_1 Overview . . . 137
3_12_2 Procedure to Dynamically Update RSA OAEP Public Keys . . . 137
3_13 Support for Voice Referral for EMV . . . 138
3_13_1 Overview . . . 138
3_13_2 Functional Description . . . 138
3_13_3 Transaction Example . . . 139
4_Host Interface Messages . . . 142
4_1 Communication Protocol Overview . . . 142
4_1_1 Link Level . . . 142

4_1_2 Data Message Format . . . 142
4_1_3 General Message Flow . . . 145
4_2 Communication Messages . . . 147
4_2_1 00.x: Offline Message . . . 148
4_2_1_1 Overview of the 00.x Offline Message . . . 148
4_2_1_2 00.x Offline Response Message . . . 148
4_2_1_3 Message Responses in the Offline State . . . 150
4_2_2 0x and 50.x: Authorization Response Message Format . . . 152
4_2_3 01.x: Online Message . . . 155
4_2_4 03.x: Set Session Key Message . . . 156
4_2_5 04.x: Set Payment Type Request . . . 157
4_2_6 07.x: Unit Data Request . . . 160
4_2_7 08.x: Health Stat . . . 165
4_2_8 09.x Card Status Message . . . 172
4_2_9 09.x: Set Allowed Payments Message . . . 174
4_2_10 10.x: Hard Reset Message . . . 175
4_2_11 11.x: Status Message . . . 177
4_2_11_1 Overview of the 11.x Status Message . . . 177
4_2_11_2 Appending the Form Name to the 11.x Status Response Message . . . 177
4_2_12 12.x: Account Message . . . 179
4_2_13 13.x: Amount Message . . . 181
4_2_14 14.x: Set Transaction Type . . . 182
4_2_15 15.x: Soft Reset Message . . . 183
4_2_16 16.x: Contactless Mode Request . . . 185
4_2_16_1 Overview of the 16.x Contactless Mode Request Message . . . 185
4_2_16_2 16.x Contactless Mode Request Message Format . . . 185
4_2_16_3 Google Wallet Mode Message . . . 188
4_2_16_4 Usage Examples . . . 189
4_2_16_5 Google Wallet Data Message . . . 189
4_2_17 17.x: Merchant Data Write . . . 190
4_2_17_1 Overview of the 17.x Merchant Data Write Message . . . 190
4_2_17_2 Contactless Key Card . . . 190
4_2_17_3 Google Wallet Merchant ID/Secret Pair Injection . . . 191
4_2_17_4 Softcard SmartTap . . . 191
4_2_17_5 17.x Merchant Data Write Request Message Format . . . 191
4_2_17_6 17.x Merchant Data Write Response Message Format . . . 197
4_2_17_7 Executing Commands as a Batch . . . 202
4_2_17_8 17.x Merchant Data Write Message Usage Examples . . . 203
4_2_18 18.x: Non-Payment Card Message . . . 206
4_2_19 19.x: BIN Lookup . . . 206
4_2_20 20.x: Signature Message (On-Demand) . . . 211
4_2_20_1 Overview of the 20.x Signature Messages (On-Demand) . . . 211
4_2_20_2 20.x Signature Message Format . . . 211

4_2_20_3 Usage Examples . . . 213
4_2_20_4 A Note About Keys and Buttons Pressed During the Signature Process . . . 214
4_2_20_5 Signature Ready Response Message (On-Demand) . . . 214
4_2_21 21.x: Numeric Input Request Message (On-Demand) . . . 215
4_2_21_1 Use of the 21.x Message to Send Encrypted Clear Entry Data . . . 215
4_2_22 22.x: Application ID Request . . . 217
4_2_23 23.x: Card Read Request (On-Demand) . . . 218
4_2_23_1 Overview of the 23.x Card Read Request Message . . . 218
4_2_23_2 23.x Message Flow When Using iConnectEFT . . . 219
4_2_23_3 New Flag for Coupon Support . . . 219
4_2_23_4 MSC Field . . . 220
4_2_23_5 Swiping an Invalid Card or Cancelling Manual Entry . . . 220
4_2_23_6 Execution of the 23.x Message . . . 220
4_2_23_7 23.x Card Read Request (On-Demand) Message Format . . . 221
4_2_23_8 Sample Responses . . . 224
4_2_24 24.x: Form Entry Request (On-Demand) . . . 224
4_2_24_1 Setting Prompt Text . . . 225
4_2_24_2 Setting Button Text . . . 225
4_2_24_3 Changing Button Visibility . . . 225
4_2_24_4 Setting the Form Timeout . . . 226
4_2_24_5 Displaying the Offline Form on iSMPc Terminals . . . 226
4_2_24_6 Using Multiple Buttons with the 24.x Message . . . 229
4_2_24_7 Displaying Text Data . . . 229
4_2_24_8 Displaying numerals or symbol characters using the 24.x message . . . 230
4_2_25 25.x: Terms and Conditions Request (On-Demand) . . . 230
4_2_26 26.x: Run Script Request (On-Demand) . . . 233
4_2_27 27.x: Alpha Input Message (On-Demand) . . . 235
4_2_27_1 Use of the 27.x Message to Send Encrypted Clear Entry Data . . . 236
4_2_28 28.x: Set Variable Request . . . 238
4_2_28_1 Overview of the Set Variable Message . . . 238
4_2_28_2 Changing Contactless Mode . . . 239
4_2_28_3 Configuring GMT Variables for PayPal Authorization . . . 239
4_2_28_4 Variables . . . 240
4_2_28_5 28.x Set Variable Request and Response Message Formats . . . 246
4_2_29 29.x: Get Variable Request . . . 248
4_2_29_1 Overview of the 29.x: Get Variable Message . . . 248
4_2_29_2 29.x Get Variable Request and Response Message Formats . . . 249
4_2_30 30.x: Advertising Request Message (On-Demand) . . . 250
4_2_31 31.x: PIN Entry Messages (On-Demand) . . . 252
4_2_31_1 Overview of the 31.x PIN Entry Message . . . 252
4_2_31_2 PIN Entry with Credit Selection Option . . . 252
4_2_31_3 Account Number Verification . . . 253
4_2_32 34.x: Save and Restore State Messages . . . 256

4_2_33 36.x Notification of Command Execution . . . 257
4_2_33_1 Overview of the 36.x Notification of Command Message . . . 257
4_2_33_2 36.x Notification of Command Message Format . . . 257
4_2_33_3 Usage Example . . . 258
4_2_34 40.x: Survey Messages . . . 258
4_2_34_1 40.0 Survey Request . . . 258
4_2_34_2 40.0 Survey Response . . . 259
4_2_34_3 40.0 Survey Request Message Display Behavior . . . 260
4_2_34_4 40.x: Survey Question Request . . . 261
4_2_34_5 40.x: Survey Question Response . . . 262
4_2_35 41.x Card Read Message . . . 266
4_2_36 50.x: Authorization Request . . . 269
4_2_37 51.x: Beep On-Demand Message . . . 273
4_2_38 52.x: PayPal Preauthorization Message . . . 274
4_2_39 58.x Terminal Discovery Message . . . 276
4_2_39_1 Overview . . . 276
4_2_39_2 Usage Examples . . . 276
4_2_39_3 Terminal Discovery Message Format . . . 277
4_2_40 60.x: Configuration Write . . . 278
4_2_41 61.x: Configuration Read . . . 282
4_2_42 62.x: File Write . . . 285
4_2_42_1 Overview of the 62.x File Write Message . . . 285
4_2_42_2 Aborting the Previous File Download . . . 285
4_2_43 63.x: Find File . . . 288
4_2_43_1 Overview . . . 288
4_2_43_2 63.x Find File Message Format . . . 289
4_2_44 64.x: Delete File . . . 291
4_2_45 70.x: Update Form Element Message . . . 292
4_2_45_1 Setting Prompt Text . . . 292
4_2_45_2 Setting Button Text . . . 292
4_2_45_3 Changing Button Visibility . . . 292
4_2_45_4 Setting the Form Timeout . . . 293
4_2_46 82.x On-Guard and KME Session Key Injection Command . . . 294
4_2_47 83.x On-Guard and KME Enable Message . . . 294
4_2_48 85.x On-Guard and KME Non-Payment Card Message . . . 297
4_2_49 86.x On-Guard and KME BIN Lookup (PIN Encouragement) Message . . . 297
4_2_50 87.x On-Guard and KME Card Read Data . . . 300
4_2_51 88.x On-Guard and KME Translate Encrypted Card Data Command . . . 303
4_2_52 89.x On-Guard and KME Register BIN Record Command . . . 303
4_2_53 90.x: P2PE Data Message . . . 303
4_2_53_1 Voltage Encryption – Generating a Key On Demand . . . 303
4_2_53_2 Voltage – Getting Encryption Transmission Block (ETB) . . . 305
4_2_53_3 RSA OAEP Encryption - Dynamically Updating RSA-OAEP Public Keys . . . 306

4_2_53_4 RSA OAEP Encryption - Deleting Public Keys from the Terminal . . . 308
4_2_53_5 RSA OAEP Encryption - Selecting Public Keys from the Terminal . . . 310
4_2_54 91.x: Print Message . . . 311
4_2_54_1 Overview of the 91.x Printer Message . . . 311
4_2_54_2 91.x Printer Message Format . . . 311
4_2_54_3 91.x: Barcode Printing . . . 314
4_2_55 93.x: Terminal Authentication Messages . . . 316
4_2_56 94.x and 95.x: Barcode Configuration Messages . . . 318
4_2_56_1 Barcode Configuration Messages . . . 318
4_2_56_2 Barcode Actions . . . 320
4_2_56_3 Barcode Message Tables . . . 320
4_2_56_4 Barcode Data Message . . . 320
4_2_56_5 Barcode Encryption Messages . . . 322
4_2_56_6 Barcode Illumination Messages . . . 327
4_2_56_7 Barcode Image Messages . . . 331
4_2_56_8 Barcode Lighting Messages . . . 333
4_2_56_9 Barcode Power Messages . . . 336
4_2_56_10 Barcode Reset Messages . . . 339
4_2_56_11 Barcode Scan Messages . . . 341
4_2_56_12 Barcode Symbology Messages . . . 347
4_2_56_13 Barcode Trigger Messages . . . 356
4_2_57 97.x: Reboot . . . 358
5_Forms . . . 360
5_1 Form Contents and Descriptions . . . 361
5_1_1 Advertising . . . 362
5_1_2 Approved/Disapproved . . . 365
5_1_3 Card Swipe Forms . . . 366
5_1_3_1 Card Swipe . . . 367
5_1_3_2 Card Swipe On Demand . . . 369
5_1_3_3 Card Swipe with Language Selection . . . 372
5_1_3_4 Remove Inserted Card . . . 375
5_1_4 Cash Back Forms . . . 375
5_1_4_1 Cash Back Other . . . 376
5_1_4_2 Cash Back Selection . . . 377
5_1_4_3 Cash Back Selection without No . . . 380
5_1_4_4 Cash Back Verification . . . 382
5_1_5 Contactless Enabled Forms . . . 383
5_1_5_1 Contactless Card Read Request . . . 384
5_1_5_2 Contactless Card Swipe . . . 387
5_1_5_3 Contactless Card Swipe with Language Selection . . . 389
5_1_6 Initialization Screen . . . 393
5_1_7 Input Entry Forms . . . 394
5_1_7_1 Alphanumeric Entry . . . 395

5_1_7_2 Input Entry . . . 396
5_1_8 Language Selection . . . 398
5_1_9 Message . . . 400
5_1_10 Offline . . . 404
5_1_11 Payment Forms . . . 405
5_1_11_1 Amount Verification . . . 406
5_1_11_2 Payment Selection . . . 408
5_1_12 PayPal Forms . . . 411
5_1_12_1 Card Swipe with PayPal . . . 412
5_1_12_2 Card Swipe with PayPal and Language Selection . . . 413
5_1_12_3 Contactless Card Swipe with PayPal and Language Selection . . . 416
5_1_12_4 Contactless Card Swipe with PayPal Selection . . . 418
5_1_12_5 PayPal Data Input (On-Demand form) . . . 419
5_1_12_6 PayPal PIN Entry . . . 421
5_1_12_7 PayPal Please Wait . . . 422
5_1_13 PIN Entry . . . 423
5_1_14 Enter PIN or Press Green for Credit . . . 425
5_1_15 Signature Forms . . . 427
5_1_15_1 Post-Sign . . . 427
5_1_15_2 Pre-Sign . . . 429
5_1_15_3 Signature (On-Demand) . . . 431
5_1_16 Smart Card (SMC) and EMV Forms . . . 434
5_1_16_1 Contactless Smart Card (EMV) and Swipe . . . 435
5_1_16_2 Contactless Smart Card (EMV) and Swipe with Language Selection . . . 438
5_1_16_3 Smart Card (EMV) and Swipe . . . 441
5_1_16_4 Smart Card (EMV) and Swipe with Language Selection . . . 445
5_1_17 Survey Swipe . . . 448
5_1_18 Terms and Conditions Forms . . . 449
5_1_18_1 Terms and Conditions . . . 450
5_1_18_2 Terms and Conditions Signature . . . 452
5_2 Using the Function Keys to Select Menu Options . . . 453
5_3 Button IDs and Images . . . 455
5_3_1 iCMP Button IDs and Images . . . 456
5_3_2 iPP320 Button IDs and Images . . . 458
5_3_3 iPP350 Button IDs and Images . . . 460
5_3_4 iSC250 Button IDs and Images . . . 463
5_3_5 iSC350 Button IDs and Images . . . 470
5_3_6 iSC480 Button IDs and Images . . . 477
5_3_7 iSMP and iSMPc Button IDs and Images . . . 483
5_3_8 Mobile Terminal Battery Level Icons . . . 486
5_3_9 Reserved Form Buttons . . . 487
5_4 Prompts . . . 489
5_4_1 Line Breaks in Prompts . . . 489

5_4_2 Custom Prompts (CUSTPROMPT.xml) . . . 490
5_4_3 Security Prompts (SECURPROMPT.xml) . . . 490
5_4_3_1 Button Text . . . 498
5_4_4 Transaction Prompts (PROMPT.xml) . . . 500
5_5 Terms and Conditions (TC1.xml) . . . 523
5_6 Form Customization . . . 523
5_6_1 Bluetooth Status Icon . . . 524
5_6_1_1 Defining the Bluetooth Object . . . 524
5_6_1_2 Icons Displayed . . . 524
5_6_2 RBA Form Variables . . . 525
5_6_2_1 IP Address and Port Display Variable for TCP-IP . . . 527
5_7 Using the iSC480 Terminal Screen to Display Contactless Status . . . 528
5_7_1 Overview . . . 528
5_7_2 Implementation . . . 528
6_Drivers and Tools . . . 530
6_1 Ingenico iConnectEFT Constants . . . 530
6_2 Ingenico iConnectREST . . . 530
6_2_1 Overview of iConnectREST . . . 530
6_2_2 Supported Payment Terminals and Connection Methods . . . 531
6_3 RBA Testing Tool . . . 532
7_Configuring the Application . . . 533
7_1 DFS Organization . . . 533
7_1_1 Data Description . . . 533
7_1_2 Data Line . . . 534
7_1_3 File Name Line . . . 534
7_1_4 File Rules . . . 535
7_2 DAT Files . . . 535
7_2_1 Advertising Parameters (ads.dat) . . . 535
7_2_1_1 Overview . . . 535
7_2_1_2 Automatic Startup of Advertising . . . 536
7_2_1_3 On-Demand Startup of Advertising . . . 536
7_2_2 Barcode Parameters (barcode.dat) . . . 540
7_2_3 BIN Lookup (stb.dat) . . . 544
7_2_4 BIN Processing (allBins.dat, bin0.dat - bin20.dat) . . . 545
7_2_4_1 . . . 559
7_2_4_2 Transaction Codes Table . . . 560
7_2_4_3 Card Source . . . 561
7_2_5 Card Configuration (cards.dat) . . . 562
7_2_5_1 Use of the Verify Amount Flag During EMV Transactions . . . 563
7_2_5_2 Converting Binary to Hexadecimal for Card Sources Allowed . . . 564
7_2_5_3 Card Configuration Table . . . 565
7_2_6 Cash Back Configuration (cashback.dat) . . . 568
7_2_7 Changes to security.dat or secbin.dat . . . 571

7_2_8 Compatibility Flags (compat.dat) . . . 571
7_2_9 Contactless Reader Configuration (cless.dat) . . . 574
7_2_10 EMV Flags (emv.dat) . . . 576
7_2_11 Form Files (forms.dat) . . . 577
7_2_12 MAC Entry (mac.dat) . . . 582
7_2_13 Main Flow (mainFlow.dat) . . . 583
7_2_14 MSR Card Swipe Options (msr.dat) . . . 592
7_2_15 PayPal Configuration (paypal.dat) . . . 593
7_2_16 PIN Entry (pin.dat) . . . 595
7_2_17 Security BIN (secbin.dat) . . . 597
7_2_18 Security Parameters (security.dat) . . . 599
7_2_19 Signature Items (sig.dat) . . . 608
7_2_20 Status Messages (status.dat) . . . 610
7_2_21 Store/Lane Information (store.dat) . . . 612
7_2_22 User Defined Variables (var.dat) . . . 614
7_3 Format Specifiers . . . 614
7_3_1 General Attributes . . . 615
7_3_2 Specific Attributes . . . 617
7_3_3 Using Multiple Format Specifier Attributes . . . 618
7_3_4 Unknown Format Specifiers . . . 618
7_3_5 Examples of Format Specifiers . . . 619
7_3_6 Clear-Text Key Press Input Support . . . 627
8_EMV Implementation . . . 630
8_1 Introduction to EMV Transactions . . . 630
8_2 EMV Transaction Sequence . . . 631
8_2_1 Card Detection . . . 632
8_2_2 Language Selection for EMV Transactions . . . 633
8_2_3 Application Selection . . . 633
8_2_3_1 Overview . . . 633
8_2_3_2 Application Selection Process Flow . . . 634
8_2_3_3 Canadian Specific Requirements . . . 635
8_2_3_4 Available EMV Application IDs . . . 635
8_2_3_5 Application Selection for '0019_0003' = '0' . . . 638
8_2_3_6 Application Selection for '0019_0003' = '1' . . . 639
8_2_3_7 Application Selection for '0019_0003' = '2' . . . 640
8_2_3_8 Application Selection for '0019_0003' = '3' . . . 641
8_2_4 Read Application Data . . . 641
8_2_5 Data Authentication . . . 642
8_2_6 Cardholder Verification . . . 642
8_2_6_1 PIN Entry for EMV Transactions . . . 642
8_2_6_2 Signature Management . . . 643
8_2_6_3 EMV Reversal . . . 643
8_2_6_4 EMV Fallback . . . 643

8_2_7 Terminal Risk Management . . . 644
8_2_8 Terminal Action Analysis . . . 644
8_2_9 First Card Action Analysis . . . 644
8_2_10 Online Transaction Authorization . . . 644
8_2_11 Second Card Action Analysis . . . 644
8_2_12 Transaction Completed . . . 644
8_3 EMV Host Interface Messages . . . 645
8_3_1 Overview of EMV Host Interface Messages . . . 645
8_3_2 EMV '33.00.x' Transaction Initiation Message . . . 646
8_3_2_1 Overview . . . 646
8_3_2_2 EMV '33.00.x' transaction Initiation Request Message Format . . . 647
8_3_2_3 EMV '33.00.x' Transaction Initiation Response Message Format . . . 649
8_3_3 EMV '33.01.x' Status Message . . . 651
8_3_3_1 Overview . . . 651
8_3_3_2 EMV '33.01.x' Status Request Message Format . . . 651
8_3_3_3 EMV '33.01.x' Status Response Message Format . . . 654
8_3_4 EMV '33.02.x' Track 2 Equivalent Data Message . . . 662
8_3_5 EMV '33.03.x' Authorization Request Message . . . 665
8_3_6 EMV '33.04.x' Authorization Response Message . . . 668
8_3_6_1 EMV '33.04.x' Authorization Response Error Reply Message . . . 670
8_3_6_2 Authorization Response Codes . . . 673
8_3_7 EMV '33.05.x' Authorization Confirmation Response Message . . . 673
8_3_8 EMV '33.07.x' Terminal Capabilities Message . . . 676
8_3_8_1 Overview of the Terminal Capabilities Request Message . . . 676
8_3_8_2 Overview of the EMV Terminal Capabilities Response Message . . . 676
8_3_8_3 EMV '33.07.x' Terminal Capabilities Message Format . . . 677
8_3_8_4 Usage Example . . . 682
8_3_8_5 EMV '33.07.x' Terminal Capabilities Message Flow . . . 683
8_3_9 EMV '33.08.x' Set Variables Message . . . 683
8_3_9_1 Overview . . . 683
8_3_9_2 EMV '33.08.x' Set Variables Request Message Format . . . 684
8_3_9_3 EMV '33.08.x' Set Variables Response Message Format . . . 686
8_3_10 EMV '33.09.x' Set Tag Data Message . . . 688
8_3_10_1 Overview . . . 688
8_3_10_2 EMV '33.09.x' Set Tag Data Request Message Format . . . 688
8_3_10_3 EMV '33.09.x' Set Tag Data Response Message Format . . . 691
8_3_10_4 EMV '33.09.x' Set Tag Data Message Usage Example . . . 693
8_3_11 EMV '33.10.x' Get Tag Data Message . . . 693
8_3_11_1 Overview . . . 693
8_3_11_2 EMV '33.10.x' Get Tag Data Request Message Format . . . 693
8_3_11_3 EMV '33.10.x' Get Tag Data Response Message Format . . . 696
8_3_11_4 '33.10.x' Set EMV Data Message Usage Example . . . 698
8_3_12 Transaction Step List . . . 698

8_3_13 EMV Tag Data Format . . . 700
8_3_14 EMV and Non-EMV Tags Transmitted in Host Interface Messages . . . 702
8_4 EMV Transaction Flow . . . 707
8_4_1 EMV Purchase Transaction Flow . . . 708
8_4_2 EMV Contactless Transaction Flow . . . 710
8_4_3 EMV Full Refund Transaction Flow . . . 712
8_4_4 EMV Partial Refund Normal Transaction Flow . . . 714
8_4_5 EMV Partial Refund On-Demand Transaction Flow . . . 716
8_4_6 MSD Contactless Transaction Flow . . . 718
8_4_7 Non-EMV Tag Definitions . . . 719
8_5 EMV On-Demand Flow . . . 724
8_5_1 Configuration . . . 724
8_5_2 Initiate On-Demand . . . 724
8_5_3 Initiate EMV Flow . . . 725
8_6 EMV with P2PE Enabled . . . 725
8_6_1 EMV Tags Used with P2PE Enabled . . . 725
8_6_2 Encryption of PAN-Related Data . . . 725
8_6_3 EMV Tag Encryption . . . 726
8_6_3_1 EMV Tag Handling During Encryption . . . 726
8_7 EMV Configuration Parameters . . . 728
8_7_1 Notes on EMV Configuration Parameters . . . 728
8_7_1_1 Overview . . . 728
8_7_1_2 Merchant and Acquirer Responsibilities and Parameter Management . . . 728
8_7_1_3 Configuration Files . . . 728
8_7_1_4 Important . . . 729
8_7_1_5 Data element format conventions . . . 729
8_7_2 Application ID (AID) Parameters in EMVCONTACT.XML . . . 730
8_7_3 Application ID (AID) Parameters in EMVCLESS.XML . . . 732
8_7_4 Certificate Authority Public Keys in EMVCONTACT.XML . . . 740
8_7_5 EMV.DAT Configuration File . . . 741
8_7_6 ICS Tags in EMVCONTACT.XML . . . 743
8_7_7 ICS Tags in EMVCLESS.XML . . . 750
8_8 MAC Messages (Canada Only) . . . 753
8_8_1 80.x MAC Calculation Message Format . . . 753
8_8_2 81.x MAC Verification Message Format . . . 756
8_9 Configuring the EMV Application . . . 758
8_10 EMV Configuration and Flow . . . 758
8_10_1 Configuration . . . 758
8_10_2 Initiate Transaction . . . 758
8_10_3 Process Online PIN . . . 759
8_10_4 PIN Block Tag Format in Authorization Request Message . . . 759
8_11 EMV Full and Partial Refunds . . . 759
8_11_1 Overview . . . 759

8_11_2 Important Tags Used in Refund Transactions . . . 760
8_11_3 Cancelling a Refund Transaction . . . 760
8_12 EMV Cashback . . . 761
8_12_1 Overview . . . 761
8_12_2 Cashback/Total Amount Confirmation . . . 762
8_12_3 Sample Messages . . . 763
8_12_4 Terminal and ECR Notification . . . 763
9_Appendices . . . 764
9_1 Appendix A. Differences Between U32 RBA and Telium RBA . . . 764
9_1_1 A.1. At A Glance . . . 764
9_1_2 A.2. Support . . . 765
9_1_2_1 Features . . . 765
9_1_2_2 Terminals . . . 765
9_1_2_3 POS Environments . . . 766
9_1_3 A.3. Signing Requirements . . . 766
9_1_4 A.4. Transaction Flows . . . 766
9_1_5 A.5. Host Interface Messages . . . 767
9_1_6 A.6. Variable Names . . . 770
9_1_7 A.7. Forms . . . 770
9_1_8 A.8. Prompts . . . 770
9_1_9 A.9. RBA Configuration (config.dfs) . . . 771
9_1_9_1 General . . . 771
9_1_9_2 Parameters . . . 771
9_1_10 A.10. Format Specifiers . . . 777
9_2 Appendix B. 3-Byte ASCII Signature Format . . . 777
9_2_1 B.1. Specifications . . . 778
9_2_2 B.2. Coordinate Data Reconstruction . . . 778
9_2_3 B.3. Format for Signature Data . . . 779
9_2_4 B.4. Pen-Up Control Character . . . 780
9_2_5 B.5. Segment Start Control Character . . . 780
9_2_6 B.6. Coordinate Character Data Set . . . 780
9_2_6_1 Coordinate Data Character 1 . . . 780
9_2_6_2 Coordinate Data Character 2 . . . 780
9_2_6_3 Coordinate Data Character 3 . . . 780
9_2_7 B.7. Unused Control Codes . . . 780
9_3 Appendix C. RBA Script Language . . . 781
9_3_1 What is a Script? . . . 781
9_3_2 Comments . . . 781
9_3_3 Tags . . . 781
9_3_4 Tag Parameters . . . 781
9_3_5 Button Parameter . . . 781
9_3_6 Form Parameter . . . 782
9_3_7 Text Parameter . . . 782

9_3_8 Sample Script with Comments . . . 783
9_4 Appendix D. PayPal Overview . . . 784
9_4_1 D.1. Minimum Production Requirements . . . 785
9_4_2 D.2. PayPal Validation Flow . . . 785
9_4_3 D.3. Configuration . . . 787
9_4_3_1 PayPal.dat . . . 787
9_4_3_2 Forms . . . 787
9_4_4 D.4. Calculating GMT Offset (Variable 205) . . . 789
9_4_4_1 Scenario 1 . . . 789
9_4_4_2 Scenario 2 . . . 789
9_5 Appendix E. RBA Best Practices . . . 789
9_5_1 E.1. Working With RBA . . . 789
9_5_1_1 Host Interface Messaging Tips . . . 789
9_5_1_2 Retrieving EPS Encrypted Data . . . 790
9_5_2 E.2. Customizing RBA . . . 791
9_5_2_1 Editing a Prompt . . . 791
9_5_2_2 Adding a Custom Prompt . . . 792
9_5_2_3 Editing a Form . . . 792
9_5_2_4 Adding a Form . . . 792
9_5_2_5 Editing Config.dfs . . . 792
9_5_3 E.3. Packaging a New RBA Load . . . 793
9_6 Appendix F. External Display for Telium Terminals . . . 793
9_7 Appendix G: eWIC Implementation . . . 794
9_7_1 G.1. eWIC Overview . . . 794
9_7_1_1 Overview . . . 794
9_7_1_2 Card Inventory . . . 795
9_7_1_3 Modes . . . 795
9_7_1_4 Primary Types of WIC Smart Cards . . . 795
9_7_2 G.2. eWIC WMP Messages . . . 796
9_7_2_1 WMP Message Specifications . . . 796
9_7_2_2 WMP Request/Response Message Pairs . . . 796
9_7_2_3 WIC Transaction and RBA Exclusivity . . . 797
9_7_2_4 N on-WIC Exemptions . . . 797
9_7_2_5 WMP Message Format . . . 798
9_7_2_6 _99 eWIC Reset Message . . . 802
9_7_3 G.3. eWIC Transaction Flows . . . 803
9_7_3_1 Messages in WIC Transactions . . . 803
9_7_3_2 POS-Terminal Server/Client System . . . 803
9_7_3_3 eWIC Sample Message Flows . . . 805
9_7_3_4 Balance Transaction Flow Breakdown . . . 807
9_7_3_5 Debit Transaction Flow Breakdown . . . 810
9_7_3_6 Transactions Canceled at PIN Entry . . . 818
9_7_3_7 Cancelled WIC PIN Entry . . . 821

9_7_4 WIC Error Codes 823
9_8 Appendix H: Creating a .TGZ File 827
10 Revision History 829

This document covers the operation of the Telium Retail Base Application (RBA) is organized into the following sections:

1. Introduction - Document organization, terminals covered in this manual, supported applications, capabilities, operation requirements, definitions, and terminal profiles.
2. Terminal Process Flow - Communication between the terminal and Point-of-Sale, transaction flow overview, and communications settings for all types of MSR transactions.
3. Functions - Card swipe functions, digital receipt, transaction amount function, and signature. The following functions are also covered in this section:
    a. MSR Encryption- All current methods of encryption and how they are used in MSR transactions.
    b. Google Wallet Implementation- Payment flow and wallet modes for Google Wallet.
    c. Softcard SmartTap Implementation- Setup, configuration, and modes for Softcard SmartTap.
    d. Contactless Key Card Support- Supported card types, flow, setting mode, enabling and requesting card tap.
    e. Offline Remote Key Injection (RKI) Support- Requirements and limitations, toggling and initiating RKI, process, and logs.
    f. Dynamic Update of RSA-OAEP Public Keys- Overview and procedure for dynamically updating RSA-OAEP encryption keys.
    g. Support for Voice Referral for EMV- Overview, function, and example transaction.
4. Host Interface Messages- In-depth information on communications protocol, descriptions and breakdowns of all messages exchanged between the terminal and host during transactions, as well as a list of iConnectEFT Constants.
5. Forms - Screen displays, button images, and cardholder prompts during the transaction process. This section also includes a description of the forms used to generate the screen images.
6. Drivers and Tools - iConnectEFT Constants, iConnectREST, and RBA Testing Tool.
7. Configuring the Application - Retail Based Application prompts and parameters that may be changed to customize the application for various Ingenico Telium terminals.
8. EMV Implementation - This section provides detailed information on how Ingenico payment terminals process EMV transactions.
9. Appendices
    a. Appendix A. Differences Between U32 RBA and Telium RBA- Support, requirements, flows, names of host interface messages and variables, forms and prompts, configuration, and format specifiers.
    b. Appendix B. 3-Byte ASCII Signature Format- Specifications, Coordinate Data Reconstruction, Format for Signature Data, Signature Characters and Coordinate Character Data Sets, Unused Control Codes
    c. Appendix C. RBA Script Language- Definition, Comments and Tags, Sample Script, and Parameters for Forms, Text, Tags, and Buttons.
    d. Appendix D. PayPal Overview- Requirements, Validation Flow, Configuration, and how to calculate offset from Greenwich Mean Time (GMT).
    e. Appendix E. RBA Best Practices- Best practices for working with RBA, customizing RBA, and packaging a new RBA load.
    f. Appendix F. External Display for Telium Terminals- Compatibility, requirements, and instructions for connecting an external display.
    g. Appendix G: eWIC Implementation- Overview of the WIC program, WMP Message usage and breakdowns, transaction flows, and a list of error codes.
    h. Appendix H: Creating a .TGZ File- Proper procedure for generating a .TGZ file for downloading to a terminal.
10. Revision History

1_Introduction

The Retail Base Application is recommended for customers who desire a “plug-and-play” application for their IBM 4680/4690 system, or any other POS system, that conforms to standard IBMEFT protocol. All additions to the standard IBMEFT protocol are covered in this manual. However, to take advantage of the iSC250/iSC350/iSC480 signature capture features, some additional code must be added to the register application (see Retrieval Using Get Variable).

The Telium terminal operate on a direct-connect basis to a point-of-sale (POS) device, or to any RS-232, USB, Ethernet, or tailgate-based host device.

1_1 Terminals Covered in this Manual

Terminals covered in this manual include the following:

- iCMP (also referred to as iCM122)
- iSMP (also referred to as iMP350)
- iSMPc (also referred to as iMP352)
- iPP320
- iPP350
- iSC250 and iSC Touch 250
- iSC350
- iSC Touch 480
- iUN2xx (includes iUP250 and iUR250 and/or iUC150)
- iWL250

> The product name is the Terminal ID (this only displays when using the 07.x: Unit Data Request and 08.x: Health Stat messages.

> All references to the iSC250 throughout this document are relevant to the iSC250 and iSC Touch 250 unless otherwise stated. All references to the iSC480 pertain to the iSC Touch 480.

> The iSC Touch 250 requires SDK 9.18.0 or later. The iSC Touch 250 hardware is PCI PTS version 4.x certified.

The following image gallery provides a pictorial overview of the terminals covered in this manual.

iCMP (iCM122)

iSMP (iMP350)

iSMPc (iMP352)

iPP320

iPP350

iWL250

iUP250 iUR250 iUC150

iSC250

iSC Touch 250

iSC350

iSC Touch 480

Image Gallery of Terminals Covered in this Manual

The application supports:

- 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 Manual

This manual provides a full overview of the Retail Base Application, including an explanation of all the tools that may be used to change parameter-driven facets of the application.

This manual also 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. 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 (**DIV350774 iSC350 Operation and Product Support Guide** or **DIV350784 iPP300 User's Guide**) which explains how to download the software package which includes 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 • iMP352 • iPP320 • iPP350 • iSC250 • iSC Touch 250 • iSC350 • iSC Touch 480 • iUN2xx (includes iUP250, iUR250 and iUC150) • iWL250.

1_4 RBA Integration Kit Contents

The integration kit contains:

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

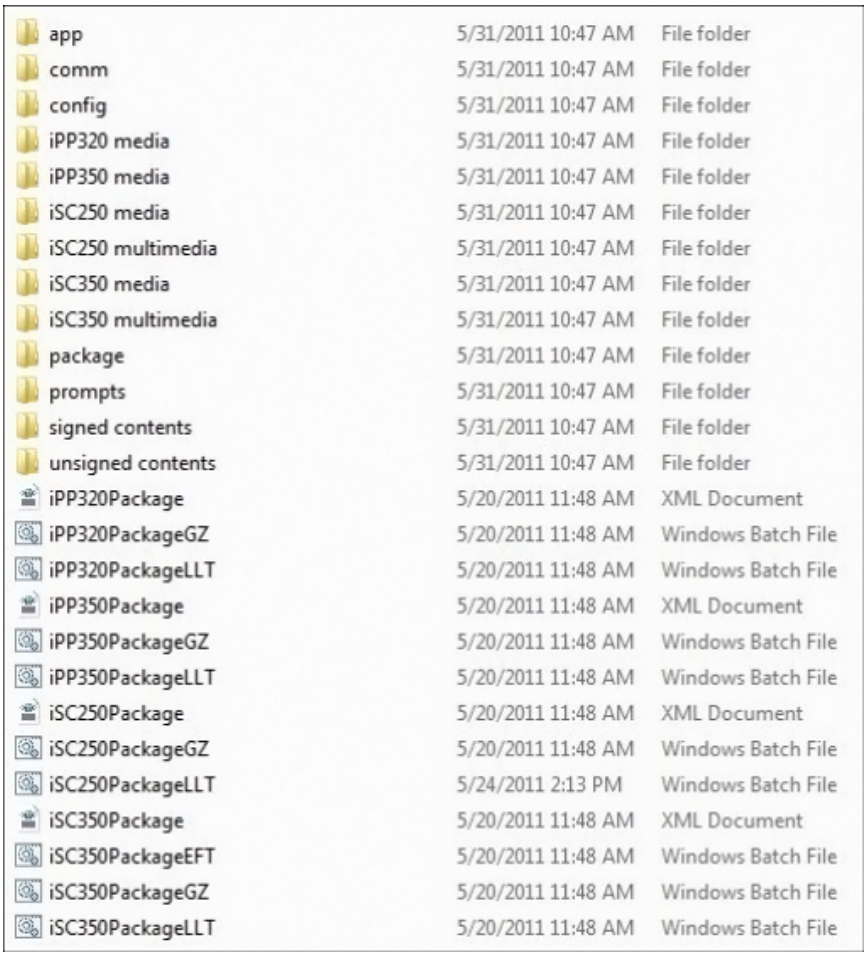
1_4_1 Ingenico Documentation

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
  - iSC250
  - iSC350
  - iSC480
  - iUP250
  - iWL250
- Documentation for Ingenico Telium utilities provided with the kit:
  - LLT
  - Bluetooth Pass-Through Service
  - Form Builder
  - Script Builder
  - Data Packaging Tool

1_4_2 RBA Application Package



The RBA Application Package Structure

The RBA Application folder included with the Integration Kit includes the following items:

- `app` folder – contains the RBA application's binary files.
- `comm` folder – contains data files and TDA.XML files to set terminals to use specific communication types.
- `config` folder – contains the following:
  - *.DAT files
  - config.dfs parameter file
  - ctr, ctr_config, and ctr_trans files
- Terminal-specific `media` folders – contain images and RBA application form (*.K3Z) files.
- Terminal-specific `multimedia` folders – contain audio and video resources.
- `prompts` folder – contains RBA application 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.
- `signed contents` folder – contains signed RBA content.
- Terminal-specific files for each supported Ingenico terminal:
- `Package definition *.XML file` – also known as the manifest. This file specifies all the information needed to package *._GZ files that will be used to load RBA content onto the corresponding terminal:
- Package name
- Files to package
- Each file's path within the original RBA application package folder structure
- Each file's 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 (`iSC350Package.XML`) is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<Package package_name="PKG" mockup="no" app="RBA" >
  <APPLICATION>
    <Item source_path="app" file_name="8295380124.AGN"/>
  </APPLICATION>
 <UNSIGNED_CONTENT>
    <!-- FILES -->
    <Item source_path="config" file_name="ads.dat"/>
    <Item source_path="config" file_name="allBins.dat"/>
    <Item source_path="config" file_name="barcode.dat"/>
```

```
<Item source_path="config" file_name="bin0.dat"/>
<Item source_path="config" file_name="bin1.dat"/>
<Item source_path="config" file_name="bin2.dat"/>
<Item source_path="config" file_name="bin3.dat"/>
<Item source_path="config" file_name="bin4.dat"/>
<Item source_path="config" file_name="bin5.dat"/>
<Item source_path="config" file_name="bin6.dat"/>
<Item source_path="config" file_name="bin7.dat"/>
<Item source_path="config" file_name="cards.dat"/>
<Item source_path="config" file_name="cashback.dat"/>
<Item source_path="config" file_name="cless.dat"/>
<Item source_path="config" file_name="compat.dat"/>
<Item source_path="config" file_name="forms.dat"/>
<Item source_path="config" file_name="mainFlow.dat"/>
<Item source_path="config" file_name="msr.dat"/>
<Item source_path="config" file_name="pin.dat"/>
<Item source_path="config" file_name="sig.dat"/>
<Item source_path="config" file_name="status.dat"/>
<Item source_path="config" file_name="stb.dat"/>
<Item source_path="config" file_name="store.dat"/>
<Item source_path="config" file_name="var.dat"/>
<!-- FORMS -->
<Item source_path="iSC350 media" file_name="ads.K3Z"/>
<Item source_path="iSC350 media" file_name="alpha.K3Z"/>
<Item source_path="iSC350 media" file_name="amtv.K3Z"/>
<Item source_path="iSC350 media" file_name="cashb.K3Z"/>
<Item source_path="iSC350 media" file_name="cashba.K3Z"/>
<Item source_path="iSC350 media" file_name="cashbo.K3Z"/>
<Item source_path="iSC350 media" file_name="cashbv.K3Z"/>
<Item source_path="iSC350 media" file_name="ccod.K3Z"/>
<Item source_path="iSC350 media" file_name="clswipe.K3Z"/>
<Item source_path="iSC350 media" file_name="cod.K3Z"/>
<Item source_path="iSC350 media" file_name="cswipe.K3Z"/>
<Item source_path="iSC350 media" file_name="input.K3Z"/>
<Item source_path="iSC350 media" file_name="lang.K3Z"/>
<Item source_path="iSC350 media" file_name="lswipe.K3Z"/>
<Item source_path="iSC350 media" file_name="msg.K3Z"/>
<Item source_path="iSC350 media" file_name="offline.K3Z"/>
<Item source_path="iSC350 media" file_name="pay1.K3Z"/>
<Item source_path="iSC350 media" file_name="pin.K3Z"/>
<Item source_path="iSC350 media" file_name="sign.K3Z"/>
<Item source_path="iSC350 media" file_name="swipe.K3Z"/>
<Item source_path="iSC350 media" file_name="tc.K3Z"/>
<Item source_path="iSC350 media" file_name="tcsign.K3Z"/>
<!-- IMAGES -->
<Item source_path="iSC350 media" file_name="AD1.JPG"/>
<Item source_path="iSC350 media" file_name="AD2.JPG"/>
<Item source_path="iSC350 media" file_name="AD3.JPG"/>
<Item source_path="iSC350 media" file_name="AD4.JPG"/>
<Item source_path="iSC350 media" file_name="AD5.JPG"/>
<Item source_path="iSC350 media" file_name="AD6.JPG"/>
<Item source_path="iSC350 media" file_name="AD7.JPG"/>
<Item source_path="iSC350 media" file_name="AD8.JPG"/>
<Item source_path="iSC350 media" file_name="AD9.JPG"/>
```

```
    <Item source_path="iSC350 media" file_name="AD10.JPG"/>
    <Item source_path="iSC350 media" file_name="AD11.JPG"/>
    <Item source_path="iSC350 media" file_name="AD12.JPG"/>
    <Item source_path="iSC350 media" file_name="AD13.JPG"/>
    <Item source_path="iSC350 media" file_name="BAR.JPG"/>
    <Item source_path="iSC350 media" file_name="BLUE_D.JPG"/>
    <Item source_path="iSC350 media" file_name="BLUE_U.JPG"/>
    <Item source_path="iSC350 media" file_name="CHECKBOX_CHK.JPG"/>
    <Item source_path="iSC350 media" file_name="CHECKBOX_EMP.JPG"/>
    <Item source_path="iSC350 media" file_name="DOWN_D.JPG"/>
    <Item source_path="iSC350 media" file_name="DOWN_U.JPG"/>
    <Item source_path="iSC350 media" file_name="RADIO_EMP.JPG"/>
  <Item source_path="iSC350 media" file_name="RADIO_SEL.JPG"/>
    <Item source_path="iSC350 media" file_name="SIGBOX.JPG"/>
    <Item source_path="iSC350 media" file_name="STEEL.JPG"/>
    <Item source_path="iSC350 media" file_name="UP_D.JPG"/>
    <Item source_path="iSC350 media" file_name="UP_U.JPG"/>
    <Item source_path="iSC350 media" file_name="CLESS.PNG"/>
    <Item source_path="iSC350 media" file_name="SCRBACK.PNG"/>
    <Item source_path="iSC350 media" file_name="SCRDOWN_D.PNG"/>
    <Item source_path="iSC350 media" file_name="SCRDOWN_U.PNG"/>
    <Item source_path="iSC350 media" file_name="SCRTHUMB.PNG"/>
    <Item source_path="iSC350 media" file_name="SCRUP_D.PNG"/>
    <Item source_path="iSC350 media" file_name="SCRUP_U.PNG"/>
    <Item source_path="iSC350 media" file_name="ISC350.GIF"/>
    <Item source_path="iSC350 media" file_name="OFFLINE.GIF"/>
    <Item source_path="iSC350 media" file_name="THICK_BOT.GIF"/>
    <Item source_path="iSC350 media" file_name="THICK_TOP.GIF"/>
    <Item source_path="iSC350 media" file_name="THIN_BOT.GIF"/>
    <Item source_path="iSC350 media" file_name="THIN_TOP.GIF"/>
    <Item source_path="iSC350 media" file_name="TOP.GIF"/>
    <!-- VIDEOS -->
    <Item source_path="iSC350 multimedia" file_name="ISC350.MP4" />
    <Item source_path="iSC350 multimedia" file_name="ISC350.MGN" />
    <Item source_path="iSC350 multimedia" file_name="SWIPE.MP4" />
    <Item source_path="iSC350 multimedia" file_name="SWIPE.MGN" />
    <!-- PROMPTS -->
    <Item source_path="prompts" file_name="PROMPT.XML" />
    <!—- SIGNED CONTENTS -->
    <Item source_path="signed contents" file_name="PROMPT.PGZ" />
  </UNSIGNED_CONTENT>
  <SIGNED_CONTENT>
    <!-- PROMPTS -->
    <Item source_path="prompts" file_name="SECURPROMPT.XML"/>
    <Item source_path="prompts" file_name="CUSTPROMPT.XML"/>
    <!-- CTG -->
    <Item source_path=" iSC350 media/CTG" file_name="ZIPCODE.JPG" dest_path="CTG"/>
  </SIGNED_CONTENT>
</Package>
```

1_4_3 RBA-Related Utilities

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

- RBA Testing Tool
- Telium LLT
- Bluetooth Pass-Through Service
- Form Builder
- Script Builder
- Data Packaging Tool
- SAT

1_5 RBA Content Locations on Terminal

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

Files and their Parent Directories

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

1_6 Minimum Requirements for RBA Customization

To modify RBA configuration settings, you need the following:

- 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)

1_7 Terminal Profiles

1_7_1 Mock-up

Mock-up terminals are used for lab testing only. The mock-up message flashes on the screen every 30 seconds.

1_7_2 Production

Production terminals are used in the live environment.

1_8 LCD Display Preservation for Telium Terminals

Ingenico Telium terminals utilize backlit LCD displays to convey transaction and advertizing information. As with any LCD display, preventative actions are recommended in order to minimize the occurrence of image persistence. Image persistence occurs when an image is displayed for extended periods, leaving a temporary impression of the image on the screen which may be partially visible when the screen changes to a new image. This can be minimized by taking the following preventative actions:

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

1_9 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_Terminal Process Flow

The RBA has a pre-designed 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 of execution is altered when you change certain configuration parameters in the `config.dfs` file, or when a POS issues an on-demand message. Since each card type (e.g., debit, credit, etc.) requires specific processes, the standard flow can be individually configured for each card type.

2_1 EFT Overview

This chapter defines the communication protocol between a POS and the 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 known as electronic funds transfer (EFT). EFT 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:

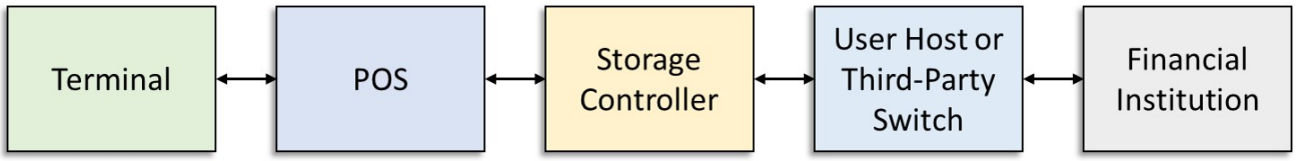
- A magnetic stripe reader (MSR) for reading the information encoded on the debit card
- A PIN keypad for entering the personal identification number (in some environments, it is required that the PIN keypad is usable for numeric entry of other data such as dollar amount of transaction)
- A display for 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.

2_1_1 Assumptions

The following is an assumed typical configuration for our industry:



Assumed Points of Communication

2_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.

2_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.

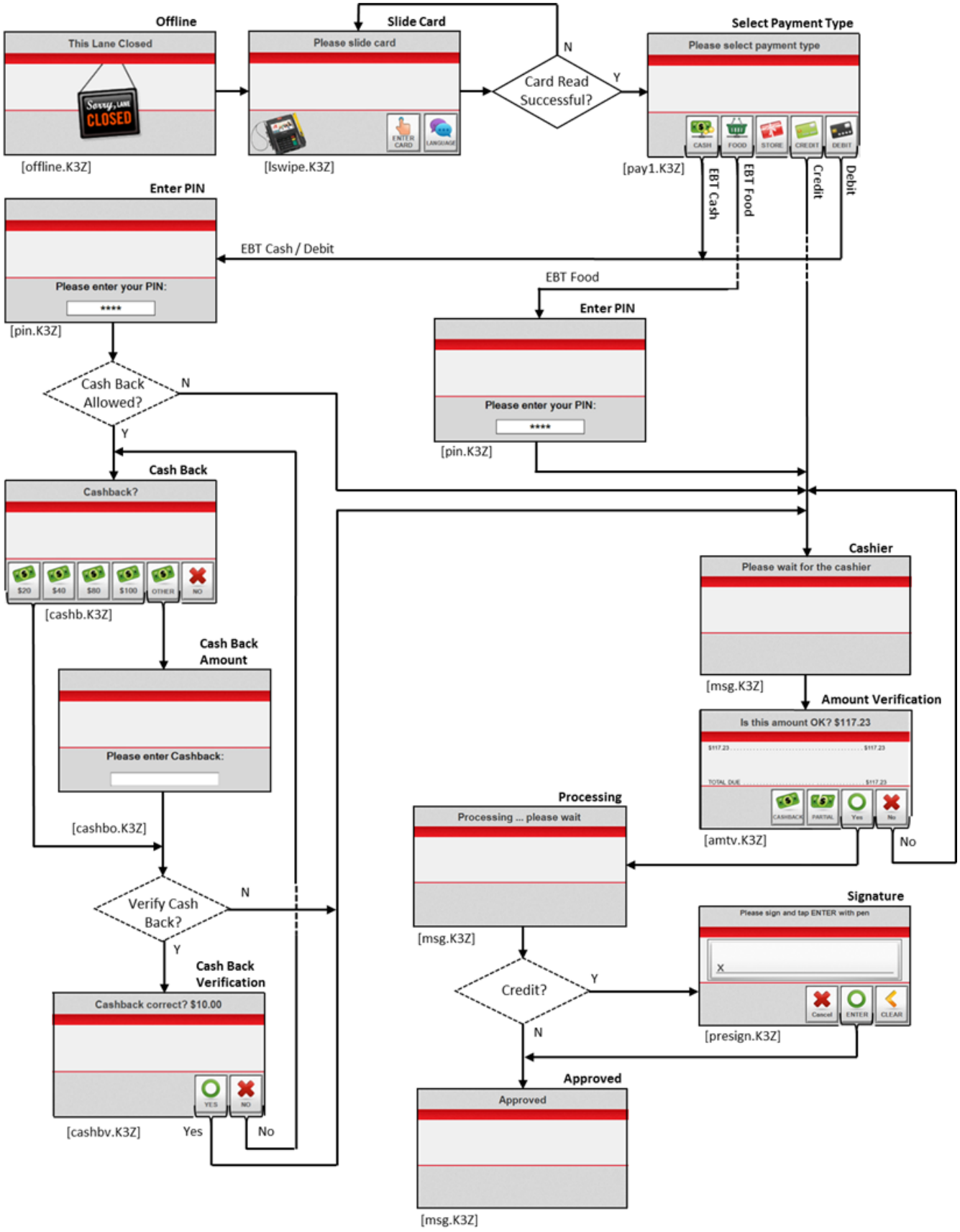
- The terminal remains at “Slide Card” until it reads data from a card swipe or receives the account number and card expiration date as entered from the POS, if the card cannot be read. It then collects the remaining required data at the terminal and builds the proper VISA authorization request message.
- 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.

2_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:


Offline
This Lane Closed
Sorry, Lane CLOSED
[offline.K3Z]
Slide Card
Please slide card
ENTER CARD
LANGUAGE
[Iswipe.K3Z]
Card Read Successful?
N
Y
Select Payment Type
Please select payment type
CASH
FOOD
STORE
CREDIT
DEBIT
[pay1.K3Z]
EBT Cash
EBT Food
Credit
Debit
Enter PIN
Please enter your PIN:
****
[pin.K3Z]
EBT Cash / Debit
EBT Food
Enter PIN
Please enter your PIN:
****
[pin.K3Z]
Cash Back Allowed?
N
Y
Cash Back
Cashback?
$20
$40
$80
$100
OTHER
NO
[cashb.K3Z]
Cash Back Amount
Please enter Cashback:
[cashbo.K3Z]
Verify Cash Back?
N
Y
Cash Back Verification
Cashback correct? $10.00
YES
NO
[cashbv.K3Z]
Yes
No
Cashier
Please wait for the cashier
[msg.K3Z]
Amount Verification
Is this amount OK? $117.23
$117.23
$117.23
TOTAL DUE
$117.23
CASHBACK
PARTIAL
Yes
No
[amtv.K3Z]
No
Processing
Processing ... please wait
[msg.K3Z]
Credit?
Y
N
Signature
Please sign and tap ENTER with pen
X
Cancel
ENTER
CLEAR
[presign.K3Z]
Approved
Approved
[msg.K3Z]

2_3 On-Demand

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 the above 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 behavior 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'

Offline On-Demand Transaction Recommendations

When performing an offline on-demand transaction, there are a few recommended deviations from standard transaction procedure.

The 00.x: Offline Message should be used for resetting an on-demand card read instead of the typical 15.6 Soft Reset Message.

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 in-between offline and on-demand transactions (potentially including a customer's company logo, messages, etc.) or
- An acceptable interstital screen before issuing the next 24.x or 30.x messages, or resuming any offline ads, if configured to do so.

See also Communication Messages for additional information.

2_4 Spin the BIN - BIN Lookup Process

The RBA 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's 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 which starts with 6011xxxxx may be a Discover card, while a 4000xxxx account may be a Visa card.

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

- Internal Spin the BIN (STB). The payment is established by the RBA from data included in the RBA 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 RBA 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, RBA displays the Invalid Card Type prompt, resets the payment, resets the payment, and then goes back 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 seven files, `bin0.dat` through `bin6.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 out to the POS.

2_4_1 How to Enable BIN Checking

RBA has the ability to automatically establish a payment type for the customer card. This can be done by using values from the RBA local configuration (internal Spin the BIN - STB), or by asking the POS to select the payment type (external STB).

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

- If enabled, the application goes to STB.
- If it is not enabled, the RBA 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 account's first digits are used for comparison.

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

2_4_2 Internal BIN Range Checking

Internal STB means that the payment selection is based on the local RBA 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'.

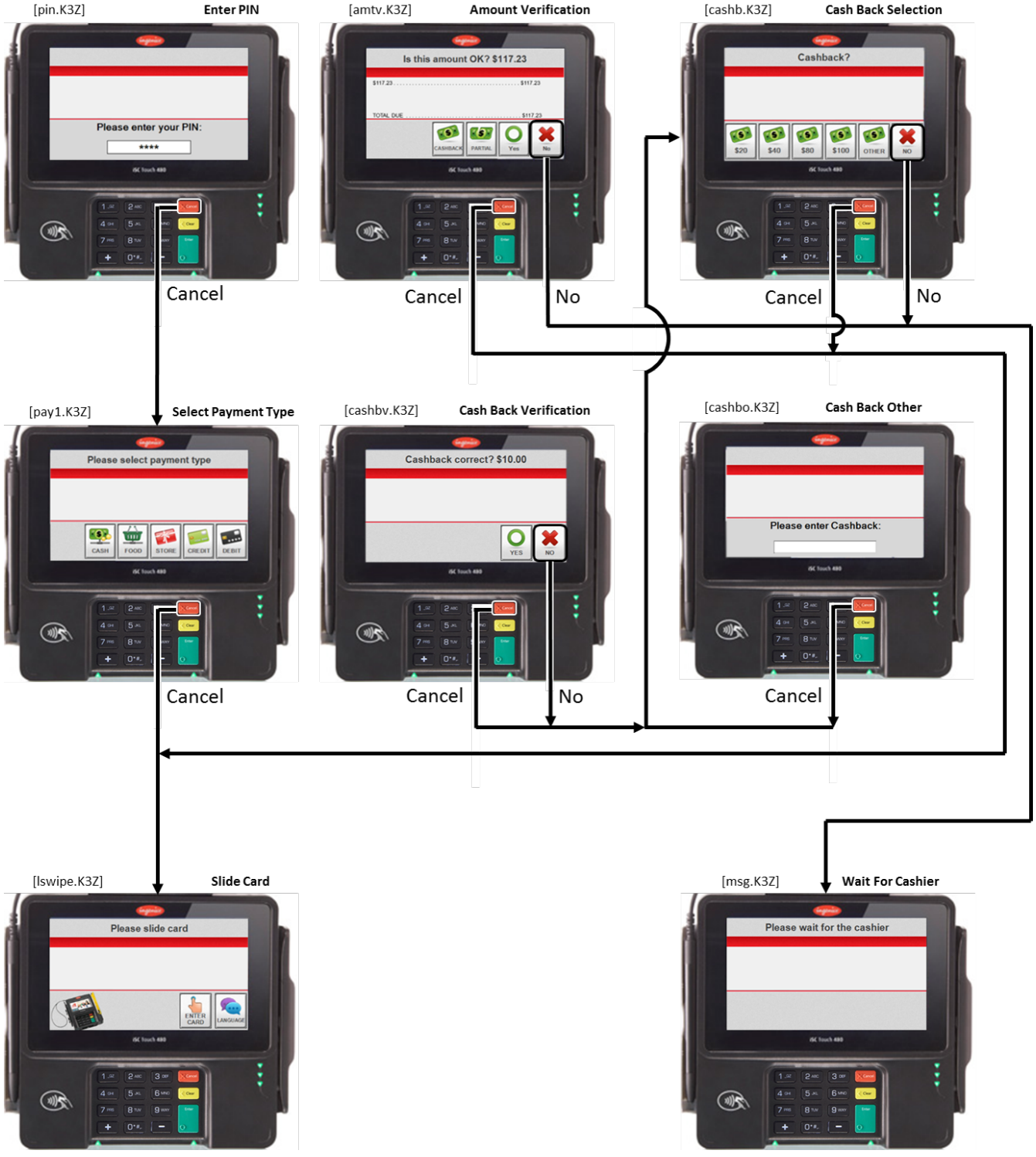
2_4_3 External BIN Range Checking

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

See 19.x: BIN Lookup for more information.

2_5 No or Cancel Process

The following sections describe how the RBA handles a CANCEL key press in various processes. Refer to the following diagram which illustrates the RBA's standard cancellation process.



RBA's Standard No or Cancellation Process

2_5_1 Amount Verification

When the NO key is pressed during the verify amount state, the RBA always sends out 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 RBA sends out the 10.x message. The RBA goes back to the transaction start and the transaction is cleared along with the language selection.

2_5_2 Cash Back

The cash back 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 cash back selection and goes to the next RBA process
  - CANCEL key press: If '0007_0004' = '0', return to swipe.K3Z, otherwise return to lswipe.K3Z
- `cashbo.K3Z`: Screen to enter specific cash back 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`

2_5_3 PIN Entry

When the CANCEL key is pressed during PIN entry, the RBA 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 RBA 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, RBA checks if the purchase amount is present in the terminal.
  - If the purchase amount is present, the RBA sends out 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 RBA clears the transaction along with the language selection, and goes to the transaction start.

2_5_4 Signature

The CANCEL button on the Signature forms for the iSC250, iSC350 and iSC480 terminals is now functional prior to signing (pre-signature) only. Once signing is initiated (post-signature), the following occurs:

- 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.

2_5_5 Transaction Cancelled

If the CANCEL key is pressed and the transaction conditions are such that the transaction is terminated, the Transaction Cancelled message may be 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 1/10th of a second).

If the CANCEL key is pressed during the Language Selection, Card Swipe, or Payment Selection process, the RBA will execute the following processes:

- If the purchase amount is present, the RBA sends out 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 was received, the RBA clears the transaction, returns the language selection to the default value, and goes to the transaction start.

2_5_6 Cancel after amount received

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

2_5_7 Cancel no amount

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

2_6 Transaction End Process

The financial transaction ends in following situations:

- As a normal part of the RBA flow.
- On the POS’s request when one of the following messages is received by the RBA:
  - 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 RBA detects a special condition, which may be:
  - The cardholder pressed the CANCEL key. After that, if the amount is received, the RBA sends out a 10.x message.
  - Some of the configuration parameters are not present in the terminal, without these, RBA cannot operate normally.
  - At the end of the signature on-demand execution.

In this process, the RBA 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 RBA transaction and go to the offline state.

When the financial transaction is cleared, the RBA makes the following changes.

- 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 may be informed 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 1/10th of a second and only in effect if advertising is on). The prompt displays for five seconds. Next RBA may do one of the following 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 RBA resets the transaction.

2_6_1 Configuring

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

- Once 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 1/10th of second.
- After the Host Response message has timed out, RBA 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.

It is important to 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'.

2_6_2 Response Messages

RBA 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.

2_7 Telium Terminal Information

The VID and PID settings required for USB communications with Ingenico Telium terminals have been compiled into a table which addresses both CDC and HID modes. In order to establish USB communications with an Ingenico terminal, a POS terminal 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 Telium terminal communication settings.

2_7_1 VID and PID Settings for HID and CDC Communications

Since 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
iPP350	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
iUP250	Ingenico iUP250	128x64	Black /white	No	No	128m	Buzzer	0x0B00	0x0057	0x0B00	0x0076
iWL250	Ingenico iWL250	320x240	240k	No	No	128m	Buzzer	0x0B00	0x0064	NS	NS

2_7_2 Communication Settings

Refer to the Telium Download User's and Developer's Guide for more information on setting up the communication type for your terminal.

Possible communication types include:

- RS-232 (19200, 8, N, 1)
- RS-232 (115200, 8, N, 1)
- USB HID (all terminals except the iUP250 and iWL250)
- USB<>SerialConv
- MagicBox (iSC250, iSC480, iPP320 and iPP350 only)
- Ethernet (DHCP)
- Ethernet (Static)
- Tailgate

For information on Bluetooth settings, refer to the **Bluetooth Setting**s section of the**Telium Download User's and Developer's Guide**. Included in this section is information on the following:

- Associating the iWL250 Terminal with Bluetooth Cradle/Base

- Bluetooth Pairing for the iWL250 and iSMP Companion
- Installing the Bluetooth Pass-Through Service Utility

Also refer to the **Configuring Device Communication Settings** section of the **Telium Download User's and Developer's Guide** for information on the Jungo Driver.

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

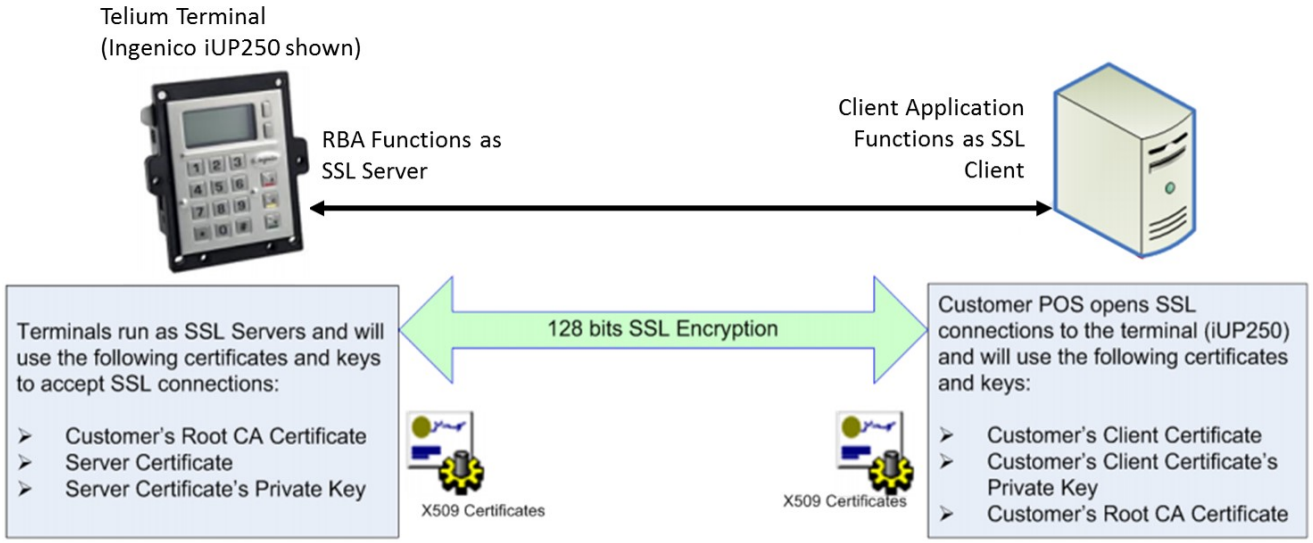
2_7_2_1 SSL/TLS for Ethernet

SSL (Secure Sockets Layer) handshake and encryption using the more updated, client-authenticated TLS (Transport Layer Security) handshake, also known as mutual authentication, may be used with Telium RBA.

In the Telium RBA implementation, 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 opposite side. Therefore, during installation, both sides should install the root certificate from the other party. During the handshake, both the client and the serve will use the root certificate to validate the certificate presented by the opposite side.

The list of certificate and key requirements for implementing SSL/TLS follows the Figure "Telium RBA's SSL Connection," shown below.

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



Telium RBA's SSL Connection

SSL Implementation Requirements

A set of requirements is outlined in the following table, including certificates, private keys, and a configuration on the PC or tablet hosting the Bluetooth Pass-Through Service Utility.

SSL Implementation Requirements

Requirement	Environment	Purpose	Description
Customer's Root CA Certificate	Ingenico Terminal	To validate the certificate presented by the client during handshake.	Usually only 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.
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.
Turn on SSL	PC on which the Bluetooth Pass-Through Service Utility was installed	To turn on/off SSL.	Refer to "Installing the Bluetooth Pass-Through Service Utility" section in the Telium Download User's and Developer's Guide.
Set SSL Protocol Version Identifier		To select the SSL protocol version.	TLS version 1.1 or 1.2 must be selected. RBA is no longer supporting SSLv3. 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 SSL sequence events that occur during a handshake, see also Wikipedia's page on Transport Layer Security.

2_7_2_2 iCMP and iSMPc Bluetooth Pairing and Unpairing

iOS Bluetooth Pairing

This pairing mode is to only be used with iOS devices. If the iCMP or iSMPc was previously paired with a "Standard" (non-iOS) Bluetooth device, the terminal will automatically reboot to allow it to pair to an iOS device.

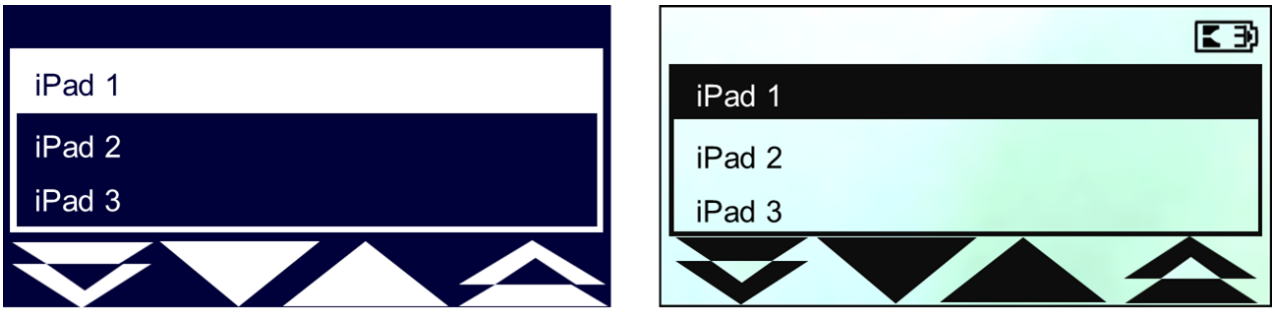
1. Ensure the terminal is powered on and that the iOS device has Bluetooth connectivity enabled.
2. The terminal should be at the "Bluetooth Pairing Required" screen (shown below). If not, the terminal must be unpaired (see Bluetooth Unpairing).

iSMPc

iCMP

3. To begin the pairing process select the "iOS" key (F1) on the iCMP or iSMPc.
    a. Some iSMPc terminals are configured to only support one type of Bluetooth pairing. In this case, the iOS and Standard options that are pictured above will be replaced with a single option that reads "Begin".

4. The terminal will automatically display all Bluetooth enabled iOS devices in the immediate area:



iSMPc iCMP

a. Use the [F2] and [F3] keys to scroll up and down, respectively, through each displayed Bluetooth device.

b. Use the [F1] and [F4] keys to page up and page down, respectively.

5. Highlight the Bluetooth device the user intends to pair with and press the [Green] key:

6. The terminal will display an eight digit randomly generated pairing PIN:

iSMPc iCMP

7. The host device will prompt for a pairing PIN. The user should enter the generated PIN into the prompt and select “Pair” on the iOS device.
8. The iOS device will display the logical name of the iCMP or iSMPc and show the status of the pairing process.
    a. Due to the validation and exchange of secure credentials, the iOS device may cycle through “Connected” and “Not Connected” statuses.

Standard Bluetooth Pairing

This pairing mode is to only be used standard Bluetooth devices.

1. Ensure the iCMP or iSMPc is powered on and that the Standard Bluetooth device has Bluetooth connectivity enabled.

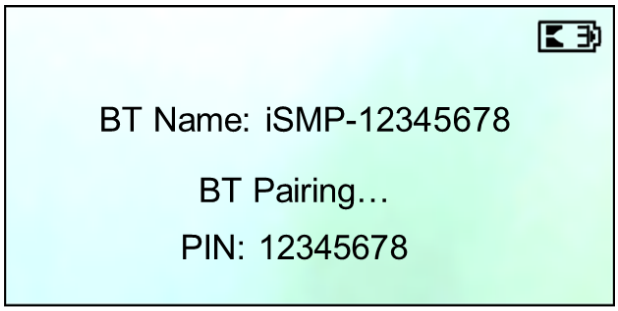
2. The terminal should be at the “Bluetooth Pairing Required” screen (shown again below), if the terminal is not at that screen then the terminal must be unpaired (see section Bluetooth Unpairing).

iSMPc

iCMP

3. To begin the pairing process select the “Standard” key (F2) on the iCMP or iSMPc.
    a. Some iSMPc’s are configured to only support one type of Bluetooth pairing. If this is the case then the iOS and Standard options that are pictured to the right will be replaced with a single option that reads “Begin”.
4. The terminal will go into discovery mode and display an eight digit randomly generated pairing PIN with the iCMP or iSMPc’s unique Bluetooth name:

iSMPc



iCMP

5. On the standard Bluetooth device, search for the terminal’s logical Bluetooth name that is on the screen of the terminal and select it to pair.
6. When the standard Bluetooth device prompts for a PIN, enter the PIN that is displayed on the terminal screen.

Bluetooth Unpairing

To unpair the iCMP or iSMPc from the host or tablet simply press the ‘Function’ key four (4) times in under two (2) seconds:

The terminal will beep and then go to the BT Pairing Required screen.

Troubleshooting

> Unpairing via scanning a barcode has been depreciated and unpairing via the above section is preferred.

If the Barcode Scanner will not power on for unpairing process:

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

4. Once the terminal has finished booting, the barcode scanner will be enabled so that the user can finish the unpairing process.

If the terminal continuously prompts the host device and the user to enter a Bluetooth PIN and the unpairing process has been completed then the user should follow the steps below.

1. Ensure that the terminal has been forgotten on the host device.
2. Turn the host device Bluetooth connectivity off.
3. Reboot the terminal as illustrated above.
4. Once the terminal has finished booting, turn the host device Bluetooth connectivity back on, ceasing Bluetooth PIN prompting.

2_7_2_3 Communications Supported per Terminal Model

The below table shows the interfaces supported by each device, as they appear in their respective menus:

Supported Communication Types by Device

Device(s)	Supported Communication Types
iPP320, iPP350, iSC250, iSC480	• "Serial" • "Ethernet" • "USB-HID" • "USB<>Serial Conv" • "Tailgate" • "MagicBox Serial"
iSC350	• "Serial" • "Ethernet" • "USB-HID" • "USB<>Serial Conv" • "Tailgate"
iUP250	• "Serial" • "Ethernet" • "USB<>Serial Conv"
iWL250, iMP350, iMP352, iCM122	• "USB<>Serial Conv" • "Bluetooth"

From either the "Communication" or "Select Comm. Type" menu, pressing the CLEAR button three times will change the menus to also include interfaces not supported by the device.

3_Functions

3_1 Advertising Support

RBA supports server-based advertising on Telium terminals. Please contact your Ingenico Representative for additional information.

3_2 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.

3_2_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.

3_2_2 Non-Standard Cards

Non-standard cards are acceptable in both the on-demand card swipe request and the RBA flow.

Here is the difference between standard cards and non-standard cards.

- 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 4-digit expiration date. The format of the date is YYMM, where YY means the 2-digit year and MM means the 2-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.

The 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 '=' character or expiration date present (non-standard card), the Track 2 expiration date is set to '0000'.
    - If the message contains the '=' character followed by the expiration date (standard card), that 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 RBA displays the payment selection followed by the "Invalid card" prompt, and goes back to the transaction start.

Swiping a non-standard card can sometimes cause a a card read error or a reboot. With parameter '0003_0018' enabled, the RBA will append Track 2 data instead of account number for some non-standard MSR cards in both 23.x and 50.x messages. The POS handles entry of the CVV and expiration date.

3_2_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.

3_2_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 section.

3_3 Scrolling (Digital) Receipt

3_3_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.

3_3_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.

3_4 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.

3_4_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.

3_5 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.

3_5_1 Configuring Signature Retrieval

3_5_1_1 Standard

Signature capture is a standard feature for the iSC250, iSC350, and iSC480 terminals. When using the off-the-shelf RBA, the RBA functions are programmed to execute in a specific order. The signature function is part of it. When the execution of the signature is finished, the 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.

3_5_1_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, the RBA can send out to the POS an unsolicited the 20.x message to inform it that the signature data is ready to be retrieved. Sending out 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 will send out 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, the 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 the 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, the RBA config prompt is used instead. When the signature is accepted, the 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 out, if available. The current process is continued.
- When signature on-demand is successfully finished, the RBA displays “Signature accepted” for three seconds and then goes to Transaction End. After that, based on the RBA configuration selection, it may 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.

3_5_2 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.		
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`.	29.10000 700		

Sequence	Message	POS	Terminal
Return first signature block, `700`.	29.20000 700 {data}		←
Request second signature block, `701`.	29.10000 701	→	
Return second signature block, `701`.	29.20000 701 {data}		←
29.x messages continue until the final message in the signature block...	...		
Request last signature block, `70` (x-1) .	29.10000 70(x-1)	→	
Return last signature block, `70` (x-1) .	29.20000 70(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.

3_5_3 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.

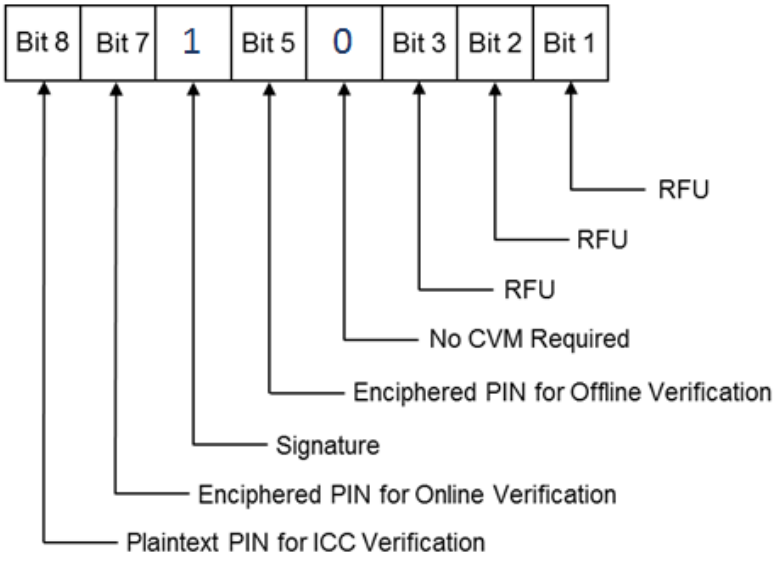
3_5_4 Disabling Electronic Signature on iSC Series Terminals

3_5_4_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.

3_5_4_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.



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.

3_6 MSR Encryption

3_6_1 Supported Encryption Methods

The following table shows the encryption methods which are supported by the RBA Application:

Encryption Methods' Support Status

Encryption Method	Support Status	'0091_0001' Enable Encryption Value ("n")
Magtek MagneSafe™ POS	Will not be supported by Telium (see Note below).	1
On-Guard	✓	2
EPS	✓	3
Voltage TEP1	✓ (Cannot be used with TailGate)	4
Voltage TEP2	✓ (Cannot be used with TailGate)	5
Voltage TEP3	Not yet supported.	6
Monetra CardShield	✓	7
Mercury Payment Systems (MPS)	Will not be supported by Telium (see Note below).	8
RSA-OAEP	Supported using KP4 key only (see Note below).	9
TransArmor	Supported using KP4 key only (see Note below).	10
TDES DUKPT Generic	✓	11
S1	✓	12
EPS P2PE	✓	

Note

Non-KP4 key encryption will no longer be supported by Telium. As such, encryption methods which use only non-KP4 keys will no longer be supported. Encryption methods which use both KP4 keys and non-KP4 keys will, however, continue to be supported by Telium provided that only KP4 keys are used.

Encryption or masking of cards with PANs containing less than 9 digits is not supported (minimum 12 for Voltage encryption types). Merchants should either whitelist these cards or disable non-standard card encryption.

3_6_2 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	Requirements Document
EPS P2PE	• Track 1 or Track 2 must be read successfully. • PAN must include at least 9 characters.	EPS (Element Payment Systems) P2PE Encryption
Magtek	• Track 1 or Track 2 must be read successfully. • PAN must include at least 9 characters. • Mod-10 Checking must verify PAN.	Magtek Encryption
Mercury Payment Systems	• Track 1 or Track 2 must be read successfully and contain data. • Read track must include sentinels. • Read track must include PAN with correct length. • PAN must include at least 9 characters. • At least one KSN must be defined in parameter '0091_0002'.	Mercury 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

Encryption Method	Minimum Requirements	Requirements Document
S1	• Track 2 with both account number and expiration date field • Minimum 3 bytes of discretionary data.	S1 Encryption
TDES DUKPT Generic	• 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
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	• PAN must include at least 12 digits. • Track data must include at least one complete PAN.	Voltage TEP1 and TEP2 Encryptions
Voltage TEP2	• PAN must include at least 12 digits. • PAN must be successfully read from track data or manually entered data.	Voltage TEP1 and TEP2 Encryptions
Voltage TEP3	Not yet supported by RBA.	

3_6_3 Enabling MSR Encryption in RBA

RBA users can enable MSR encryption by adjusting the `config.dfs` parameters in the below table found in the `security.dat` file (see also section Security Parameters (security.dat)):

Parameters for Enabling MSR Encryption

config.dfs Parameter	Description	Notes
0091_0001	Set value to “n” as noted, above in the table in Supported Encryption Methods	The former parameter was '0003_0012'.
0091_0002	Set value to key slot index that contains the data encryption key (if applicable).	Some encryption types do not use this parameter (e.g., Voltage). Former parameter was '0003_0006'.

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.

3_6_4 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

3_6_4_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: 401234 **000** 8909
		`19.D1100003401234000890 9[FS]`
		`4B99358C793844C31AC522C764CC5A676`
		`C3DFEE935`D522CB613051F2554A9D3B87C09BE4E1A55896E44AB21F4FDA82
		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: 401234 **000** 8909
		`23.04`
		`B99358C793844C31AC522C764CC5A676C3DFEE935D522CB613051F25`
		`54A9D3`B87C09BE4E1A55896E44AB21F4FDA828D248F3AE1D1025F3AC935C
		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: 401234 **000** 8909
		Track1:
		`29.200004064`
		`B99358C793844C31AC522C764CC5A676C3DFEE935D522CB613`
		`051F2554A9D3B87C09BE4E1A55896E44AB21F4FDA828D248F3AE1D1025F3`
		`AC935CDB33D1A1AD1:FFFF9876543210E00004`
		Track2:
		`29.200004071164C984EBF0C3FED6A2047073608535C68A1BA050DDB73AA`
		`FA03`DCC276CAB15:FFFF9876543210E00004

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: 401234 **000** 8909
		Track2:
		`50.12345678901234567890123456789012345678900207003254800002`
		`@D1`
		`164`C984EBF0C3FED6A2047073608535C68A1BA050DDB73AAFA03DCC276CA
		B15:FFFF9876543210E00004[FS]1@[FS]1025[FS]

3_6_4_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/^120912100000067800000
	Encrypted Track 1	5D97BDCBC8F9E12F4C99D502FB9B30E7715DBC0C8D64C27BE868554F24F176733C3798B96 76A7D68A0F3BFF8256E484AB65A88B4B2C4AB50DFA60B09585B9D57
	KSN	A08B000C000003000023
	Encrypted Track Format	EncryptedTrack1Data:KSN1
Track 2	Original	4447340101127648=12091210000067800000
	Encrypted Track 2	CEAFC2FD0BA3E76E0C062DD2DD4196E111A71424C52561E943142AD271FC0D86C45CC365 B8D7E292
	KSN	A08B000C000003000024
	Encrypted Track Format	EncryptedTrack2Data:KSN2
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/^13060000000123000000?`
	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.

Track	Type	Example
	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	5EFDACAD0F0B6997EC120D2AE568EDD504A50214F8871E6C43D3A4CFDC30D4BC:FFFF9876543210E00003
	KSN	FFFF9876543210E00003
	Encrypted Track Format	EncryptedManuallyEnteredData:KSN Manually entered data will appear as Track 2 data.

3_6_5 Generic TDES DUKPT Encryption

Generic TDES DUKPT Encryption (Triple Data Encryption Algorithm with Derived Unique Key Per Transaction) is an encryption method for Retail Based Applications. A unique key is used for each transaction. This section assumes the reader is familiar with these products. It contains information specific to the Generic TDES DUKPT P2PE feature. 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. Currently, 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.

3_6_5_1 TDES DUKPT Configuration

Configuring Security Parameters

Configuration information for Generic 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. Refer to Configuring the Application in this manual for details. The specific security parameters relevant to Generic TDES DUKPT encryption are listed in the below 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 hardcoded for Generic TDES DUKPT.
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 hardcoded 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 hardcoded for Generic TDES DUKPT.

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.

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.

3_6_5_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.

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^141250254321987123 45Q?
- 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=14125025432198712345?
- 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=121200000012340000? 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^141250254321987123 45Q?;4445222299990007=14125025432198712345?

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.

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)
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.

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.

3_6_6 Magtek Encryption

3_6_6_1 Overview of Magtek Encryption

Magtek is a format-preserving encryption type. Please refer to the Magtek MagneSafe Technical Reference Manual (Magtek part number 99300001-1) for detailed information about the implementation of Magtek MagneSafe encryption. The following table describes the configuration parameters relevant to Magtek encryption.

Configuration Parameters (in config.dfs) Used for Magtek Encryption

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption	0091_0001	0	Specify this value as '1'.
Specify Encryption Key Slot (Key Index)	0091_0002	6	Valid values include 0-9 for most encryption types. • Magtek requires a value other than ‘0’ (zero). This value must match the slot where the key was injected.
Configure Leading PAN Digits in the Clear	0091_0003	6	Magtek does not allow configuration of this parameter.
Configure Trailing PAN Digits in the Clear	0091_0004	4	Magtek does not allow configuration of this parameter.
Masking the PAN	0091_0012	0	Specify the character to use for masking the PAN. • Magtek uses only ‘0’ (zero).

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.

3_6_6_2 Data to be Encrypted

At least one present and valid Track is required for encryption to occur.

- If there is no Track 1 data, Track 2 data is required, and a simulated copy of Track 1 data will be created and used to formulate the Magtek message.
- Likewise, if there is no Track 2 data, Track 1 data is required, and a simulated copy of Track 2 data will be created and used for the message.

The following table shows all of the scenarios along with format codes required by Magtek.

Magtek Encryption Scenarios and Format Codes

Tracks Present in Input	Format Code
Tracks 1 & 2 present	E
Only Track 1 = present, Track 2 = blank	I
Only Track 1 = present, Track 2 = error	J
Only Track 2 present, Track 1 = blank	G
Only Track 2 = present, Track 1 = error	H

To learn more about the specific constraints on the content of each track, the specific pieces of data that are ciphered, and the varying formats for each (which are dependent upon PAN length), please refer to the Magtek MagneSafe Technical Reference Manual (Magtek part number 99300001-1).

3_6_6_3 Configuring Mod-10 and Account Number Check Digit

Magtek encryption requires Mod-10 checking. Refer to the Mod-10 Checking section for more information. The following table provides an example Mod-10 check digit using manual entry.

Example Mod-10 Check Digit Using Manual Entry

PAN Length	Track	Example Check Digit
16 Digits	Track 2	`;5452300003007180=120410100000000000?`
15 Digits	Track 2	`;545230000307180=120410100000000000?`

This example shows that for cards with 16 digit and 15 digit PANs, the 10th character always holds the Mod-10 check digit correction character.

Encryption Does Not Occur If…

Any errors result in sending Track 1 and Track 2 data in the clear. Please refer to the Magtek MagneSafe Technical Reference Manual (Magtek part number 99300001-1) for detailed descriptions.

Encryption Data Returned to the POS – Tracks 1 & 2

The encrypted data will be sent to the POS using all existing messages or the standard RBA flow. For example, the encrypted data will be sent to the POS using the 23.x message, if conducted through the 23.x message.

Output formats for Tracks 1 and 2 depend on the length of the input PAN. Please refer to the Magtek MagneSafe Technical Reference Manual (Magtek part number 99300001-1) for detailed information.

Track 3 Data

Magtek takes into account Track 1 and Track 2 only. Because there is no way of knowing what information Track 3 may contain, Track 3 will not be returned when using Magtek encryption.

Initiating an Encrypted Swipe/Tap/Manual Entry

Magtek encryption is used in both the standard RBA flow and with the 23.x message.

Configuring for Manual Entry – Standard RBA Flow

Both the Expiration Date and CVV are required by Magtek.

To support manual entry using the standard RBA flow, the global flag '0007_0029' must be set to ‘1’ (Display button, prompt for the Expiration Date and the CVV). Refer to the following table which describes this parameter setting and function.

Configuring Prompt for Manual Entry

Value	Description	'0007_0029' configuration
1	Display button, prompt for the Expiration Date and the CVV	REQUIRED

All valid '0007_0029' parameter values are acceptable for Magtek encryption. Values for this parameter are as follows:

- 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).

Note that when the terminal is configured for ‘payment selection before swipe/tap/manual entry’ ('0007_0005' = '1'), the settings (in cards.dat) for prompting the CVV and Expiration Date for manual entry for the selected card type must be set to ‘1’. This is because the expiration date and CVV are controlled based on card type rather than the above global flag ('0007_0029'). Refer to the '0011_xxxx' parameters in config.dfs for details on card type settings.

3_6_6_4 Configuring for Manual Entry – 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
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).

3_6_7 Mercury Encryption

3_6_7_1 Configuration Parameters (in config.dfs)

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption	0091_0001	0	Specify this value as '8'.
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	Mercury does not use this parameter.
Configure Trailing PAN Digits in the Clear	0091_0004	4	Mercury does not use this parameter.
Masking the PAN	0091_0012	0	Specify the character to use for masking the PAN. Mercury uses either * (asterisk) or 0 (zero). For Non-ISO cards, the PAN will not be masked and both Track 1 and Track 2 will be sent in the clear.

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.

3_6_7_2 Data to be Encrypted

All three tracks should be available for Mercury encryption. Each uses some specific formatting, described below:

- Track 1 and Track 2 data will include the start and end sentinels.
- Track 3 data will include the start and end sentinels, as well as the LRC check character.
- Track 1 and Track 2 will be encrypted independently of one another, i.e., there will be a separate KSN for each track.

The Track 1 and Track 2 KSNs can (but do not necessarily) match.

When both Track 1 and Track 2 are present and valid, the format of the data fed to the ciphering algorithm will be as follows:

Algorithm for Valid Tracks 1 and 2

Track Number	Algorithm
1	%Btrack1? (for ISO cards)
1	%track1? (for non-ISO EBT cards)
2	;track2?

When only Track 1 is present and valid, the format of the data fed to the ciphering algorithm will be as follows:

Algorithm for Valid Track 1

Track Number	Algorithm
1	%Btrack1? (for ISO cards)
1	%track1? (for non-ISO EBT cards)
2	;?

Encryption of non-ISO EBT cards with no Format Code (B) in Track 1 is supported.

When only Track 2 is present and valid, the format of the data fed to the ciphering algorithm will be as follows:

Algorithm for Valid Track 2

Track Number	Algorithm
1	%?
2	;track2?

3_6_7_3 Encryption Does Not Occur If…

At least one present and valid conforming track must be read for encryption to occur. Data will be sent in the clear only if:

- Neither Track 1 nor Track 2 has been read successfully
- Neither Track 1 nor Track 2 contains data
- Neither track conforms (there are missing sentinels, the PAN is too long or too short, etc.)
- Any combination of the above for both tracks

3_6_7_4 Other Reasons

If there is a missing key (e.g., '0091_0002' is configured incorrectly), or the ciphering algorithm returns an error, track data will not be sent. Instead, one of the following will occur:

terminal Responses

Form Displayed on Terminal	Terminal Response
Swiping from the swipe screen...	The terminal will go offline.
Swiping from the card read request screen (23.x message)...	The terminal will issue an encryption error.

3_6_7_5 Encryption Data Returned to the POS

Using the 29.x message: If card data is read from the swipe screen as part of the standard RBA flow, the data will be available to the POS using the 29.x: Get Variable Request message. The following table suggests variables to use with the 29.x message in this case.

29.x Message Variables

Masked Track Number	Algorithm
1	Use variable `406` with the 29.x message.
2	Use variable `407` with the 29.x message.
3	Use variable `401` with the 29.x message.

The masked Track 1 and Track 2 data will be sent along with the KSNs and Track 3 data. Track 3 will be formatted as follows:

encryptedtrack1:track1KSN:encryptedtrack2:track2KSN:track3

There are colons (:) separating the encrypted tracks and KSNs, and Track 3 data (if present) will be in the clear.

3_6_7_6 Initiating an Encrypted Swipe/Tap/Manual Entry

Mercury encryption is used in both the standard RBA flow, and with the 23.x message.

3_6_7_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
2	Displayed	Yes	Yes	No
3	Displayed	Yes	No	Yes
4	Displayed	Yes	No	No

3_6_7_8 On-Demand Transactions

If card data is read from the 23.x card read request screen, the encrypted data will be sent to the POS using the 23.x: Card Read Request (On-Demand) message. Below is an example of data sent in a 23.x message for an on-demand transaction with Mercury encryption enabled:

```
23.0%B476173******0010^CARDHOLDER VISA ^1012201000000000000?[FS]
476173******0010=1012201************?[FS];4761730000000010=10122010000000000000?[FS]
BD303C7853EE862C35B764D51238081BFFB1AF53EB6C6422649F12BB4DC63B99E8EFEE08A1F24B4F71DFC0C6D5939F
21111010000000000001:
40F758A5B2FE69012DAAA62CB8F2E750512EF077A5D60D7DF084BEEC58DB70908BA1E9FF39BFEF29:
21111010000000000002
```

The breakdown of the above data sample can be viewed in the table below. Note that there is no Track 3 data sent in this message, as it would be appended to the end after another colon:

Mercury 23.x Card Read Request Breakdown

Data	Value
Message ID.	23.
Message status.	0
Masked Track 1 Data.	%B4761730000000010^CARDHOLDER VISA ^10122010000000000000?
Field separator.	[FS]
Masked Track 2 Data.	;4761730000000010=10122010000000000000?
Field separator.	[FS]
Encrypted Track 1.	BD303C7853EE862C35B764D51238081BFFB1AF53EB6C6422649F12B B4DC63B99E8EFEE08A1F24B4F71DFC0C6D5939F6E585D16EA8477 F4097F829EBBB1F19699FFCC5665DA654640
Colon.	:
KSN1.	21111010000000000001
Colon.	:
Encrypted Track 2.	40F758A5B2FE69012DAAA62CB8F2E750512EF077A5D60D7DF084B EEC58DB70908BA1E9FF39BFEF29
Colon.	:
KSN2.	21111010000000000002

The decrypted track data from the above table is as follows:
Track 1: B4761739001010010^CARDHOLDER VISA ^1012201512349999999
Track 2: 4761739001010010=1012201512349999999

3_6_7_9 Card Swipe and Contactless Transactions

With Mercury encryption enabled, the following data will be sent in a 50.x: Authorization Request message for card swipe and contactless transactions:

```
50.12345678901234567890123456789012345678900208020443900001@D;
4761730000000010=10122000000000000000?[FS]1@[FS]1025[FS]
```

The breakdown of the above data sample can be viewed in the table below:

Example Mercury Encrypted 50.x Message Breakdown

Data	Value
Message ID.	`50.`
Acquirer BIN.	`123456`
Merchant Number.	`789012345678`
Store Number.	`9012`
Terminal Number.	`3456`
Merchant Category.	`7890`
Merchant Country Code.	`123`
Merchant ZIP Code.	`45678`
Time Zone Difference.	`900`
Transaction Code.	`20`
Terminal Serial Number.	`80204439`
Index Code.	`0`
Transaction Number.	`0001`
Message Status Code.	`@`
Account Data Source.	`D`
Masked Track 2 Data.	`;4761730000000010=10120000000000000000`
Field Separator.	`[FS]`
PIN Block.	`1@`
Field Separator.	`[FS]`
Transaction amount, omitting decimal. In this case, '1025' equates to 10.25.	`1025`

Data	Value
Field separator.	[FS]

The decrypted track data from the above 50.x message can be found using the Decrypt Card Data tool with the option 'Data Variant" unchecked (deselected).

3_6_8 Monetra CardShield Encryption

3_6_8_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'.
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.

3_6_8_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.

3_6_8_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?

3_6_8_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.

3_6_8_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.

3_6_8_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

3_6_8_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
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).

3_6_8_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.

3_6_8_9 Monetra CardShield Encryption Examples

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**: %B 601108*****3514 ^INGENICO/TEST CARD ^0705101**********? **Track 2**: (BLANK) **Track 3**: 212A38115DC56F04850BEB5E91014F401A277713FAB73BE3C66C4893 9BF7289A3CC4154DD23A145D00CD035852CFCFA97EECE48FF9658F8F47F27A 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

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>Sample 29.x Requests and Responses<table><tr><th>29.x Request</th><th>29.x Response</th></tr><tr><td>29.00000398</td><td>29.20000398444522000000007</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>

3_6_9 On-Guard Encryption

3_6_9_1 Overview

This section provides information about Telium RBA support for On-Guard point-to-point encryption. On-Guard encryption utilizes 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)
- Smart Card Reader (SCR) for EMV cards
- Manual entry

The purpose of the E2EE solution is to isolate the POS system (Electronic Cash Register or Host device) from processing clear text card data, and in doing so, reduce the impact of PCI DSS reviews. New commands or modifications to existing commands will be described in the New RBA Messages for On-Guard and KME Encryption section of this document.

On-Guard encryption and KME encryption can be enabled, disabled, and configured by a signed configuration file loaded into the SYSTEM drive on the Telium 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. Once the RBA reads and parses this file, it will be deleted. The configuration extracted from this file will be saved into the application disk. As an expected change to the general-release version, the E2EE configuration information will be incorporated into the existing “security.dat” file described in the Security Parameters (security.dat) section of this document.

An E2EE 'activate' command has been added to simplify 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. It should be noted that once E2EE encryption is enabled by either configuration file or by command, it cannot be disabled. The RBA 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 will be defined in the existing “secbin.dat” file described in the RBA documentation.

Refer to the following sections for more information pertaining to On-Guard encryption:

- On-Guard Configuration
- On-Guard Card Data Encryption Rules
- New RBA Messages for On-Guard and KME Encryption
- Handling of Existing RBA Messages
- E2EE Card Data Encryption

3_6_9_2 On-Guard Configuration

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,
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
```

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.

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.

3_6_9_3 On-Guard Card Data Encryption Rules

This section describes how data will be formatted prior to E2EE encryption.

Type A Formatting

Type A formatting applies to KME and will be described in a separate document.

Type B Formatting

Type B formatting applies to On-Guard encryption. This section describes how card data will be formatted for On-Guard encryption. 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.

1. For Track 2 data read from a swiped or MSD card, the data to be encrypted will be the Track 2 data converted to ASCII.
2. For chip card data (contact or contactless) with Track 2 Equivalent Data (EMV tag `T57` or equivalent) available and converted to ASCII format, the data to be encrypted will be the ASCII data with a separator (‘=’ , hex 0x3D) inserted after the PAN if not already present.
3. For manually captured cardholder data, the data to be encrypted will be the PAN followed by the expiration date (YYMM format) in ASCII. After the data is encrypted, the ASCII character ‘M’ will be added to the beginning of the buffer, and a separator (“=”) will be added at the location 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.

4. For chip card data (contact or contactless) with Track 2 Equivalent Data (EMV tag `T57` or equivalent) not available, the data to be encrypted will be the PAN. Once the PAN is encrypted, a separator (“=”) will be added after the PAN, followed by the encrypted Application Expiration date in YYMM format (taken from EMV tag `T5F24`), followed by the encrypted Service Code (tag `T5F30`).

3_6_9_4 New RBA Messages for On-Guard and KME Encryption

The following messages have been added to support On-Guard and KME encryption. Please note that some of these messages are subject to change in the general-release version.

- 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
- 87.x On-Guard and KME Card Read Data

Other new message types have been implemented as part of On-Guard and KME encryption support. They will not be described in detail here because they are not generally necessary for On-Guard use, and they are subject to change in the general-release version. These include:

- 82.x On-Guard and KME Session Key Injection Command
- 88.x On-Guard and KME Translate Encrypted Card Data Command
- 89.x On-Guard and KME Register BIN Record Command

Additionally, the existing 31.x: PIN Entry Messages (On-Demand) has been modified to accommodate the E2EE encryption feature.

3_6_9_5 Handling of Existing RBA Messages

Message Handling with E2EE Encryption Enabled

With E2EE enabled, a different set of messages will be used in place of existing RBA messages for certain functionality. As an expected change to the general release, messages will be more consistent between E2EE and non-E2EE configurations. The following table describes the alternate message process which occurs when E2EE is enabled.

Alternate Process for On-Guard E2EE

Message	When E2EE is Enabled
12.x: Account Message	This message command is disabled and this message will be ignored.
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	The 86.x On-Guard and KME BIN Lookup (PIN Encouragement) Message is sent in place of this message. A 19.x message from the POS will be ignored.
23.x: Card Read Request (On-Demand)	The 87.x On-Guard and KME Card Read Data should be used in its place. The 23.x card read request message is disabled and will return an invalid command response with the error code '9' (declined).

31.x PIN Entry Command Message in E2EE Mode

The customer account number is included in the 31.x: PIN Entry Messages (On-Demand). When E2EE encryption is enabled, the clear text PAN is not available to the POS. It must instead supply the AES encrypted PAN in the Customer's Account Number field,and the key type field must be set to "m" or "d".

The AES PAN is the result of the encryption of the card's PAN using an internally generated key. Accordingly, the 31.x message cannot be used separately. The encrypted AES PAN value must be extracted from the encrypted data sent by the terminal. Refer to the 31.x: PIN Entry Messages (On-Demand) section for a description of the 31.x message with E2EE incorporated.

3_6_9_6 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:

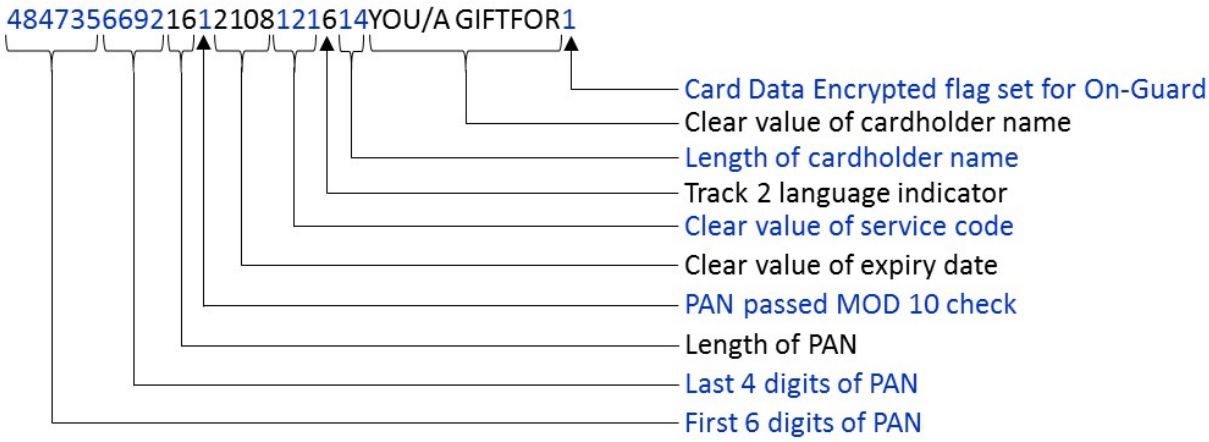
- Exit type = '0' indicating a good card read.
- Card Data = ' 484735669216121081216 14YOU/A GIFTFOR1

  BFFFF98765432292000050114328438595666498708=00823303648413132DE47

  E75CB39344F75AA4B82BDFF0AA2264683140DF3D6C473E2C74A7BE9B8B

  59D5A11512B5361AB781A382B768E69BE9C02 '

FOR MSR and contactless MSD, the data in this example is formatted as follows:



MSR and Contactless MSD Data Format

With On-Guard encryption enabled, the following data is formatted as described in the below table for Data Sent with On-Guard Encryption Enabled.

B FFFF98765432292000050114 32 8438595666498708=008233036484131 32 DE47E75CB39344

F75AA4B82BDFF0AA22 64 683140DF3D6C473E2C74A7BE9B8B59D5A11512B5361AB781A

382B768E69BE9C0 2

Data Sent with On-Guard Encryption Enabled

Data	Value
Specifies IngeCrypt data format.	B
DUKPT Key Serial Number (KSN) + custom field.	FFFF98765432292000050114
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

3_6_9_7 E2EE Card Data Encryption

If E2EE encryption is enabled, the RBA application will encrypt the card data which in turn will be transmitted to the Host for decryption and processing. The format of the Track 2 field in the 50.x: Authorization Request message will be modified. E2EE encrypted data will replace the MSR data normally incorporated at offset 62 in the 50.x message. The encrypted card data will vary, 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

MSR and Contactless MSD in E2EE Mode

For MSR and contactless MSD, data will be formatted as follows:

MSR and Contactless MSD Format

Data	Length	Type	Description
PAN, first 6 digits	6	ASCII	Clear value of the first 6 digits of the PAN.
PAN, last 4 digits	4	ASCII	Clear value of the last 4 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. • 1 = Encrypted ASCII data.

Note 1

When Track 1 and Track 2 are requested with E2EE enabled, "Service code", "Language code", and "Expiry date" will be filled out from Track 2 data only. If only Track 1 data is available, then these fields will 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 will be zero.

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 will be sent:

Remaining Fields with Card Data Encrypted Flag set to 0 for Clear ASCII Data

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.

If the Card Data Encrypted flag is set to '1', then the following fields will be sent:

Remaining 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	24	ASCII	DUKPT Key Serial Number (KSN) + custom field.
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.

Manual Card Data Entry in E2EE Mode

For Manual card data entry, the data will be formatted as follows:

Manual Entry Data Format

Data	Length	Type	Description
PAN, first 6 digits	6	ASCII	Clear value of the first 6 digits of the PAN.
PAN, last 4 digits	4	ASCII	Clear value of the last 4 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	'000'
Language code	1	ASCII	D'0'
Card Data Encrypted flag	1	ASCII	• 0 = Clear ASCII data. • 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 will be sent:

Manual Entry Remaining 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 will be sent:

Manual Entry Remaining 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	24	ASCII	DUKPT Key Serial Number (KSN) + custom field.
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.
Extended Language code	1	ASCII	First digit of Track 2 card language indicator.

EMV Contact and Contactless in E2EE Mode

When an EMV transaction is in process (contact or contactless), the EMV '33.03.x' Authorization Request Message will be used in place of the 50.x Authorization Request message. In this message, the following tags will be substituted with encrypted values:

- T5A (PAN)
- T57 (Track 2 equivalent data)

Additionally, a new Ingenico-specific tag (`TFF1D`) will be added.

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 encrypted value. There are currently multiple EMV library commands which return contact and contactless Track 2 Equivalent Data (tag `T57`), PAN (tag `T5A`), and Expiry Date (tag `T5F24`) in the absence of tag T57. If E2EE encryption is enabled and the BIN is not present in the validated BIN table, then the Track 2 data will be encrypted using the appropriate E2EE encryption key. The EMV tag `TFF1D` for the E2EE Cryptogram will be created using the format outlined in the following table.

Creating Tag TFF1D

Data	Length	Type	Notes
PAN, first 6 digits	6	ASCII	Clear value of the first 6 digits of the PAN.
PAN, last 4 digits	4	ASCII	Clear value of the last 4 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. • 1 = Encrypted ASCII data.

Note 1

When Track 1 and Track 2 are requested with E2EE enabled, "Service code", "Language code", and "Expiry date" will be filled out from Track 2 data only. If only Track 1 data is available, then these fields will 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 will be zero.

The remaining fields depend on the value of the Card Data Encrypted flag. If this flag is to '0' for clear ASCII data, then the following fields will be sent:

Manual Entry Remaining Fields with Card Data Encrypted Flag set to 0

Data	Length	Type	Notes
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.

If the Card Data Encrypted flag is set to '1', then the following fields will be sent:**Manual Entry Remaining Fields with Card Data Encrypted Flag set to 1**

Data	Length	Type	Notes
Encrypted Format Type	1	ASCII	B = IngeCrypt (IC) data format.
IC KSN	24	ASCII	DUKPT Key Serial Number (KSN) + custom field.
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.

EMV Tags in E2EE Encryption Mode

When E2EE encryption is enabled, the following applies to EMV tags:

Tags affected by On-Guard E2EE

Tag	Tag Name	With E2EE Encryption Enabled
TFF1D		Always included.
T57	Track 2 equivalent data	All zeros.
T5A	PAN	If present: • First 6 digits and last 4 digits will be in the clear. • All other digits will be 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.

3_6_10 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. The following section applies to both encryption types unless noted otherwise.

See http://en.wikipedia.org/wiki/RSA_(algorithm) and ftp://ftp.rsasecurity.com/pub/rsalabs/rsa_algorithm/rsa-oaep_spec.pdf for details of the algorithm.

The current Telium implementation can be used with MSR and contactless data. Using the 23.x message to request manually-entered card data is also supported.

For both encryption types, the public key consists of a 2048-bit modulus and an exponent which is normally ‘010001’. These values need to be 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.

3_6_10_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

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.
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. Currently ignored for RSA-OAEP. • 1 = Track 1 • 2 = Track 2 (Default)

Parameter Name	DFS Data Index	Default Value	Description
Terminal ID (in `store.dat`)	0004_0013	12345678	For TransArmor, the Terminal ID to be used in the encrypted data block. Must be 8 digits or fewer. Currently not needed for RSA-OAEP.

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.

3_6_10_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.

3_6_10_3 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

<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.

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>Sample 29.x Requests and Responses<table><tr><th>29.x Request</th><th>29.x Response</th></tr><tr><td>29.00000398</td><td>29.20000398444522000000007</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>

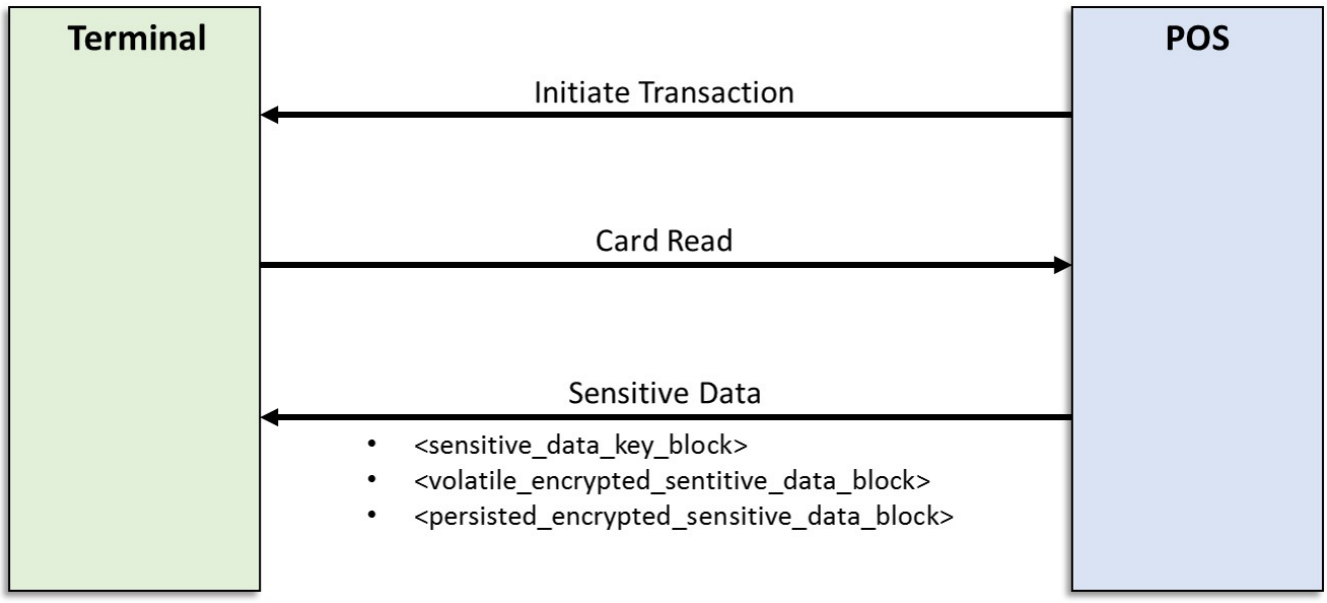
3_6_11 S1 Encryption

3_6_11_1 S1 Encryption Overview

S1 encryption is a format-preserving encryption method implementing Authenticated Encryption Mechanism 5 in [ISO19772].

3_6_11_2 S1 Encryption Process

The following diagram illustrates the S1 P2PE encryption process.



S1 Encryption Flow

This P2PE protocol provides data origin authentication on all encrypted messages transmitted from the terminal to the host. This is implemented by appending a secure MAC (Message Authentication Code) to encrypted data sent to the host. Once this occurs, the host is required to validate the secure MAC prior to initiating decryption.

Refer to the above diagram for the S1 Encryption Process. Once the initial card information has been read by the POS, the POS will query the terminal for sensitive information. The terminal will then transfer two distinct encrypted data blocks to the POS, accompanied by a sensitive data key block which specifies key attributes used by the host to identify or derive the decryption key. These encrypted data blocks include PCI stipulations which are read by the POS and are used to determine if certain card information is to be preserved (e.g., full track or discretionary data), as sensitive card information is grouped into persistent and volatile data blocks. The terminal compares the card BIN to a preloaded S1 whitelist to determine whether P2PE is mandated for the card type, and determines what obfuscation is to be applied to unencrypted card information returned to the POS.

3_6_11_3 Configuration Parameters (in config.dfs)

There are several encryption-related parameters and parameter files in the Telium RBA Developer's Guide which are used to enable or configure S1 P2PE encryption. Additionally, there are several other encryption-related parameters and parameter files used by other encryption types that are not used for S1 P2PE. See the below diagram for a description of the relevant parameters for S1 encryption in `config.dfs`:

Parameters used by S1 Encryption

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption (in `security.dat`)	0091_0001	0	
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 - 6. • Only slots 0 - 5 can be used.
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).

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.

3_6_11_4 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.

3_6_11_5 Load Whitelist

The S1 Whitelist is contained in an .XML file named `S1LIST.XML`. This file must be used when implementing S1 encryption. This .XML file also contains all of the S1 P2PE whitelist data, but in a user friendly and modifiable format. The `S1List.XML` file must be packaged and then signed by Ingenico 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.

Load this .PGZ file to the HOST as you would normally load other signed .PGZ files. From here, the .XML file will be unpacked automatically to its secure location.

3_6_11_6 Card Swipes During the Standard Flow

Support for S1 encryption has been added for card swipes during the standard flow card swipe screen. Previously, only card swipes using the 23.x: Card Read Request (On-Demand) message were allowed with S1 encryption selected. Card swipes performed from the standard flow card swipe screen resulted in errors which prompted the terminal to go offline. With S1 encryption now supported in the standard flow, this is no longer an issue and this action is no longer being processed as an error.

3_6_11_7 Working with Sensitive Data

After the card is read, a 23.x response message is sent to the POS. Sensitive data is stored and then transmitted in the Track 3 portion of the 23.x response message. Elements of sensitive data are concatenated together, separated by ‘:’s (colons). If any Track 3 data exists on the card, this data is obfuscated and then appended to the end of the sensitive data following a ‘:’ (colon).

Track 3 will be formatted as follows:

SensitiveDataKeyBlock:VolatileEncryptedSensitiveDataBlock:PersistedEncryptedSensitiveDataBlock:Obfuscatedtrack3DataFromCard

The sensitive data, as with any binary data sent in an RBA message, must be in hex-ASCII format in accordance with the RBA message protocol.

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 ISO4909 Track 2 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.

3_6_12 Voltage TEP1 and TEP2 Encryptions

3_6_12_1 Overview of Voltage TEP1 and TEP2 Encryption

Voltage TEP1 encryption is whole-track encryption, while TEP2 encryption is structure-preserving encryption. Refer to Configuration Parameters (in `config.dfs`). There are several encryption-related parameters and parameter files in the Telium RBA which are used to enable/configure Voltage P2PE. Additionally, there are several other encryption-related parameters and parameter files used by other encryption types that are not used for Voltage P2PE.

3_6_12_2 PAN Encryption

When Voltage TEP2 encryption is enabled, the PAN must contain a minimum of 12 digits. Up to 6 leading digits as well as the last 4 digits will be preserved and sent in the clear. For PANs containing 14 or more digits, the number of encrypted digits will be the PAN length less 10. As an example, a PAN containing 15 digits will have 5 digits encrypted. Similarly, a PAN containing 16 digits will have 6 digits encrypted. In all cases, a minimum of 4 digits including the Luhn value will be encrypted. PANs embedded in Track 1 and Track 2 data are processed similarly. The track embedded PAN length will be shorter than the original PAN length, but the leading and trailing digits will be preserved. The Luhn check will generate the same result as the plaintext version. The additional track data is encrypted using Base64 alphabet and is 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 will be 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 will always be the same length as the PAN entered using the terminal keypad. The following tables summarize Voltage TEP1 and TEP2 encryption of the PAN for a swiped card.

Voltage TEP1 Encryption of PAN for Swiped Card

Message Type	After Voltage TEP2 Encryption
19.x: BIN Lookup	• 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)	• All digits of the PAN are encrypted.
29.x: Get Variable Request	• All digits of the PAN are encrypted.
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	• The encrypted PAN length included in the message is less than the original PAN length. • The first 6 and the last 4 digits of the PAN are sent in the clear. • The remaining middle digits of the PAN are encrypted.	5444009999222205 544400 806 2205
23.x: Card Read Request (On-Demand)	• The encrypted PAN length included in the message is less than the original PAN length. • The first 6 and the last 4 digits of the PAN are sent in the clear. • The remaining middle digits of the PAN are encrypted.	5444009999222205 544400 806 2205
29.x: Get Variable Request	• The PAN length is preserved, all digits are included in the message. • The first 6 and the last 4 digits of the PAN are sent in the clear. • The remaining middle digits of the PAN are encrypted.	5444009999222205 544400 411407 2205
50.x: Authorization Request	• The encrypted PAN length included in the message is less than the original PAN length. • The first 6 and the last 4 digits of the PAN are sent in the clear. • The remaining middle digits of the PAN are encrypted.	5444009999222205 544400 806 2205

Refer to Voltage TEP1 and TEP2 Encryption Examples for examples of TEP1 and TEP2 encryption. Also refer to the following table which 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

Parameter Name	DFS Data Index	Default Value	Description
Enable Encryption (in `security.dat`)	0091_0001	0	Specify this value as: • '4' for Voltage TEP1. • '5' for Voltage TEP2.
Max Number of Transactions with Same Key (in `security.dat`)	0091_0005	0	Maximum number of transactions with the same key. • 0 = Don’t change keys based on transaction count. • 1 - 65000 = Change key after this many transactions with the same key.
Periodically Change Keys (in `security.dat`)	0091_0006	0	Periodically change keys (Requires setting the terminal’s date and time). All letters must be entered 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 XXth 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 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 Name	DFS Data Index	Default Value	Description

Parameter Data Encoded in base64 (in `security.dat`)	0091_0011		Sample data – not for production. Actual data must be obtained from your authorizer.
			
			`AkkI5lUTQBRABHW+Q8WeFmGOQOXIe6Av2WvRhsSVLm1BWKza4u/1JCnS6dPW5L`
			`1pC8xd6H`
			`/7Uc0vAWh8Na07Ge5lsfCW3wbPRDouotjVd7QJoODLqe1CGQpNJSN/`
			`Wbx5j3E5iYnWuEhSifDqy9M2zZzXWAHdzd6vkRrFkLRyavYGUsaJa4fzg1y/BE`
			`OYWaURa9I4uUD0exrG4VWrRdQ5GWJGQ3oNQC`
			`/U9Bd6FaOxfnAKttNGiidLN5mN`
			`B6ZxjuBGMY8jM3WxJKvmceyyIW18t4ddUILrTptQ2ZU1BkoZjlbJvLuggdGv2l`
			`DZP4CMtznrvC0c5II8QaoRD7+fIpfYZZR+789TqSo3YeI5BB5rdv9iR/ul0ct/`
			`+DOUJSUERhVjTps2N5sJ5BX4OJy3dnD`
			`/BDiLulQK2VBLtiJ1QNJyPmfZPUBGT5`
			`xGlMrbXw4e5WX87bmc9jAkXjZkgoP5pJxm`
			`/MXuD1vtCcNVCrFnZ/Qi/5SI/PEv`
			`xxYnZkfCfzWGJM`
			`/PgwA1L2T3VFBfTQeTBLBLTV8us5rsGUce`
			`/nyzuDs4ZFRbOe`
			`qVcWEdrnZT`
			`/GWSUC97YA0AFsxT6GkbFR1qv7OAPIzeyw1W6WpciGC4bgATdeIx`
			`SD76mm+kPy4DwckTFdVKiDdo4N6rIBUjph1eV5`
			`/StpNXQ4vmwTMhbo2v9b+5Wg`
			`cGboiWcAcc8AiuDzeYxZL2iJUY077U3dYKpTsYLZuWaLFYiLuwgdBQ+lsZrTYx`
			`PEjx/bp4JHvxMl8tBtU5l3DRj0rk4H7H/syGDtb7z/////////v//////`
			`//////fPmI+YrYeuVmH8NJpSgBYtALj7Vu2ypnWI65c0CPZRCWlYxkCtTybOIG`
			`ej5cYKTxLFyP12yWn017VRN+NBnozJlF6nMoOEw+GWdSzo7kpi+Et8Yyt0LEA+`
			`BPu7vGvUFUmD67v21PL7i+UYjHMN7erBhbGIB+e8ywafDBEQqaIOFza3libHVl`
			`LnZvbHRhZ2UuY29t`

Masking Character	0091_0012	0	Character to use for masking. • For '0091_0012', only '0' and '*' are valid.

Parameter Name	DFS Data Index	Default Value	Description
Parameter Data Encoded in TransArmor Key.	0091_0013		Parameter data encoded in base64 TransArmor key. Sample data: `MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvy YRQA3cS4wV9yk+6b FzA7KLDmE+D /SOP+Q5bNOPG9nUDkAPalRBz12KA5SDxTw2vO1BIeSFUQl YTpzE Db/XkfNNm5e6nqf12M4kdHP1F2EXW4WArilUZegAVw /Y7FvAkA8PQFbfgmBirS a5GS /fuAHjemqEf0DxIgq552IDeFw3nB0vccK6ePue5sVB9Sm2 vWpKm/lj2UE4 P6z2ngZr5V31cSAVN08USxHvz+MEnoUBKt6aKvfRUAp4iF yIpxlp4eylxY8ziz PekS29lcRMsI9hGug2CoKFhhUJ1gD8G280zIoWCxysNvl4 0k/l8OTtPKrnlhAz QcIyy/RB0lwb6QIDBlU1`
Exponent for TransArmor	0091_0014	010001	Exponent for TransArmor. • This value in binary is equivalent to '65537'. Do not change this value.
Map to Public Key	0091_0015	12345678901	This is the map to the Public key. • Length should be 11 bytes.
Track 1 or Track 2 Selection for TransArmor Encrypted Data Block	0091_0016	2	For TransArmor, this value sets the encryption target as Track 1 or Track 2. • 1 = Encryption target is Track 1. • 2 = Encryption target is Track 2. Track 2 should be used in the encrypted data block.
Length of Encrypted CVV in Voltage Encryption	0091_0017	8	For Voltage encryption, this is the length of the encrypted CVV. This is used for manual entry only. • Valid lengths are 7 - 23.
Terminal Communication Authentication Selection	0091_0018	0	Terminal communication authentication selection, '93.' Challenge /Response message authentication. • 0 = Disabled - no additional application-layer authentication. • 1 = On-connection - authentication required after terminal connects (e.g., Bluetooth connect signal/event).

Parameter Name	DFS Data Index	Default Value	Description
Length of encrypted CVV returned by Voltage (in `security.dat`)	0091_0017	8	This parameter is for use with Voltage P2PE types, only. Defines the length of the encrypted CVV returned by Voltage encryption (both TEP1 and TEP2). Used only for manual entry. Valid lengths are 7 – 23 digits. If an invalid length is assigned (e.g., a length of 54), the length will default back to the default of 8 digits.
Terminal Communication Authentication	0091_0018	0	Terminal communication authentication selection. See section 93.x: Terminal Authentication Messages. • 0 = Disabled, no additional application-layer authentication. • 1 = On-connection, authentication required after the terminal connects (e.g., Bluetooth connect signal/event).

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.

3_6_12_3 Obsolete Voltage Parameters

The Voltage parameters outlined in the table, below, have been redefined with new DFS Data Indices and reside in `secbin.dat`.

Parameter Name	Obsolete DFS Data Index	Obsolete Default Value	NEW DFS Data Index	NEW Default Value
Identity String	0091_0001	id@sample.com	'0091_0009' (in `secbin.dat` file)	id@sample.com (no change)
Identity Date	0091_0002	*	`'0091_0010'` (in `secbin.dat` file)	* (no change)
Parameter data encoded in base64	0091_0003	(n/a)	`'0091_0011'` (in `secbin.dat` file)	(n/a)

3_6_12_4 PAN Encryption

Voltage encryption now handles account messages containing only the PAN. The PAN is always encrypted, regardless of whether or not expiry date and CVV are also included in the message. Refer to the following Voltage TEP1 Encryption Examples table and Voltage TEP2 Encryption Examples table.

3_6_12_5 Voltage TEP1 and TEP2 Encryption Examples

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’.

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+gbywafLVwhNuPeiLznET9OsW8zwzfIeSDOV6p[FS]
PAN Only Sent in 12.x Account Message	PAN in clear = 4012345678909 12.4012345678909 50.12345678901234567890123456789012345678900208020453600001@A4012345678909[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=333 50.12345678901234567890123456789012345678900208020453600003@ A1LSakAaNvUH2WG9eX+CPAED4Zaf[FS]1@[FS]12345[FS]

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	5444009999222205	**Encrypted Track** 1: wHcwtWNPmagNhPLrAuITygvL0dbplOGHRKhEUj/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	50.12345678901234567890123456789012345678900202016846600002@D/u2vCFy3nvM/zxim FEeA9CZAqYD[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.

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
PAN Encryption	407714 **742987** 4992 • 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@T40123 1639 8909=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@T40123 1639 8909=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	5444009999222205	544400 806 2205 • 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	544400 806 2205 • 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	544400 411407 2205 • 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	544400 806 2205 • 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 • 50.1234567890123456789012345678900208034438800009@T4445228903600007=1512=10571712[FS]1@[FS]100[FS]
'2'	PAN and expiration date .	• 60110000901023171 • 1512	• 23.0[FS]60110000901023171=1512 • 50.1234567890123456789012345678900208034438800009@T4445228903600007=1512[FS]1@[FS]100[FS]
'3'	PAN and CVV .	• 60110000901023171 • 369	• 23.0[FS]4445228903600007==99236979 • 50.1234567890123456789012345678900208034438800009@T4445228903600007==10571712[FS]1@[FS]100[FS]
'4'	PAN only.	• 60110000901023171	• 23.0[FS]4445228903600007 • 50.1234567890123456789012345678900208034438800009@T4445228903600007[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>Sample 29.x Requests and Responses<table><tr><th>29.x Request</th><th>29.x Response</th></tr><tr><td>29.00000398</td><td>29.20000398444522000000007</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>

3_6_13 MSR Encryption Processing

When MSR encryption is enabled, card data for swiped, manual, and contactless transactions is encrypted immediately after being captured. In the case of keyed transactions, RBA saves the keyed information as Track 2 data before processing. Track 1, Track 2, and manually entered data are all encrypted using a TDES DUKPT algorithm.

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 will have the proper length, but only the first few digits will be correct. `config.dfs` parameter '0005_0008' sets the number of digits.

If MSR encryption is attempted and the operation fails, RBA shuts down and the terminal goes offline.

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/^120912100000067800000?
	Encrypted Track 1	5D97BDCBC8F9E12F4C99D502FB9B30E7715DBC0C8D64C27BE868554F24F176733C37 98B9676A7D68A0F3BFF8256E484AB65A88B4B2C4AB50DFA60B09585B9D57
	KSN	A08B000C000003000023
Track 2	Original	4447340101127648=12091210000067800000?
	Encrypted Track 2	CEAFC2FD0BA3E76E0C062DD2DD4196E111A71424C52561E943142AD271FC0D86C45C C365B8D7E292
	KSN	A08B000C000003000024
Manually Entered	Original Track 1 Data	%M4744750029324780^MANUAL ENTRY/^13060000001230000000?
	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
	KSN	FFFF9876543210E00003

Contactless track data will be read as Track 2.

By default, the first six digits and the last four digits of card numbers will be listed in the clear when encryption is enabled. To modify these settings, parameters '0091_0003' (unmasked leading digits) and '0091_0004' (unmasked trailing digits) can be changed.

This does not apply to Magtek or Mercury encryption. Magtek requires that the first six digits and last four digits be listed in the clear. Mercury uses its own masking convention.

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.

3_6_14 MSR Encryption Data Variables

After the RBA processes a 23.x: Card Read Request/Response message exchange involving MSR encrypted data, three system variables are populated with transaction data:

- Variable `398` – when the setting of '0091_0012' designates a valid encryption type, the Account Number from the previous 23.x card read will be returned. Some of the digits will be masked/encrypted based on the '0091_0003' and '0091_0004' settings.
- Variable `399` – contains the Account Name in the clear.
- Variable `400` – contains the Account Expiration Date in the clear.

Data stored in each of these variables can be retrieved using the 29.x: Get Variable Request.

3_6_15 Retrieving MSR information using the 29.x (Get Variable) Message

The 29.x: Get Variable Request message can be used to retrieve the following pieces of information:

Using the 29.x Message to Retrieve MSR Data

Information Type	Variable	Description

Information Type	Variable	Description
Card Data Encryption Status for Card Request Message	395	Card data encryption status for the 23.x: Card Read Request (On-Demand) message. • 0 = Card data not encrypted. • 1 = Magtek Encryption • 2 = On-Guard Encryption Configuration • 3 = EPS (Element Payment Systems) Encryption • 4 = Voltage TEP1 Encryption • 5 = Voltage TEP2 Encryption • 6 = Voltage TEP3 Encryption (not currently supported). • 7 = Monetra CardShield Encryption • 8 = Mercury Encryption • 9 = RSA-OAEP Encryption • 10 = TransArmor Encryption • 11 = Generic TDES DUKPT Encryption • 12 = S1 Encryption Possible reasons for card data not being encrypted include: • P2P encryption not configured. • Card not found on whitelist. • The card is a nonstandard card which should not be encrypted.
Mod-10 check value	396	Card Read Transaction Flow MOD-10 check value.
MOD-10 check value	397	Card read on-demand calculated MOD-10 check value (23.x message). Available for use with all encryption types, but used with S1 only at this time.
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.

Information Type	Variable	Description
Service Code	413	New service code which is used to determine whether or not the swiped card is an EMV card. This variable is always available for card type verification whether EPS encryption is enabled or not. The service code will typically have a value of '101' or '201'. 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.
Card Data Encryption Status for Standard Flow	414	Card data encryption status for the standard flow. • 0 = Card data not encrypted. • 1 = Magtek Encryption • 2 = On-Guard Encryption Configuration • 3 = EPS (Element Payment Systems) Encryption • 4 = Voltage TEP1 Encryption • 5 = Voltage TEP2 Encryption • 6 = Voltage TEP3 Encryption (not currently supported). • 7 = Monetra CardShield Encryption • 8 = Mercury Encryption • 9 = RSA-OAEP Encryption • 10 = TransArmor Encryption • 11 = Generic TDES DUKPT Encryption • 12 = S1 Encryption Possible reasons for card data not being encrypted include: • P2P encryption not configured, or configured incorrectly. • Card not found on whitelist. • The card is a nonstandard card which should not be encrypted.
Track 1	406	Used to retrieve masked Track 1 data.
Track 2	407	Used to retrieve masked Track 2 data.
Track 3	411	Used to retrieve Track 3 data.

3_6_16 Requesting the PIN Block Using the Masked PAN

A PIN block may be requested using a masked PAN. This is because the terminal stores the account number and keeps a record of the masked PAN that it sends to the POS.

- First, the POS sends the 31.x PIN Request message with the PAN masked.
- The terminal will then compare the masked PAN that it sent to the POS to the masked PAN which you are telling it to encrypt the PIN block with. One of the following will happen:
  - If the two masked PANs match, the terminal will use the account number to encrypt the PIN block
    -or-
  - If the PANs do not match, the terminal will encrypt the PIN block with the masked PAN coming from the POS which will result in a failed decryption.

3_6_17 Mod-10 Checking

The Mod-10 (Luhn) formula is used to verify the account number (PAN) using a check digit. The check digit verifier is included in the RBA 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.

The following Mod-10 check digit variables are always updated:

- Variable number `396` (Card Read Transaction Flow MOD-10 check value).
- Variable number `397` (Card Read On-Demand calculated MOD-10 check value, 23.x: Card Read Request (On-Demand)).

> Refer to the Card Transaction Codes table in BIN Processing (allBins.dat, bin0.dat - bin20.dat) for information about enabling Mod-10.

3_6_18 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.

3_6_19 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.

3_6_20 Signing Requirements for .DAT File Changes

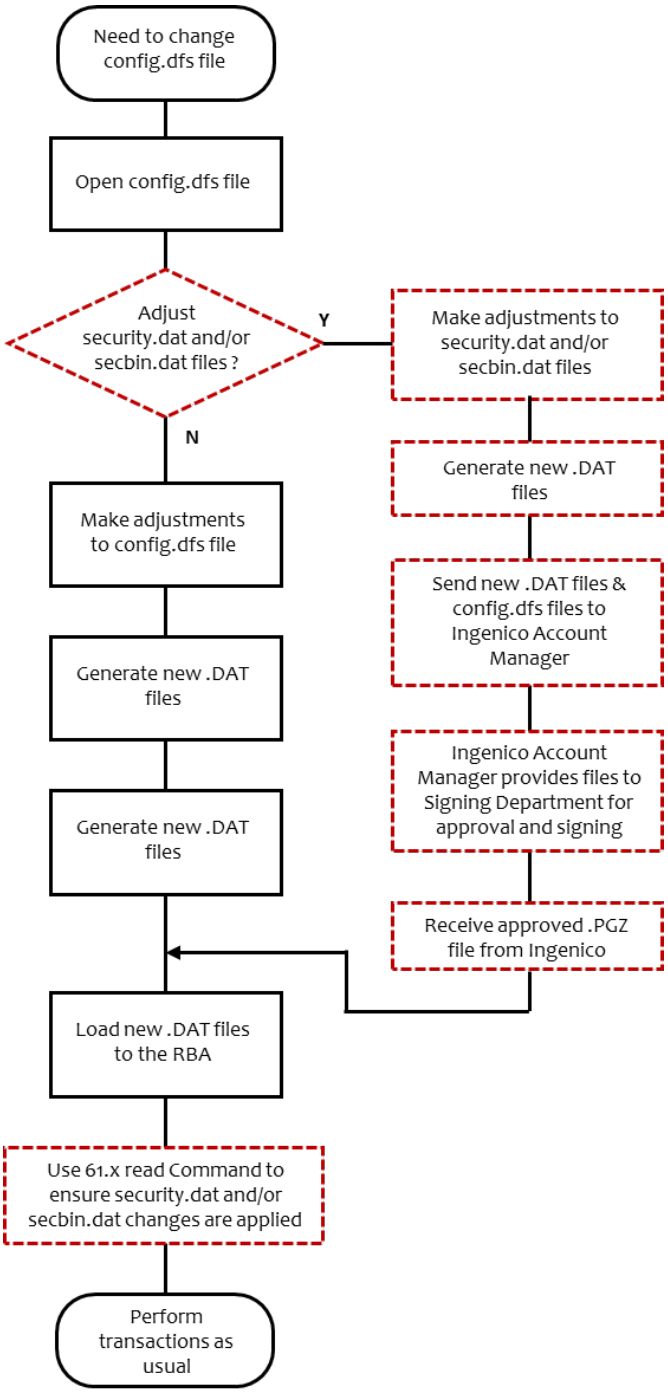
If your organization requires changes to the `security.dat` and/or `secbin.dat` files, you must follow the procedure outlined in this section. Changes to these .DAT files require specific Ingenico signature and approval prior to your implementation.

- **`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.



MSR Encryption Configuration Process

3_7 Call Scheduling

Call Scheduling will be used to download or upgrade firmware and data files at specific dates and times. Refer to the Call Scheduling section in the Telium Download User's and Developer's Guide for more information.

3_8 Contactless Key Card Support

3_8_1 Introduction to Contactless Key Card Support

This section describes the implementation of Contactless Key Card support for Telium Retail Base Applications (RBA). 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). Telium RBA uses these APIs to service Contactless Key Card communications with the Point-Of-Sale (POS). The GRAF APIs are transparent to the POS which only processes RBA messages.

As of this time, the following card types are supported:

- MIFARE Classic 1K
- MIFARE Ultralight
- MIFARE Mini
- MIFARE Classic 4K

Refer to the RBA Low-Level Contactless Key Card Support section for more information on Contactless Key Card Support.

3_8_2 RBA Low-Level Contactless Key Card Support

Telium 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 more more information on the interface and process. Refer to the following sections for more information on RBA Low-Level Contactless Key Card support.

- 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.

3_8_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.			

3_8_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).

3_8_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.

3_9 Google Wallet Implementation

3_9_1 Overview of Google Wallet

Google Wallet is a consumer payment application based on Near Field Communication (NFC) which enables customers to store gift card, Loyalty card and coupon information on mobile phones. Newer Ingenico terminals already include NFC and contactless capability. The following image shows a mobile phone with Google Wallet installed being used to make a payment on an Ingenico iSC250 terminal.

Mobile Phone Google Wallet Application Used for Payment

Refer to the 17.x: Merchant Data Write message for a description of the MID/secret pair injection process. Also refer to the following sections for more detail on payment flow and support modes:

- General Payment Flow
- Supported Google Wallet Modes

3_9_2 Google Wallet Host Interface Messages

- Refer to the 16.x: Contactless Mode Request section for information on setting Google Wallet mode and retrieving data.
- Refer to the 17.x: Merchant Data Write section for a description of the message formats used for injecting MID/Secret pairs into Ingenico terminals.

> All terminals must be equipped with an Ingenico contactless module in order to enable Google Wallet functionality.

3_9_3 General Payment Flow

The payment method for Google Wallet follows a set priority which is described as follows:

1. If gift card data is present in the Google Wallet data, then the gift card track is used for the payment flow regardless of the presence of ISO track data.
2. If no gift card track data is present in the Google Wallet data then ISO track data, if present, is used for the payment flow. An example would be PayPass.
3. If nether gift card track data nor ISO track data are present, then another payment source may be utilized for the payment flow.

3_9_4 Supported Google Wallet Modes

Google Wallet transactions can be processed using any of the following contactless modes:

1. PPSE First - terminal starts in PPSE mode first, then switches to Mifare if needed.
2. Mifare First - terminal starts in Mifare mode first, then switches to PPSE mode if needed.

3_10 Softcard SmartTap Implementation

3_10_1 Overview of Softcard SmartTap

Softcard SmartTap provides customers with a way to use their smartphone to pay and save at participating merchants. Customers can load their credit cards, Loyalty cards, and special offers by merchants onto their smartphone using this feature. When the smartphone is tapped, special offer and Loyalty card information is sent to the Ingenico terminal along with the payment information. This information is then sent by the terminal to the POS and used to adjust the final transaction amount, including applicable discounts. This adjusted amount is returned to the terminal which then proceeds with the authorization request once confirmed by the customer. Payment information from the smartphone is used by the POS to authorize the transaction.

3_10_2 Configuring the Terminal and Sending Softcard SmartTap Data to the POS

The POS uses the 17.x: Merchant Data Write Request message to configure the terminal for Softcard SmartTap transactions. A 17.x: Merchant Data Write Response message indicates if the configuration was successful. When a smartphone is tapped, SmartTap data is sent using a 16.x Contactless Mode Request message. SmartTap data received from the card or smartphone is sent to the POS in two different situations:

Sending SmartTap Data to the POS

Situation	Description
"Please slide card" state	When the smartphone or card is tapped, any Softcard SmartTap data retrieved will be sent as a 16.x response message from the terminal to the POS.
23.x Card Read Request message	Then the smartphone or card is tapped, any Softcard SmartTap data retrieved will be sent as a 16.x response message from the terminal to the POS when the payment information has been successfully read. A 23.x: Card Read Request (On-Demand) message containing track data is sent last.

Refer to the following sections for 16.x and 17.x Softcard SmartTap message format and usage.

- 17.x: Merchant Data Write
- 16.x: Contactless Mode Request

3_10_3 Supported Contactless Modes

The following table lists and describes the supported contactless modes for Softcard SmartTap.

Supported Contactless Modes for Softcard SmartTap

Mode	Function	Description
1	PPSE Only	The terminal starts in PPSE payment mode. Softcard SmartTap is not supported in this mode.
5	1st - PPSE 2nd - Softcard SmartTap	The terminal starts in PPSE payment mode, then switches to Softcard SmartTap mode.
6	1st - Softcard SmartTap 2nd - PPSE	The terminal starts in Softcard SmartTap mode, then switches to PPSE payment mode.
7	Softcard SmartTap Only	The terminal starts in Softcard SmartTap mode. PPSE payment mode is not supported.

Selecting or changing contactless modes is implemented using the 28.x: Set Variable Request and 29.x: Get Variable Request messages with RBA variable `412`.

3_10_4 Erroneous Card Tap

If a card is not correctly read while the terminal is in contactless modes 5, 6 or 7, the terminal will issue a beep and display an error message for a few seconds. After this, depending on the origination of the erroneous read, the terminal will return to one of two screens:

- If the erroneous read occurred during the swipe screen, the terminal will return to the swipe screen following the beep and error message display.

- If the erroneous read occurred during the 23.x: Card Read Request (On-Demand) screen, the terminal will return to the screen prior to this screen following the beep and error message display.

3_11 Offline Remote Key Injection (RKI) Support

3_11_1 Overview

This section describes support of Remote Key Injection (RKI) in the Retail base Application. Offline remote key loading is implemented using a function based on a TDES symmetry key. Offline remote key injection utilizes Ingenico tools and is maintainable using standard SDK support. This implementation utilizes WinKeyfac (an existing key management tool) and modified existing terminal application software. WinKeyfac has been updated and can now read a file containing injected serial numbers. WinKeyfac uses this file to generate a key bundle file (*.rki) which can be used to update those terminals with specified keys. Each entry in a key bundle contains an injected serial number and the information required to update a specified key. Once this file has been downloaded to a terminal, the application can read this file and search for any entries which contain a matching injected serial number and perform the specified update.

3_11_2 Prerequisites/Requirements

- Terminals must have injected serial numbers in order for this feature to work. These terminal serial numbers are the customer serial numbers which have been injected into the terminals.
- Each key must be unique to the target terminal. A unique KSN can be generated using a diversification algorithm with the Customer Serial Number (CSN).
- For PCI V3 compliant terminals, Key Pattern 4 (KP4) must be enabled first. Injecting DUKPT Lite keys is not permitted. A request for injecting these will be declined with a "40" result code.
- The following are required by the financial application supporting offline RKI:
  - Notify the TSA upon power-up using the API to indicate that Offline RKI is being supported.
  - Must recognize the key file bundle which will have the ".rki" filename extension.
  - Must scan the key bundle file for the injected customer serial number.
  - Use the TSA to load each TR-31 key bundle into the appropriate slot as indicated in the key bundle file.
  - Integrated environment:
    - Must provide the host with a mechanism to receive acknowledgement of successful Offline RKI.
  - Standalone application:
    - Must maintain a log of key bundle headers which were successfully loaded, including the date.

3_11_3 Enabling/Disabling Remote Key Injection

Remote key injection can be enabled or disabled by setting the parameter `RKIENABLE` in the signed `RKI.XML` file which is stored in the application directory. If the `RKI.XML` file is not present, then remote key injection must be disabled. If this file is present, a call to `VarGetByInt( "RKIENABLE", &RKIEnable )` will return the value of the `RKIENABLE` parameter. If set to '0' then remote key injection is disabled. If set to '1' then remote key injection is enabled. 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

3_11_4 Offline RKI Process

The Telium System Application (TSA) supports remote key injection, implemented by a third part software application, which loads keys encoded in TR-31 2010 Key Block format. This is referred to as Offline Remote Key Injection, or Offline RKI. In order to implement this process, the Key Block Protection Key (KBPK, also referred to as base key) from which the TR-31 formatted key bundle is generated must first be loaded into the target terminal. An Application Programing Interface (API) has been added to indicate to the TSA that Offline RKI is supported. When the TSA initializes following power-up, it checks for the presence of the KBPK. If present, the TSA will diversify the stored KTK using the SEC_SpecifLoadKey function with the Customer Serial Number in the secret area. Upon completion, the KBPK is deleted as it will then exist in the Key transfer Key location.

The key bundle file will include the name of the serial number input file, and the extension will be changed to ".rki". The input file name should not contain spaces. The key bundle format is as follows:

Injected Serial No.	FS	WKF Key Label	FS	Key Usage for Financial Key Labels	FS	TR-31 Key Bundle

Key Bundle Format

The KBPK can be loaded into a terminal using the following methods:

- The KBPK can be directly loaded using the Telium Key Injection Application (KIA) injection command, mode T4.
    - The injected key automatically replaces the existing key located in the KTK/PIA slot the first time the terminal powers up. This is implemented by the TSA.
- The KBPK can be derived from the existing KTK and injected serial number when an application submits a request to the TSA to enable Offline RKI or to load a TR-31 formatted key block for the first time.
    - The derived KBPK will be loaded into the KTK slot and replace the existing KTK.
- The KBPK can be encoded in the TR-31 format and requested by the third part software application to load it into the designated key location.

After the KBPK key has been loaded as a key transfer key, Offline RKI is recognized by the TSA as enabled and is indicated in the Application Comment field by assigning the key word value "0000000000000DIV". Key Pattern 4 (KP4) will then be enforced by the TSA. The TSA will first verify that the terminal's secret areas are mapped as KP4 when given a key to load. If the secret areas are not mapped as KP4 then all secret areas other than 0xXXYY2130 will be deleted and remapped to KP4.

Once the Offline RKI mode is enabled by the TSA, the following apply:

- Remote Key Injection can be enabled and disabled with the RKIenable.PGZ file; no keys are lost in the process.
- The KTK contains a specialized key which is intended for use by the TSA_LoadKeyFromKeyBlock API as a KBPK. Accordingly, the TSA_LoadMasterKey function can not be used to load a master key into the terminal. An error code will be returned following any attempt to do so.

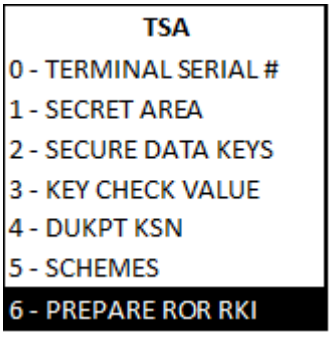
- An external application can inject the following keys by calling the TSA API TSA_LoadKeyFromKeyBlock:
  - CDMK (Application Download Authorization Key, also referred to as Code MAC Key). This key replaces any existing CDMK key.
  - CEFMK (Clear Text Entry Authorization Key, also referred to as the Clear Text Entry Prompt MAC Key). This key replaces any existing CEFMK key. All MAC data generated from the previous CEFMK key is deleted.
  - PEFMK (Secure Text Authorization Key, also referred to as the Secure text Entry Prompt MAC Key). This key replaces any existing PEFMK key. All MAC data generated from the previous PEFMK key is deleted.
  - Master Key. This key replaces the existing Master Key. Its corresponding Session Key is deleted.
  - Session Key. This key replaces the existing Session Key.
  - DUKPT IK. The initial key is reloaded.

Upon successful KBPK injection, the following conditions will be enforced:

- If the key pattern is mode 4 prior to KBPK injection then the key is injected and the terminal remains in KP4 mode.
- If the key pattern is not KP4 prior to KBPK injection then the following actions are taken before injecting the key:
  - All existing keys in secret areas with the exception of ID 0xXXYY2130 are erased.
  - The key pattern is changed to KP4.

3_11_5 Initiating Offline RKI through the TSA

An Application Programing Interface (API) has been added to indicate to the TSA that offline RKI is being supported. From the TSA menu, select option "6 - Prepare for RKI" as shown in the following illustration. When this option is selected, the TSA will delete the system file `RKI.DIA` if it exists. The terminal will display "WILL AUTO RESTART" and initiate a restart.



Initiating Offline RKI

Once the process has been completed, the TSA will remove the RKI indicator from the secret area.

3_11_6 Log Files

The log file is a text file that includes a line for each injection attempt. The log file is readable by users and can also be parsed by an application. Each line in the log file will contain the following information:

- Time stamp for when the installation was attempted.
- Status code. This code provides the result of the injection attempt. It does provide an indication of the key that is currently loaded.
- Human readable message explaining the status code.
- Second and third fields from the original record (e.g., IPPKOPIN).

- For DUKPT keys, include the current KSN (20 digits). For other keys, include the KCV (6 digits).

3_11_7 Functional Limitations

- The maximum .rki file length is 1000 lines.
- Each terminal uses a unique key to transport key bundles. Accordingly, a key set cannot be shared with other terminals.

3_12 Dynamic Update of RSA-OAEP Public Keys

3_12_1 Overview

Dynamic updating of RSA-OAEP Public Keys enables on demand updating of keys without sending the terminal to Ingenico for key injection. To support this feature, the following encryption support message subtypes were added to 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.

Additionally, two new security parameters were added to the Security Parameters (security.dat) configuration file:

- '0091_0032' - Public Key for Signature Verification.
- '0091_0033' - Public Key for Data Encryption.

The '90.5' request message sent from the POS to the terminal includes a key name, key file data, and signature data. Security parameter '0091_0032' (Public Key for Signature Verification) must be set prior to sending the request message. When this message is received, the terminal uses the signature and 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 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 '90.6' request message is used by the POS to delete an encryption public key from the terminal while the '90.7' request message is used to select an encryption public key. 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

To rotate the public key, the following steps are taken:

1. Generate a new public/private key pair.
2. Sign the new public key with the existing initial private key.
3. Send the signed data to the terminal via the 90.x message. The response to this message will indicate if the update was successful.

> This feature is only supported in v12.2.1 and v12.2.2 of RBA.

3_12_2 Procedure to Dynamically Update RSA OAEP Public Keys

1. The customer generates a **Signing** Public and Private key pair.

2. The customer shares the **Signing** Public key with Ingenico.
3. Ingenico returns a `SECURITY.PGZ` file which is a signed file containing the customer's **Signing Public key**.
4. The customer downloads the `SECURITY.PGZ` file to the terminal so that the **Signing Public key** is applied.
5. The terminal is rebooted following successful download of the `SECURITY.PGZ` file.
6. The customer generates an **Encrypting** Public and Private key pair.
7. The customer signs the **Encrypting Public key** with the **Signing Private key**.
8. The customer sends an RBA message to the terminal containing the **Signed Encrypting Public key**.
9. The terminal receives the **Signed Encrypting Public key** from the POS and validates the signature using the **Signing Public key** included in the `SECURITY.PGZ` file.
10. 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.

3_13 Support for Voice Referral for EMV

3_13_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.

3_13_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.

3_13_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		
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).			

Sequence	Message	ECR	Terminal
The POS sends an amount message to terminal.	'13.2000'		→
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.			
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.			

Sequence	Message	ECR	Terminal
The POS sends a hard reset message to the terminal.	'10.'		

4_Host Interface Messages

4_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 Each of these is discussed in the following sections.

As stated in the previous section, there are two types of communication messages that flow between the POS and the terminal: link level responses and data messages. This section defines the format of each type.

4_1_1 Link Level

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 1 byte, where:

- Link Level Response is the following one-character ASCII code:
  - ACK is hexadecimal 06
  - NAK is hexadecimal 15

4_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, as defined below:
- 00.x = Offline Message
- 01.x = Online Request/Response
- 03.x = Set Session Key Message
- 04.x = Set Payment Type Request/Response
- 07.x = Unit Data Request/Response
- 08.x = Health Status Request/Response

- 09.x = Set Allowed Payments
- 09.x = Card Status Message
- 10.x = Reset Message
- 11.x = Status Request/Response
- 12.x = Account Message
- 13.x = Amount Message
- 14.x = Set Transaction Type Message
- 15.x = Soft Reset Message
- 16.x = Contactless Mode Request
- 17.x = Inject MID/Secret Pairs
- 18.x = Informational Card Message
- 19.x = BIN Lookup Message
- 20.x = Signature Request Message
- 21.x = Numeric Input Request/Response
- 22.x = Application ID Request/Response
- 23.x = Card Read Request/Response
- 24.x = Form Input Request
- 25.x = Terms and Conditions Request
- 26.x = Run Script
- 27.x = Alpha Input Request/Response
- 28.x = Set Variable Request/Response
- 29.x = Get Variable Request/Response
- 30.x = Advertising Request Message
- 31.x = PIN Request Message
- 34.x = Save and Restore State Message
- 36.x = Notification of Command Execution
- 40.x = Survey Message
- 41.x = Card Read Message
- 50.x = Authorization Request/Response
- 60.x = Configuration Write
- 61.x = Configuration Read
- 62.x = File Write
- 63.x = Find File
- 64.x = Delete File
- 70.x = Update Form Element Message
- 80.x = MAC Calculation Message Format

- 81.x = MAC Verification Message Format
- 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
- 87.x On-Guard and KME Card Read Data
- 88.x On-Guard and KME Translate Encrypted Card Data Command
- 89.x = On-Guard and KME Register BIN Record Command
- 90.x = MSR Encryption Support Message
- 91.x = Printer Message
- 93.x = Terminal Authentication Messages
- 94.x = Barcode Configuration Messages
- 95.x = Barcode Configuration Messages
- 97.x = Reboot
- 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`

4_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		
	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.

4_2 Communication Messages

This section describes RBA's 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 predetermined sequence, certain messages can also be executed by the host on-demand, which means that they interrupt the normal operation of the application in order to perform the new function requested by the message. The following on-demand messages are available in the RBA:

- 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

For the exact behavior when an on-demand message is received, see the message's corresponding section. Here are some general rules for on-demand functions:

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

⚠ Please note that to adhere to enhanced security standards, dynamic text is not supported in this release of RBA. The POS will use prompt indices to display plain text messages.

4_2_1 00.x: Offline Message

4_2_1_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.

4_2_1_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.

4_2_1_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 RBA 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 RBA 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 RBA 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 RBA Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed [FS] then 11.12
24.x: Form Entry Request (On-Demand)	Nothing is received in the RBA 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 RBA 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

Messages Allowed	Example Response	11.x Status
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 RBA 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 RBA Testing Tool (until you send a Status Request 11.x, then you get 11.10PleaseSign[FS]).	11.00LaneClosed [FS] then 11.15 Advertising[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 RBA 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		
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

4_2_2 0x and 50.x: Authorization Response Message Format

Both the Authorization Response message format and the Alternate Authorization Response message format are defined by the VISA Second Generation specification. Either message is acceptable.

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

0.x Authorization Response Message Format (Received by RBA 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
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.)
M	1	Constant	ASCII control character – FS
M + 1	1	Constant	ASCII control character – ETX
M + 2	1	Binary	LRC check character.

4_2_3 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'.

4_2_4 03.x: Set Session Key Message

The 03.x Session Key message is used to send a new session encryption key to the terminal. It is designed to flow in both directions as a request/response pair, with a 03.x request flowing from the POS to the terminal and the 03.x response from the terminal to the POS. The 03.x request message data consists of the new session key used for data encryption. The 03.x response message contains a 3-character status code indicating the result of the key injection process. The only allowable message response to the 03.x request is a 03.x response.

Although this message can be sent at any time, it is recommended to send it before a new transaction has been started on the terminal. The new session key is then 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.
20	1	Constant	ASCII control character – ETX
21	1	Binary	LRC check character.

4_2_5 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.

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. Allowable values are: • 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.
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. ⓘ 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_REQ_AMOUNT. Transaction Amount. • 3 to 9 characters. 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. If the transaction amount is blank, a 13.x: Amount Message is required for the amount.
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.

Offset	Length	Type	Description
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. 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_RES_AMOUNT. Transaction Amount. • 3 to 9 characters. 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. If the transaction amount is blank, a 13.x: Amount Message is required for the amount.
M	1	Constant	ASCII control character – ETX.
M + 1	1	Binary	LRC check character.

4_2_6 07.x: Unit Data Request

The Unit Data Request Message flows from the POS to the terminal. It is used to request the hardware and software configuration on the terminal.

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	6	Alphanum	iConnectEFT Constant = P07_RES_MANUFACTURE Manufacture ID – “INGNAR”
10	1	Constant	ASCII control character – FS
11	6	Alphanum	iConnectEFT Constant = P07_RES_DEVICE Terminal ID – “iPP320,” “iPP350,” “iSC250,” "iSC350,” "iSC480," or “iMP350”
17	1	Constant	ASCII control character – FS
18	Variable	Alphanum	iConnectEFT Constant = P07_RES_UNIT_SERIAL_NUMBER Unit Serial Number. During Alert Irruption condition, "ALERT" is sent in place of the manufacture serial number. Otherwise, it will send the serial number as normal.
M	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
M + 1	Variable	Decimal	iConnectEFT Constant = P07_RES_RAM_SIZE RAM size in KB.
N	1	Constant	ASCII control character – FS
N + 1	Variable	Decimal	iConnectEFT Constant = P07_RES_FLASH_SIZE Flash memory size in KB.
O	1	Constant	ASCII control character – FS
O + 1	4	Decimal	iConnectEFT Constant = P07_RES_DIGITIZER_VERSION Digitizer Version .
O + 5	1	Constant	ASCII control character – FS
O + 6	4	Decimal	iConnectEFT Constant = P07_RES_SECURITY_MODULE_VERSION Security Module Version.
O + 10	1	Constant	ASCII control character – FS
O + 11	Variable	Decimal	iConnectEFT Constant = P07_RES_OS_VERSION OS Version.
P	1	Constant	ASCII control character – FS
P + 1	4	Decimal	iConnectEFT Constant = P07_RES_APPLICATION_VERSION Application Version, data format is XXYY.
P + 5	1	Constant	ASCII control character – FS
P + 6	4	Decimal	iConnectEFT Constant = P07_RES_EFTL_VERSION EFTL Version.
P + 10	1	Constant	ASCII control character – FS
P + 11	4	Decimal	iConnectEFT Constant = P07_RES_EFTP_VERSION EFTP Version.

Offset	Length	Type	Description
P + 15	1	Constant	ASCII control character – FS
P + 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.
Q	1	Constant	ASCII control character – FS
Q + 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.
R	1	Constant	ASCII control character – FS
R + 1	Variable	Alphanum	iConnectEFT Constant = P07_RES_MANUFACTURING_SERIAL_NUMBER Manufacturing Serial Number
S	1	Constant	ASCII control character – FS
S + 1	4	Alphanum	iConnectEFT Constant = P07_RES_EMV_DC_KERNEL_TYPE Kernel type.
S + 5	1	Constant	ASCII control character – FS
S + 6	4	Alphanum	iConnectEFT Constant = P07_RES_EMV_ENGINE_KERNEL_TYPE Engine kernel type.
S + 10	1	Constant	ASCII control character – FS
S + 11	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_DISCOVER_KERNEL_TYPE Contactless Discover kernel type.
S + 15	1	Constant	ASCII control character – FS

Offset	Length	Type	Description
S + 16	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_EXPRESSPAY_V3_KERNEL_TYPE Contactless ExpressPay v3 Kernel type.
S + 20	1	Constant	ASCII control character – FS
S + 21	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_EXPRESSPAY_V2_KERNEL_TYPE Contactless ExpressPay v2 Kernel type.
S + 25	1	Constant	ASCII control character – FS
S + 26	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_PAYPASS_V3_KERNEL_TYPE Contactless PayPass v3 kernel type.
S + 30	1	Constant	ASCII control character – FS
S + 31	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_PAYPASS_V3_APP_TYPE Contactless PayPass v3 application type.
S + 35	1	Constant	ASCII control character – FS
S + 36	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_VISA_PAYWAVE_KERNEL_TYPE Contactless VISA PayWave kernel type.
S + 40	1	Constant	ASCII control character – FS
S + 41	4	Alphanum	iConnectEFT Constant = P07_RES_CLESS_INTERAC_KERNEL_TYPE Contactless Interac kernel type.
S + 45	1	Constant	ASCII control character – FS
S + 46	Variable	Alphanum	iConnectEFT Constant = P07_RES_CLESS_INTERFACE_IS_SUPPORTED • "YES" = Contactless interface supported for transactions. • "NONE" = Contactless interface not supported for transactions.
T	1	Constant	ASCII control character – ETX

Offset	Length	Type	Description
T + 1	1	Binary	LRC check character.

RBA's default configuration returns the full size of RAM and Flash memory in kilobytes. If compatibility with previous versions (e.g., older than version 3.0.0), which limited the RAM and Flash memory fields to four digits (max 9999 kilobytes) is desired, set '0013_0016' (Field Size for RAM and Flash Memory Size) to the value '0'. See also Compatibility Flags (compat.dat).

4_2_7 08.x: Health Stat

The Health Stat Message flows from the POS to the terminal. It is used to request health and usage statistics from the terminal, as well as identification information for the terminal. Refer to the below table for a description of the 08.x request message format.

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".

> **Info**
>
> 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.”
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)

Offset	Length	Type	Description
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)
M + 1	Variable	Alphanum	iConnectEFT Constant = P08_RES_SERIAL_NUMBER Unit serial number. During Alert Irruption condition, "ALERT" is sent in place of the manufacture serial number. Otherwise, it will send the serial number as normal.
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)

Offset	Length	Type	Description
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
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.
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

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)

Offset	Length	Type	Description
104	4	Decimal	Average backlight enable time.
108	1	Constant	ASCII control character – ETX
109	1	Binary	LRC check character

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

4_2_8 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 to provide card insertion status (e.g., inserted, removed).

Anytime a WIC, EMV, or other type of smart card is inserted in the terminal to initiate a transaction, the terminal sends the appropriate message indicating that the card has been inserted. The card reader must be enabled, or the message will not be sent. This mitigates battery consumption while not in use. Additionally, the 09.x message will be sent if a 23.x message is sent to the terminal.

In the event a damaged or invalid card is inserted, the terminal will send a '09.990200P' message to inform the POS that his has occurred. An invalid card type may include other smart cards if disabled (e.g., disabled EMV card; '0019_0001' = '0'). Refer to the following table which 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.	09.990200P

In order to avoid interrupting the EMV/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. The value of parameter '0007_0047' in Main Flow (mainFlow.dat) determines when and whether the 09.x message is sent.

Refer to the following table which describes the 09.x Card Status message format.

09.x Card Status 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	2	Numeric	iConnectEFT Constant = P09_RES_CARD_TYPE Card type: • 00 = Generic smart card (SMC). • 01 = WIC smart card. • 02 = EMV smart card. • 99 = Damaged or invalid card. The 09.x Card Status message only returns status for smart cards.
6	2	Numeric	Message version/variant = P09_RES_MSG_VERSION • 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. Future variants may be implemented for all card types.
8	2	Numeric	Transaction type = P09_RES_TRANSACTION_TYPE • 00 = Unknown (typically for invalid cards). • 01 = Purchase. Future transaction types maybe implemented (e.g., refund, debit).

Offset	Length	Type	Description
10	1	Alphanumeric	Card status = P09_RES_CARD_STATUS • I = SMC inserted. • R = SMC removed. • P = Unknown problem. Future status options may be added, especially for other card types.
11	1	Constant	ASCII control character – ETX
12	1	Binary	LRC check character

4_2_9 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

4_2_10 10.x: Hard Reset Message

The 10.x Hard Reset message is architected to flow in either direction. It 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. Its purpose is to clear out 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 out any previous actions and returns to the ready state. The customer may choose another form of payment or redo the payment from the start. 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 complete.

The RBA decides 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 = don't clear, 1 = clear).

The 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 if the line display should be cleared or not, 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 the 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 out by the RBA, the customer has either cancelled the transaction or declined the total. In either case, a new amount message must be sent to the terminal.

If the customer pushes the [Cancel] 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

4_2_11 11.x: Status Message

4_2_11_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.

4_2_11_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."
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. • 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. • 99 = Idle. ⚠ 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.

4_2_12 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.

4_2_13 13.x: Amount Message

The 13.x Amount Message flows from the POS to the terminal. It is used to provide the terminal with the current Balance Due amount, and to request an Authorization Request message from the terminal.

The Amount message has 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 value is unconditionally accepted.

When a 13.x message with multiple fields is received, only one of the fields is used. The Amount Index value from the Cards section of `config.dfs` (see Card Configuration Table) selects the correct amount field to use with the authorization message.

- If the Amount Index value is '0', or if it points a field that does not exist, the RBA 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: Set Payment Type Request, forced payment with amount (single amount field)
- 28.x: Set Variable Request, set amount (single amount field)

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 (3 to 9 characters). ASCII string representing the BAL DUE in cents (e.g., $123.89 BAL DUE is represented by 12389) (optional for alternate amount).
P	1	Constant	ASCII control character – ETX
P + 1	1	Binary	LRC check character.

4_2_14 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.

4_2_15 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. • 7 = Reset PIN. • 8 = Clear Line Item Display.
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character.

4_2_16 16.x: Contactless Mode Request

4_2_16_1 Overview of the 16.x Contactless Mode Request Message

The 16.x Contactless Mode Request message is used by the terminal to send 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, Google Wallet, and Softcard SmartTap features. A message subtype included in the 16.x request message determines the message usage (e.g., Contactless Key Card, Google Wallet, Softcard SmartTap). There are two message format deltas from the standard 16.x request message format which support Contactless Key Card. These deltas which support Google Wallet and Softcard SmartTap are described in the tables provided for the 16.x message format.

4_2_16_2 16.x Contactless Mode Request Message Format

The following tables describes the 16.x Contactless Mode Request message content for Contactless Key Card, Google Wallet, and Softcard SmartTap.

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' indicating Contactless key Card mode. • 1 = Google Wallet. • 4 = Softcard SmartTap • 5 = Contactless Key Card 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 will always be "0".
7	1	Alphanum	Card index number. • The first card detected will always have a value of "0".

Offset	Length	Type	Description
8	1	Alphanum	Card type. • 0 = Unsupported or unknown card type. • 1 = MIFARE Classic 1K. • 2 = MIFARE Ultralight. • 3 = MIFARE Mini. • 4 = MIFARE Classic 4K.
9	Variable	Hex-ASCII	UID of the card. • MIFARE Classic 1K (format is 4-byte NUID, 8 hex-ASCII digits). • MIFARE Ultralight (format is 7-byte NUID, 14 hex-ASCII digits). • MIFARE Mini (format is TBD). • MIFARE Classic 4K (format is TBD).
M		Constant	ASCII control character – ETX
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.

16.1: Contactless Mode Request Format for Google Wallet

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “16.”
4	1	Alphanum	Message subtype = '1' indicating Google Wallet. • 1 = Google Wallet mode. • 4 = Softcard SmartTap • 5 = Contactless Key Card 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 Google Wallet data is available. • 1 or greater = Additional Google Wallet data is available.
7	Variable	Alphanum	ASCII representation of Google Wallet hex data.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

16.4: Contactless Mode Request Format for Softcard SmartTap

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “16.”
4	1	Alphanum	Message subtype = '4' indicating Softcard SmartTap. • 1 = Google Wallet mode. • 4 = Softcard SmartTap • 5 = Contactless Key Card 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 Softcard SmartTap data is available. • 1 or greater = Additional Softcard SmartTap data is available.
7	Variable	Alphanum	ASCII representation of Softcard SmartTap hex data.
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

4_2_16_3 Google Wallet Mode Message

A Google Wallet Mode message flows from the POS to the terminal. it is used to set the Google Wallet mode for reading track data during Google Wallet transactions. The following table describes this message format.

16.x Google Wallet Message Format

Offset	Length	Format	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – "16."
4	1	Decimal	Google Wallet mode. • 2 = PPSE first. • 3 = MIFARE first.
5	1	Constant	ASCII control character – ETX
6	1	Binary	LRC check character

4_2_16_4 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	←	

4_2_16_5 Google Wallet Data Message

Once a mobile phone with the installed Google Wallet application enters the contactless field and connects with the terminal, the terminal will read the information (e.g., gift card, Loyalty card). A 16.x data message is sent to the POS which will include status and data availability. If the read operation was successful, the message will indicate successful data retrieval and available data. The following example message illustrates a successful read operation.

Example 1:

'16.101<ASCII representation of Google Wallet hex data>'

If the read indicates an unsupported type or data length exceeding the communication buffer length, this information will be provided in the 16.x data message. Refer to the following example messages.

Example 2:

Read indicates an unsupported type.

'16.120'

Example 3:

Read indicates that the data length exceeds the communication buffer length.

'16.151'

> Telium RBA messaging is ASCII-based. Accordingly, any binary data must be sent as hex-ASCII.

> For details about this message, including a list of supported contactless modes, please refer to one of the following Ingenico documents: **DIV350980 Rev A Isis SmartTap Implementation Guide**; **DIV350955 Rev A Google Wallet Implementation Guide**.

> 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.

4_2_17 17.x: Merchant Data Write

4_2_17_1 Overview of the 17.x Merchant Data Write Message

The 17.x Merchant Data Write Message is used to support the following RBA features:

- Contactless Key Card
- Google Wallet
- Softcard SmartTap

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. This message can also be used to inject Merchant ID/Secret Pairs for Google Wallet or to configure the terminal for Softcard SmartTap. A message subtype included in the 17.x request message determines the message usage (e.g., Contactless Key Card, Google Wallet, Softcard SmartTap). There are two message format deltas from the standard 17.x request and 17.x response formats used to support Contactless Key Card. These deltas which support Google Wallet and Softcard SmartTap are described in the tables provided for the 17.x Merchant Data Write Request Message Format and 17.x Merchant Data Write Response Message Format.

Refer to 17.x Merchant Data Write Message Usage Examples for usage examples with the 17.x Merchant Data Write Message.

4_2_17_2 Contactless Key Card

The POS sends Contactless Key Card commands to the terminal using the 17.x Merchant Data write Request message. This message includes the command type (e.g., 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 (e.g., successful, failed), confirms the command type and availability of data, and provides the actual data which was requested (e.g., Authenticator Sector data).

4_2_17_3 Google Wallet Merchant ID/Secret Pair Injection

When the 17.x Merchant Data Write Request message is received with a message subtype of '1,' the '17.1' message follows a format specific to Google Wallet. The POS uses the Merchant ID/Secret Pair Injection Request message to inject the terminal with Merchant ID /secret pairs used to process Google Wallet transactions. A Merchant ID/Secret Pair Injection Request message must be sent each time the terminal is turned on or rebooted in order for the terminal to support Google Wallet transactions.

All Merchant ID/secret pairs to be injected must be included in the same 17.x message using field separator [FS] characters. Once a '17.1' Merchant ID/Secret Pair Injection Request message is received, all previously injected Merchant ID/secret pairs are replaced with those included in the new message. A '17.1' Merchant ID/Secret Pair Injection Response message is sent from the terminal to the POS indicating injection status (e.g., successful, failed). A '17.1' Merchant ID/Secret Pair Injection Request message received with no Merchant ID/secret pairs will prompt the terminal to clear all injected Merchant ID/secret pairs.

4_2_17_4 Softcard SmartTap

The POS uses the 17.x: Merchant Data Write Request message to configure the terminal to support the Softcard SmartTap feature. When this message is received with a message subtype of '4', the RBA will configure the terminal based on the configuration data included in the request message. When configuration is completed, the terminal sends a 17.x Merchant Data Response message to the POS indicating success status.

4_2_17_5 17.x Merchant Data Write Request Message Format

Refer to the following tables which describes the 17.x Merchant Data Write Request message format for Contactless Key Card, Google Wallet, and Softcard SmartTap, respectively.

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. • 1 = Google Wallet. • 4 = SoftCard SmartTap. • 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, read 4 consecutive 4-byte pages.
			• W = Write.
			  ○ For MIFARE Classic 1K, write 16-byte block to authenticate sector.
			  ○ For MIFARE Ultralight, write 4-byte pages.
			• I = Increment (future use).
			• D = Decrement (future use).
			• M = Move (future use).
			• 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	2	Hex ASCII	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.
8	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.
9	1	Alphanum	Command ID that is used in conjunction with the 36.x notification message. • If not sending a 36.x notification message for this command, then the value of the ID is ignored and can assume any 1-byte (2 hex-ASCII digits) value. • If sending a 36.x notification message for this command, the value of the ID should be selected such that the individual command can be correctly identified by the POS when sent in the 36.x message.
10	Variable	Hex-ASCII	Command data.

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>

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>1 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>

Read.

For MIFARE Classic 1K:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>1 byte (2 hex-ASCII digits)</td><td>Sector index.</td></tr>
<tr><td>2</td><td>1 byte (2 hex-ASCII digits)</td><td>Block index.</td></tr>
</table>

For MIFARE Ultralight:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>1 byte (2 hex-ASCII digits)</td><td>Page address of the first of four consecutive 4-byte pages to be read.</td></tr>
</table>

Write.

For MIFARE Classic 1K:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>1 byte (2 hex-ASCII digits)</td><td>Sector index.</td></tr>
<tr><td>2</td><td>1 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>

For MIFARE Ultralight:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>1 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>

</td></tr>
</table>

Offset	Length	Type	Description
			• Complete Tap (omits this field).
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character

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.

17.1 Merchant Data Write Request Message Format for Google Wallet

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “17.”
4	1	Decimal	Message subtype. • 1 = Google Wallet. • 4 = SoftCard SmartTap. • 5 = Contactless Key Card mode. All other values are reserved.
5	4	Constant	Merchant ID. • '0000' is required for Google Wallet features to function correctly. • 'FFFE' prompts the terminal to create logs for Google Wallet transactions.
9	48	Alphanum	Secret (24 bytes in length)
57	1	Constant	ASCII control character - FS
58	4	Alphanum	First merchant ID
62	48	Alphanum	Secret (24 bytes in length)
110	1	Constant	ASCII control character - FS This field separator character is only required if an additional MID/secret pair is expected to follow.
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character

17.4 Merchant Data Write Request Message Format for Softcard SmartTap

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “17.”
4	1	Decimal	Message subtype. • 1 = Google Wallet. • 4 = SoftCard SmartTap. • 5 = Contactless Key Card mode. All other values are reserved.
5	Variable	Alphanum	Hex representation of the configuration data required to call "Get SmartTap."
M	1	Constant	ASCII control character - ETX
M + 1	1	Binary	LRC check character.

4_2_17_6 17.x Merchant Data Write Response Message Format

Refer to the following tables which describes the 17.x Merchant Data Write Request message format for Contactless Key Card, Google Wallet, and Softcard SmartTap, respectively.

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. • 1 = Google Wallet. • 4 = SoftCard SmartTap. • 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	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 will always be "0".

Offset	Length	Type	Description

7	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, read 4 consecutive 4-byte pages.
			• W = Write.
			  ◦ For MIFARE Classic 1K, write 16-byte block to authenticate sector.
			  ◦ For MIFARE Ultralight, write 4-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 4-byte page and store the new value in a destination 4-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 4-byte page and store the new value in a destination 4-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 4-byte page into a destination 4-byte page.
			• C = Complete Tap.

8	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.

<table>
<tr><th>Offset</th><th>Length</th><th>Type</th><th>Description</th></tr>
<tr><td>9</td><td>2</td><td>Hex ASCII</td><td>Command ID that is used in conjunction with the 36.x notification message.
If not sending a 36.x notification message for this command, then the value of the ID is ignored and can assume any 1-byte (2 hex-ASCII digits) value.
If sending a 36.x notification message for this command, the value of the ID should be selected such that the individual command can be correctly identified by the POS when sent in the 36.x message.</td></tr>
<tr><td>11</td><td>Variable</td><td>Hex-ASCII</td><td>Command data.
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>1 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>
Read.
For MIFARE Classic 1K:<table><tr><th>Sequence</th><th>Format</th><th>Description</th></tr><tr><td>1</td><td>1 byte (2 hex-ASCII digits)</td><td>Sector index.</td></tr><tr><td>2</td><td>1 byte (2 hex-ASCII digits)</td><td>Block index.</td></tr></table>
For MIFARE Ultralight:<table><tr><th>Sequence</th><th>Format</th><th>Description</th></tr><tr><td>1</td><td>1 byte (2 hex-ASCII digits)</td><td>Page address of the first of four consecutive 4-byte pages to be read.</td></tr></table></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>

Write.

For MIFARE Classic 1K:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>1 byte (2 hex-ASCII digits)</td><td>Sector index.</td></tr>
<tr><td>2</td><td>1 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>

For MIFARE Ultralight:
<table>
<tr><th>Sequence</th><th>Format</th><th>Description</th></tr>
<tr><td>1</td><td>1 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>

Complete Tap (omits this field).

</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>

17.1 Merchant Data Write Response Message Format for Google Wallet

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “17.”
4	1	Decimal	Message subtype. • 1 = Google Wallet. • 4 = SoftCard SmartTap. • 5 = Contactless Key Card mode. All other values are reserved.
5	4	Constant	Status: • 0 = Successful injection. • 1 = Failed injection. • 2 = Unsupported type. • 4 = Invalid format. • 5 = Data to be returned exceeds the communication buffer.
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character

17.4 Merchant Data Write Response Message Format for Softcard SmartTap

Offset	Length	Format	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “17.”
4	1	Decimal	Message subtype. • 1 = Google Wallet. • 4 = SoftCard SmartTap. • 5 = Contactless Key Card mode. All other values are reserved.
5	4	Constant	Status: • 0 = Successful injection. • 1 = Failed injection. • 2 = Unsupported type. • 4 = Invalid format. • 5 = Data to be returned exceeds the communication buffer.
N	1	Constant	ASCII control character - ETX
N + 1	1	Binary	LRC check character

4_2_17_7 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.

For details about this message, please refer to one of the following Ingenico documents: **DIV350980 Rev A Isis SmartTap Implementation Guide**; **DIV350955 Rev A Google Wallet Implementation Guide**.

4_2_17_8 17.x Merchant Data Write Message Usage Examples

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	17.500R000002030001020304050607080 90A0B0C0D0E0F	←	

- 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	17.500000400010203040506070 8090A0B0C0D0E0F	←	

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		

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]R000002010001020304050607080 90A0B0C0D 0E0F [FS]W000002010F0E0D0C0B0A090807060504030201 00[FS]R000002010F0 E0D0C0B0A090807060504030201 00[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] R000004030201000405060708090A0B0C0D0E0F[FS]C0000		

4_2_18 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.
N	1	Constant	ASCII control character – ETX
N + 1	1	Binary	LRC check character.

4_2_19 19.x: BIN Lookup

The 19.x BIN Lookup message is sent to the POS to allow for external BIN range lookup for purposes of pre-selecting the payment type for the customer. It will automatically be disabled if the unit is set to "Select Payment Type" first.

If a customer swipes a payment card, this message is sent to the POS. If the POS responds within the set timeout range, the application will either:

- automatically select the payment type for the customer,
  Or:
- if "UNKNOWN" was returned, allow the customer to proceed and select the payment type.

For automatic payment selection to occur, the following must occur:

- 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 from the request

If there is any mismatch, the response is ignored and the terminal continues waiting using the original timeout value.

When the payment type is automatically chosen for the customer, the **Select Payment** 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), there is a CANCEL button, which will take the transaction back to the **Select Payment** screen without clearing the swiped card data from the application. This allows the customer to change the payment type to any other payment type, at which point the application proceeds as a normal Slide Card first transaction without Spin 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 set timeout range, the application proceeds to allow the customer to manually select the payment type and ignore any further payment type messages.

An account data origin option of "A" has been added to enable the application to distinguish between manual entry and an account message.

19.x PIN Encouragement Request Message Format (Received from RBA)

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_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 = Account Message (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’.

Offset	Length	Type	Description
5	1	Alphanum	iConnectEFT Constant = P19_REQ_TRACK1_DATA_STATUS Track 1 good read indicator. • 0 = Bad read. • 1 = Good read. If account number is from 12.x account message, set to ‘0’.
6	1	Alphanum	iConnectEFT Constant = P19_REQ_TRACK2_DATA_STATUS Track 2 good read indicator. • 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_RESERVED Reserved (use ‘0’).
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)

Offset	Length	Type	Description
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. A 0007341111597242000
Contactless	19. d 11000114005578000000150
MSR / Card Swipe	19. D 11000106011000990911111
Manual Entry	19. T 00000069999999800009901

19.x PIN Encouragement Response Message Format

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_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.

4_2_20 20.x: Signature Message (On-Demand)

4_2_20_1 Overview of the 20.x Signature Messages (On-Demand)

Signature capture is initiated by the POS using the 20.x: Signature Request Message (On-Demand) message . When this message is received, the terminal will display the requested form and prompt. The cardholder may then begin signing. The 20.x: Signature Response Message (On-Demand) notifies the POS when a keypad key or screen button is pressed during the signature process.

The response message provides a status and may be used to indicate an invalid prompt or state save error (when the '0009_0006' Save State on Signature Capture Request parameter is set to '1'). A status value of '2' in the response message indicates that a button was pressed (e.g., <Clear>, <Yes>, <No>). When this occurs, a 2-byte Data Input field will follow indicating which button was pressed. A status value of '0' followed by a 1-byte Data Input field of '2' indicates that the signature was accepted.

If the signature notification message is enabled (parameter '0009_0002' = '1'), then the terminal will use the Signature Ready Response Message (On-Demand) to notify the POS that the signature has been completed and is ready for downloading. Otherwise, the POS must poll to determine when the signature is available.

The 20.x Signature Request Message may be used at the following times:

- If the payment type requires a signature but the payment is not configured to automatically prompt for a signature, the POS may send the 20.x Signature Request Message.
- The cashier compares the signature with the signature on the card, and would like the customer to resign.
- To collect a signature for something other than the payment transaction.

When any of the following messages are received during the execution of the on-demand signature capture, the message will not be executed. A response message including reject status will be returned while 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.

4_2_20_2 20.x Signature Message Format

The following tables describes the 20.x Signature Request and 20.x Signature Response message formats. Refer to the Signature Ready Response Message (On-Demand) section for a description of the 20.x Signature Ready Response message.

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 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
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 = <CANCEL> key pressed. • 2 = Button pressed. Refer to Data input field for key pressed. • 6 – Invalid prompt. • 9 – State save error (only returned during errors when '0009_0006' = 1).
5	1 or 2	Decimal	iConnectEFT Constant = P20_RES_KEY Data input field (if Status = '2'), 2 bytes. • 08 = <CLEAR> key (Backspace). • 89 = <Yes> key. • 78 = <No> key. -or- Optional signature status field (if Status = '0'), 1 byte. • 2 = Signature accepted
M	1	Constant	ASCII control character – ETX
M + 1	1	Binary	LRC check character.

4_2_20_3 Usage Examples

20.x Signature Request Message

Example 1 : 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 ENTER 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. When signing is completed, 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

' 20.0 '

indicating that the <ENTER> key was pressed.

Example 2: 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 ENTER with pen"). The format for this message will be

' 20.165[FS]SIGN.K3Z '

20.x Signature Response Message

Example 3:

The cardholder begins signing, but presses the "Clear" button on the display. In this example, the format of the 20.x Signature Response message will be

' 20.208 '

indicating that a screen button was pressed. The second field in the message ('08') indicates that it was the "Clear" button which was pressed.

Example 4:

An invalid prompt was included in the 20.x Signature Request message. The terminal returns the following 20.x Signature Response message:

' 20.6 '

A signature accepted message would have the following format:

' 20.02 '

20.x Signature Ready Response Message

Example 6:

The 20.x Signature Ready Response message has been enabled. The signature is completed and the terminal sends the following notification message to the POS:

' 20. '

4_2_20_4 A Note About Keys and Buttons Pressed During the Signature Process

The "Cancel" and "Clear" keys function as follows during the signature capture process:

- The keypad "Cancel" key is by default disabled during on-demand and normal credit transaction flows.
- The keypad "Clear" key will clear the Signature form in preparation for the new signature.

Prior to signature capture, the Signature form should be refreshed. Once the signature capture has been captured, the terminal will display the approval status as follows:

- For on-demand, the terminal will display "Signature Approved".
- For normal credit flow, the terminal will display "Approved".

4_2_20_5 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.

4_2_21 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, driver’s 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 will restart. Any portion of the PIN that was entered before the interruption will be lost. A '15.6' Stop Action soft reset will cancel the input request and continue 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 out.
- 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 out, the display shows the “Input Cancelled” prompt for three seconds, the current process terminates, and the 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. When this happens, the “Input Cancelled” prompt displays, and a '21.1' response ('1' indicating the cancel state) is sent out. The function terminates and returns to the interrupted state.

4_2_21_1 Use of the 21.x Message to Send Encrypted Clear Entry Data

With some customers prompting the cardholder for sensitive information such as their social security number, a new method for encrypting this clear entry data and sending it to the POS has been established. This information can now be 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. In order to facilitate this, configuration parameters '0091_0019' through '0091_0022' have been reallocated to support generic message encryption. Refer to Security Parameters (security.dat) for more detail on the use of these parameters. These parameters now support encryption of barcode data as well as encryption of clear entry data such as the cardholder's sensitive information. The encryption key used in both cases is the base64 barcode key. The following new configurations have been added to enable encryption for specific messages:

- '0091_0026' - Used to enable encryption of clear entry data via 21.x and 27.x messages.
- '0091_0027' - Used to 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.
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 form's 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.”
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.

4_2_22 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
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.

4_2_23 23.x: Card Read Request (On-Demand)

4_2_23_1 Overview of the 23.x Card Read Request Message

The 23.x Card Read Request message flows from the POS to the terminal. The 23.x message stops the execution of the current process, displays the Card Swipe process screen, and waits for a card swipe on the terminal magnetic stripe reader (MSR). When the MSR track data is available, it sends out a 23.x response with entered data, and returns to the process it interrupted.

The 23.x request is always ACKed. Rules for the message are:

- 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.

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 the processing of 20.x: Signature Message (On-Demand) is dependent upon 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).

After a card swipe is executed, the following can happen:

- When the MSR card read is correct, the “Card Accepted” prompt is displayed for three seconds, and the terminal sends out a 23.x response with status of '0' and attached track data. The RBA returns to the interrupted process.
- When the MSR card read is not correct, the “Card Read Cancel” prompt is displayed for three seconds, the terminal sends out a 23. x response, and RBA returns to the interrupted process.

After a contactless card touch/tap is executed, the following can happen:

- When the MSR card read is correct, the “Card Accepted” prompt is displayed for three seconds, and the terminal sends out a 23.x response with status of '0' and attached track data. The RBA returns to the interrupted process.
- When the MSR card read is not correct, the “Card Read Cancel” prompt is displayed for three seconds, the terminal sends out a 23. x response, and the RBA returns to the interrupted process.

4_2_23_2 23.x Message Flow When Using iConnectEFT

The 23.x message has different flows depending on the '0013_0014' configuration parameter setting. This setting controls whether to include or exclude the source field in the 23.x Card Read Response message. The message flow is changed based upon this configuration parameter setting. If set to exclude the source field from the 23.x response message, the iConnectEFT sends a 61.x: Configuration Read message following every 23.x message if the configuration setting of '0013_0014' is disabled.

4_2_23_3 New Flag for Coupon Support

A new flag has been added to the 23.x request message to indicate that couponing data (if enabled) should be returned in the 23.x response message, and not in the 16.x Contactless Mode Request message. An optional 1-byte 'Options' field has been added to the 23.x message to facilitate this feature. All optional fields before the 'Options' field must be present in order to use this field. When set to a value of '1' this indicates that the RBA will send a 16.x response message with a status of '6' if payment data is available and a pending a 23.x response message. Refer to the following examples.

Example 1:

In this example, the Options field is set to '1' indicating that the RBA 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]MSC[ETX][LRC]

Example 2:

In this example, the following message uses the prompt "Swipe, Tap or Insert Card” with form FORM1.K3Z having readers MSR, Cless and SCR 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 default 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, Tap or Insert Card” with the default form selection, having MSR and Cless readers enabled, without the 16.x response notification.

[STX]23.Swipe, Tap or Insert Card[FS][FS][FS]MC[ETX][LRC]

4_2_23_4 MSC Field

The MSC field is used to indicate which readers to enable (MSR, contactless and SCR). If contactless mode was not enabled previously, "card swipe on demand w/o Cless" will be set as the default form. The contactless reader will not be enabled if the appropriate contactless mode has not been enabled, regardless of the MSC field content. Refer to the following examples of MSC field use in the 23.x message.

1. 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]
2. Use prompt "Swipe or Tap Card" with default form, enabling MSR and contactless:
   [STX]23.Swipe or Tap Card[FS][FS]MC[ETX][LRC]

4_2_23_5 Swiping an Invalid Card or Cancelling Manual Entry

When swiping an invalid card, a "Bad Read" exit type is returned in the 23.x response message. The "Source of Card Read" field will be blank, and a "?" character will be inserted in the field designated for Track 1 data. The message format for an invalid card swipe will then be

' 23.x?[FS] '

When selecting manual entry and then cancelling, a "Cancelled" exit type is returned in the 23.x response message. The "Source of Card Read" field will be blank, and a "?" character will be inserted in the field designated for Track 1 data. The message format for cancelled manual card entry will then be

' 23.2?[FS] '

In both cases, the "Source of Card Read" field is blank and a "?" character is inserted Track 1 data field. The "?" character defines the card source as unknown or invalid.

4_2_23_6 Execution of the 23.x Message

The execution of the 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)
- Or by tapping the CANCEL button. When this occurs, the “Card Read Cancelled” prompt is displayed, and a '23.2' cancel response message is sent out. The function terminates and returns to the interrupted process.

> **Info**
>
> Encryption and BIN range checks on a card account number will be performed based on the settings from the configuration file.

4_2_23_7 23.x Card Read Request (On-Demand) Message Format

The following tables describe the format for the 23.x Card Read Request and 23.x Card Read Response messages.

23.x Card Read Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M23_CARD_READ_REQUEST_ON_DEMAND Message Identifier – ASCII – “23.”
4	Variable	Alphanum	iConnectEFT Constant = P23_REQ_PROMPT_INDEX Prompt index number. Can be literal text up to 230 characters, an index number from Prompt.xml, or nothing when used with an optional form_file_name.
M	1	Constant	ASCII control character – FS (This field is optional.) Only used with M + 1.
M + 1	Variable	Alphanum	iConnectEFT Constant = P23_REQ_FORM_NAME Form File Name or Number (this field is optional). Only used with M. Maximum length = 13 characters.
N	1	Constant	ASCII control character – FS (This field is optional.)
N + 1	1	Alphanum	iConnectEFT Constant = P23_REQ_OPTIONS • 1 = RBA will send 16.x response with status '6' if payment data is available and pending a 23.x response.
N + 2	1	Constant	ASCII control character – FS (This field is optional.)
N + 3	Variable	Alphanum	iConnectEFT 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. • S = Enable SCR. The order of the letters does not matter, only whether or not they are included in the string. To avoid ambiguity, the Form field must be present (even with empty value) in order for the MSC field to be sent.
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character

23.x Card Read Response Message Format (if '0013_0014' = 1)

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	iConnectEFT Constant = M23_CARD_READ_REQUEST_ON_DEMAND Message Identifier – ASCII – “23.”
4	1	Alphanum	iConnectEFT Constant = P23_RES_EXIT_TYPE Exit Type: • 0 = Good Read • 1 = Bad Read • 2 = Cancelled • 3 = Button Pressed • 6 = Invalid Prompt • 7 = Encryption Failed • 9 = Declined

5	1	Alpha	iConnectEFT Constant = P23_RES_CARD_SOURCE
			Source of card read:
			• C = Contactless Reader.
			• H = Manual entry.
			• M = MSR.
			• S = One of the following:
			  ◦ SLE5542 memory card
			  ◦ EMV card
			  ◦ WIC card
			Only included in Response if '0013_0014' parameter = 1.

6	Variable	Decimal	iConnectEFT Constant = P23_RES_TRACK1 Track 1 data. (If the above field is 3 = Button Pressed, then it will be the single byte ID of the key.)
M	1	Constant	ASCII control character – FS
M + 1	Variable	Decimal	iConnectEFT Constant = P23_RES_TRACK2 Track 2 data

Offset	Length	Type	Description
N	1	Constant	ASCII control character – FS (only if Track 3 data is present)
N + 1	Variable	Alphanum	iConnectEFT Constant = P23_RES_TRACK3 Track 3 data
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character

4_2_23_8 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^10121015555501234000000000055610057174000[FS] 4005578000000150= 10121015555557100741
Contactless	23.0 C B4005578000000150^CARDHOLDER/VISA^10121015555501234000000000055610046077000[FS] 4005578000000150= 10121015555546000771
Manual Entry '0013_0014' = '0' '0007_0029' = '1'	23.0 H 5444009999222205^MANUAL ENTRY/^14120000000123000000[FS]5444009999222205= 14120000000123000
MSR / Card Swipe	23.0 M B5444009999222205^TESTVOID/TEST^14121010000000000000071700[FS] 5444009999222205= 14121010000071700

4_2_24 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.

4_2_24_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

4_2_24_2 Setting Button Text

The syntax for setting button text is as follows:

Bid,text

where

- id = the button ID from the K3Z form.
- text = the new button text.

Example: Set the button text for id "btn1" to "text":

Bbtn1,text

4_2_24_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:

To show a button:

Bid, S

To hide a button:

Bid, H

Note the extra space following "Bid".

4_2_24_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.

4_2_24_5 Displaying the Offline Form on iSMPc Terminals

It is important to note that due to Bluetooth pairing on iSMPc terminals, the prior form (e.g., "Approved") will be displayed following the request to display the offline form. In order to correctly display the offline form, the '24.' message must specify T1 formatting and include the offline.K3Z message. The required message to display the "This Lane Closed" prompt on iSMPc terminals 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
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

<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>iConnectEFT Constant =M24_FORM_ENTRY_REQUEST_ON_DEMAND
Message Identifier – ASCII – “24.”</td></tr>
<tr><td>4</td><td>1</td><td>Decimal</td><td>iConnectEFT Constant = P24_RES_EXIT_TYPE
Exit Type:
0 = Successful.
1 = Invalid form.
6 = Invalid prompt.
8 = Timeout occurred.
9 = Declined.</td></tr>
<tr><td>5</td><td>1</td><td>Decimal</td><td>iConnectEFT Constant = P24_RES_KEYID
ID of key pressed to exit the form.</td></tr>
<tr><td>6</td><td>1</td><td>Decimal</td><td>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.</td></tr>
<tr><td>7</td><td>1</td><td>Decimal</td><td><table><tr><td>Button State:
0 = Up.
1 = Down.</td><td>Check Boxes:
0 = Not Checked.
1 = Checked.</td><td>Radio Buttons:
x, where x is a value of 1 to y,
where y is the number of buttons in the group.</td></tr></table>
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."</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>

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.

4_2_24_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.

4_2_24_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.

4_2_24_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.

Icon

> 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='000000' fontsize='20px' fontweight='normal' fontfamily='sans-serif' align='left' background='false' backgroundcolor='FFFFFF' />
```

4_2_25 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. There are also buttons which allow the customer to accept or decline the terms and conditions 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)
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	2	Alphanum	iConnectEFT Constant = P25_RES_KEY_PRESSED ID of key pressed to exit • "@" = Accept. • "B" = Decline.
7	1	Constant	ASCII control character – ETX
8	1	Binary	LRC check character.

4_2_26 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.”
4	Variable	Alphanum	iConnectEFT Constant = P26_REQ_FILE_NAME File name. Name of the script file to execute. Filename 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: 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 C. RBA Script Language.

4_2_27 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 payment transaction and prompts the customer for text entry (for example, the customer’s name). When the customer has entered the data or cancelled entry, the Alpha Input Message is sent to the POS. The payment transaction then resumes the transaction 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 will restart. Any portion of the PIN that was entered before the interruption will be lost. A '15.6' Stop Action soft reset message will cancel 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 out.

- 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 does not execute. A reject response status is returned, and the current on-demand "Enter Generic Number" process continues to operate.
- 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 out. The display will show the “Input Cancelled” prompt for three seconds, the current process terminates, and the 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. When this happens, the “Input Cancelled” prompt is displayed, and a '27.1' response (cancel state) message is sent out. The function terminates and returns 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

4_2_27_1 Use of the 27.x Message to Send Encrypted Clear Entry Data

With some customers prompting the cardholder for sensitive information such as their social security number, a new method for encrypting this clear entry data and sending it to the POS has been established. This information can now be encrypted using the same encryption key used for the 95.x: Barcode Data Messages and sent to the POS via the 27.x Alpha Input Message. In order to facilitate this, configuration parameters '0091_0019' through '0091_0022' have been reallocated to support generic message encryption. Refer to Security Parameters (security.dat) for more detail on the use of these parameters. These parameters now support encryption of barcode data as well as encryption of clear entry data such as the cardholder's sensitive information. The encryption key used in both cases is the base64 barcode key. The following new configurations have been added to enable encryption for specific messages:

- '0091_0026' - Used to enable encryption of clear entry data via 21.x and 27.x messages.
- '0091_0027' - Used to enable encryption of barcode data via the 95.x barcode data message.

Exit types "8" and "E" have been added to the 27.x Alpha Input Response message as described in the below table titled "**Alpha Input (On-Demand) Response Message Format.**"

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 Format specifier (see Format Specifiers on page 255 for more information.) (This field is optional.)
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’s 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.

4_2_28 28.x: Set Variable Request

4_2_28_1 Overview of the Set Variable Message

The Set Variable message is architected to flow in both directions as a request/response pair. It is designed to allow 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 (e.g., successful, 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 forty characters per line, displayed using the standard Ingenico monospace font.

- 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 that are displayed at any given time 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. Only assign variables to the lines that can be viewed in the terminal's scrolling receipt area. Variables that are assigned to lines that are not viewable will be ignored. As an example, if your scrolling receipt area displays five lines, only use variables `111` to `116`. Variables assigned to `117` to `119` will not be 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.
    - H = EMV.

  Payment types I through P are reserved and are customer definable. Refer to Card Configuration (cards.dat) for more information on payment types.

The variables may be combined to create a four-line item display with a total line on the bottom. This feature is accomplished by writing the item information to variable `115`, then writing the total line to variable `104`. By default, the terminal is configured to clear the line display on reset. This ability can be turned off and the line display cleared by sending a '15.8' Soft Reset Message.

4_2_28_2 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).

4_2_28_3 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) financial 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 a value equal to the time difference between Local Time and Greenwich Mean Time. This variable must be entered in seconds. As an example:

- Variable `205` = GMT Offset (in seconds).
    - Offsets east of the Prime Meridian use a positive number of seconds.
    - Offsets west of the Prime Meridian use a negative number of seconds.

Info

When calculating the GMT Offset, remember consideration must be given to any adjustments in time (e.g., 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 prior to any PayPal entry. A best practice is to periodically update the Local Time to prevent clock skew; once per day should be 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.

4_2_28_4 Variables

The following table lists and describes the available variables by number. Variables are always available unless noted otherwise in the "Restrictions" column.

Variable Numbers

Variable Number	Variable Name	Restrictions	Get	Set
1 to 25	User Variable 1 through 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
116	6th line of line display.	See Note 1.		X
117	7th line of line display.	See Note 1.		X

Variable Number	Variable Name	Restrictions	Get	Set
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 (HH MM SS).		X	X
202	Date (MM DD YY).		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 was rebooted.		X	
251	Customized Terminal Name (Ex. "Retail Base").		X	
252	Customized Terminal Version Number (Ex. "3.4.0.1").		X	
253	Original App Name (Ex. “Retail Base”).		X	
254	Original Version Number (Ex. “1.1.0.8”).		X	
255	Terminal Reference ID (“iPP350”, “iSC250”, “iSC350”, etc.).		X	
303	Total amount due. This variable returns the current transaction's 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 Transaction's Maximum Cash Back Amount.		X	

Variable Number	Variable Name	Restrictions	Get	Set
395	Card data encryption status for the 23.x: Card Read Request (On-Demand) message. • 0 = Card data not encrypted. • 1 = Magtek Encryption • 2 = On-Guard Encryption • 3 = EPS (Element Payment Systems) Encryption • 4 = Voltage TEP1 Encryption • 5 = Voltage TEP2 Encryption • 6 = Voltage TEP3 Encryption (not currently supported). • 7 = Monetra CardShield Encryption • 8 = Mercury Encryption • 9 = RSA-OAEP Encryption • 10 = TransArmor Encryption • 11 = Generic TDES DUKPT Encryption • 12 = S1 Encryption Possible reasons for card data not being encrypted include: • P2P encryption not configured. • Card not found on whitelist. • The card is a nonstandard card which should not be encrypted.		X	
396	Card Read Transaction Flow MOD-10 check value.		X	
397	Card Read On-Demand calculated MOD-10 check value.	23.x message only.	X	
398	Card read On-Demand account number.	23.x message only.	X	
399	Card read On-Demand account name.	23.x message only.	X	
400	Card read On-Demand expiration date.	23.x message only.	X	
401	Account number.		X	
402	Account name.		X	

Variable Number	Variable Name	Restrictions	Get	Set
403	Account expiration.		X	
404	Payment type see Card Configuration (cards.dat) for valid values.		X	
405	Payment track (used only for magnetic stripe flow). This variable will be populated following a valid card read. This is the same track data to be included in the 50.x Authorization Request message. This can change when the payment method is selected.		X	
406	Payment MSR track data (used only for magnetic stripe flow). This variable is populated following a card swipe if the card has 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 will remain 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	Variable Name	Restrictions	Get	Set
414	Card data encryption used for last swipe in transaction flow. • 0 = Card data not encrypted. • 1 = Magtek Encryption • 2 = On-Guard Encryption • 3 = EPS (Element Payment Systems) Encryption • 4 = Voltage TEP1 Encryption • 5 = Voltage TEP2 Encryption • 6 = Voltage TEP3 Encryption (not currently supported). • 7 = Monetra CardShield Encryption • 8 = Mercury Encryption • 9 = RSA-OAEP Encryption • 10 = TransArmor Encryption • 11 = Generic TDES DUKPT Encryption • 12 = S1 Encryption Possible reasons for card data not being encrypted include: • P2P encryption not configured. • Card not found on whitelist. • The card is a nonstandard card which should not be encrypted.		X	
440	Voltage rollover date.		X	
450-459	M/S key check value.		X	
470-479	DUKPT key check value. • 0 = No key. • 1 = Key found.		X	

Variable Number	Variable Name	Restrictions	Get	Set
500	This is for the 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. This functionality is only supported in production terminals. The timer will not be enabled on mock-up terminals.		X	
600	EMV contact configuration.		X	
601	EMV contactless configuration.		X	
602	Set the suspend list for contactless EMV transactions.			X
603	Set the update list for contactless EMV transactions.			X
604	Set the suspend timer for contactless EMV 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	

Variable Number	Variable Name	Restrictions	Get	Set
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
801	Host 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
810	SMID status (“KSN”, “KCV” keys).		X	

Note 1

Variables exceeding the window height will be ignored. The line display height is defined by the current form.

Note 2

Blocks beyond the number indicated by the `712` variable will not contain data.

4_2_28_5 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.
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

4_2_29 29.x: Get Variable Request

4_2_29_1 Overview of the 29.x: Get Variable Message

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.

4_2_29_2 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.”
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.

4_2_30 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.

4_2_31 31.x: PIN Entry Messages (On-Demand)

4_2_31_1 Overview of the 31.x PIN Entry Message

The 31.x PIN Entry Request message is sued 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.

4_2_31_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.

Ingenico iSC Touch 250 with PIN Entry and Credit Selection Option

4_2_31_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.”

Offset	Length	Type	Description
4	1	Alphanum	iConnectEFT Constant = P31_REQ_SET_KEY_TYPE Key Type selection: • M = Master/Session. • D = DUKPT. • m = E2EE Master/Session. • d = E2EE DUKPT. • P = PayPal. • * = Default.
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. Prompt index number 31 selects the PayPal PIN Entry prompt. Prompt index number 17 selects the "Enter PIN or Press Green for Credit" 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.

Offset	Length	Type	Description
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.”
4	1	Decimal	iConnectEFT Constant = P31_RES_STATUS Status: • 0 = PIN entered. • 1 = Cancelled. • 2 = Invalid Length. • 3 = Invalid Index. • 4 = PIN key pressed (key value is return in the PIN data). • 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.
5	Variable	Alphanum	iConnectEFT Constant = P31_RES_PIN_DATA PIN data (contents depend on value of Status).
M	20	Alphanum	Optional Key Serial Number (KSN) used for DUKPT encryption.
M + 20	1	Constant	ASCII control character - ETX
M + 21	1	Binary	LRC check character.

A new setting is in config.dfs which allows the 31.x message to accept an Enter key without any PIN entry. By default, at least 4 or the value set by '0006_0011' digits must be entered. If '0006_0004' is set to '1' then a valid PIN or and empty PIN will be accepted. This does not change the behavior of PIN entry during normal transaction flow. Refer to PIN Entry (pin.dat) parameter '0006_0004'.

4_2_32 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.”
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.

4_2_33 36.x Notification of Command Execution

4_2_33_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.

4_2_33_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.

4_2_33_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		

4_2_34 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.

4_2_34_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 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

4_2_34_2 40.0 Survey Response

This message is sent from the terminal to the POS when the customer responds to survey question(s) or if there is an error that prevents the survey from displaying.

Survey Request RESPONSE Format

Offset	Length	Type	Description
0	1	Constant	STX – 0x02
1	3	Constant	iConnectEFT Constant = M40_SURVEY Message identifier – “40.”
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 at this time). • 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

4_2_34_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).

4_2_34_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

4_2_34_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. 40.x currently supports only language codes 1-4 (where 1 = English, 2 = Spanish, etc.).
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?`
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`

4_2_35 41.x Card Read Message

The 41.x Card Read Request message provides a way to enable or disable the terminal card readers (MSR, contactless, SMC). If the terminal supports LEDs for the card readers, they will be illuminated as appropriate. Refer to the following tables which 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 will then be disabled and a message will be 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.
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."
4	1	Alphanum	iConnectEFT Constant = RES_SOURCE Source. • M = MSR. • C = Contactless. • S = Smartcard (SMC).
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. • Good read. • Error.

Offset	Length	Type	Description
8	1	Alphanum	iConnectEFT Constant = RES_TRACK_2_STATUS Track 2 status. • Good read. • Error.
9	1	Alphanum	iConnectEFT Constant = RES_TRACK_3_STATUS Track 3 status. • Good read. • Error.
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)

Offset	Length	Type	Description
Q + 6	Variable	Alphanum	iConnectEFT Constant – RES_ACCOUNT_NAME Account Name (only if parsed)
R	1	Constant	ASCII control character – ETX
R + 1	1	Binary	LRC check character

4_2_36 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.

Offset	Length	Type	Description
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.
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 – ETX
P + 1	1	Binary	LRC check character.

Sample Requests

Card Source	Sample Request
Contactless	50.12345678901234567890123456789012345678900207000558300001@d4005578000000150= 10121015555554400751[FS]1@[FS]1025[FS]
MSR / Card Swipe	50.12345678901234567890123456789012345678900207000558300002@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 – “FFFF”			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

4_2_37 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. Refer to the following table which describes the message format.

51.x: Beep On-Demand Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
1	3	Constant	Message Identifier – ASCII – “51.”
4	1	Decimal	Tone selection: • 0 = Low tone (4200Hz). • 1 = Middle tone (6000Hz). • 2 = High tone (9000Hz). • 3 = Error tone (3800Hz).
5	1	Decimal	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.

4_2_38 52.x: PayPal Preauthorization Message

The 52.x PayPal Preauthorization message is sent by the RBA application to the POS once the terminal has collected the information needed to identify the cardholder. This allows the POS to conduct a PayPal Preauthorization and to retrieve any coupons or discounts from the cardholder’s PayPal account. The POS can then apply these discounts to the purchase before sending the total to the RBA 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
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.

4_2_39 58.x Terminal Discovery Message

4_2_39_1 Overview

The RBA 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 has been uploaded.
- The terminal connection is open, not in use.

When the POS connects to a terminal over the TCP, the UDP connection is closed. When the POS closes the TCP connection with the terminal, the UDP connection for the 58.x Terminal Discovery message opens again.

When opened, the UDP port will use the same port number that is used for the TCP connection with the POS.

The '58.0' Terminal Discovery request message flows from the POS to the terminal. The POS uses this message to request the following terminal information:

- Serial number
- Terminal's unique MAC ID number
- Terminal's IP address

When the terminal receives the '58.0' request message, it sends a 58.x response message with this information.

> The Serial Number will be the terminal's injected serial unless unavailable. If unavailable, the hardware serial number will be inserted in the message in its place.

4_2_39_2 Usage Examples

Consider the following use case scenario: The POS is connected to a network with multiple terminals. It would like to communicate with one or more terminals with known serial numbers but does not know their IP addresses. When the POS broadcasts the '58.0' request message, all terminals will return a 58.x response message containing their serial number and IP address. The POS can then extract the IP addresses from these messages and initiate communications with the desired terminals. The format of the '58.0' request message is as follows:

[STX]58.0[ETX][LRC]

The format of the response message will be:

[STX]58. SerialNumber [FS] MacAaddress [FS] IpAddress [ETX][LRC]

where SerialNumber will be the injected serial number if one has been injected on the terminal. If not then this will be the hardware serial number.

As an example, a terminal returns a 58.x message with the following content:

[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 broadcasts 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

4_2_39_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.”
4	Variable	Alphanum	iConnectEFT Constant = P58_RES_SERIAL_NUMBER Injected serial number. If there is no injected serial number then the hardware serial number will be inserted in this field.
M	1	Constant	ASCII control character - FS
M + 1	Variable	Alphanum	iConnectEFT Constant = P58_RES_MAC_ADDRESS Terminal's 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 Terminal's IP address. • Format is "ZZZ.ZZZ.ZZZ.ZZZ"
O	1	Constant	ASCII control character – ETX
O + 1	1	Binary	LRC check character.

4_2_40 60.x: Configuration Write

The 60.x Configuration Write is used to permanently alter the RBA 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 may accept many IDN blocks. The total message length may not exceed the maximum acceptable message length (240 bytes). The RBA returns 60.x response to each 60.x request. The 60.x message is accepted by the RBA 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 will be sent out after writing to the DFS memory is finished and RAM value is updated. Time for the response message may 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.

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

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)
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

1. 4_2_41 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.

After changing all global parameters, power cycle the terminal so that all of the applications installed in the terminal will be updated with the new values.

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”

4_2_42 61.x: Configuration Read

The 61.x Configuration read message is used to retrieve a configuration parameter setting. The 61.x Configuration Read Request message provides the group number and index number of IDN block #1 where the configuration parameter is located. A 61.x Configuration Read Response message is returned with status (e.g., successful, error), group number and index number of IDN block #m, and the data retrieved from that location which is the configuration parameter setting.

Consider the following example:

1. The POS sends the 00.x: Offline Message to the terminal.
2. The terminal. goes offline.
3. The POS sends a '61.7[GS]1' message indicating that the parameter to be read is located at IDN block 7 and index number 1.
4. The terminal responds with a '61.27[GS]1[GS]30' message indicating a successful read operation, confirming the IDN block and index number, and including the parameter setting which is '30'.

Consider another example:

1. The POS sends the 00.x: Offline Message to the terminal.
2. The terminal. goes offline.
3. The POS sends a '61.5[GS]2' message indicating that the parameter to be read is located at IDN block 5 and index number 2.
4. The terminal responds with a '61.25[GS]2[GS]1' message indicating a successful read operation, confirming the IDN block and index number, and including the parameter setting which is '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 parameter setting for '0007_0001' and '0019_0001', which is '30' and '0', respectively.

The following tables describe the 61.x Configuration Read Request and 61.x Configuration Read Response message formats.

61.x Configuration Read Request Message

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
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)
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 Message

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
O + 1			... (optional, in case of multiple parameters)
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 with error code “4”.

4_2_43 62.x: File Write

4_2_43_1 Overview of the 62.x File Write Message

The 62.x File Write message is used to permanently write a file to the terminal's data file system (DFS) memory. It is primarily used to load form files and images (`.bmp`, `.jpg`, `.png`, and `.gif` formats). 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. Creating an EFTL file from the .OGZ will download in considerably less time than the time required for the 62.x File Write with most communications.

> **Info**
>
> Certain Ingenico file types (.PGZ, .TGZ, and .K3Z) require a manual reboot of the terminal after updating. Also refer to the 97.x: Reboot section.
>
> Before the POS sends 62.x message to the terminal, it renames any included file to all caps, e.g., `Test.k3z` will become `TEST.K3Z`.

Included within the 62.x request message are:

- **Record Type**: If a file is less than 200 bytes long, it can be written using a single message of Record Type 0 (according to the following **File Write Request** table). If a file is 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 7 data bits, the data must be encoded. Two methods are supported. The more efficient method requires an 8-bit data path (refer to the **8-Bit Encoding** table in this section). If only 7 bits are supported, use the 7-bit encoding (refer to the **7-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 less.
- **ASCII File Segment Number:** The first two bytes in the 6-byte reserved data field contain an ASCII file segment number. These two bytes effectively functions as a two-digit decimal value ranging from '1' to '99'. Once the segment number reaches '99' it will rollover to a new value of '01.

As a note, larger files may take some time to download via the 62.x message. It may 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 have been updated. Again, this process may take several minutes for larger files.

4_2_43_2 Aborting the Previous File Download

Provisions have been made to enable the previous file download to be aborted. If a New File wire operation is requested (62.x File Write Request message is received with a "Record Type" value of '0' or '1') then any previous file download will be aborted.

Refer to the following 62.x File Write Request Message Format table and 62.x File Write Response Message Format table for descriptions of the 62.x message.

62.x File Write Request Message Format

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX

Offset	Length	Type	Description
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. • 3 = Last record. • 9 = Current File Write request process needs to be aborted.
5	1	Alphanum	iConnectEFT Constant = P62_REQ_ENCODING_FORMAT Encoding format: • 7 = 7-bit encoding. • 8 = 8-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. Allowable values are: • 0 = Download file, unpack, and verify signing before sending to HOST (only accepts *.PGZ, *.TGZ, *.OGZ, and *.MP4) • 1 = Download file and send to HOST as-is (accepts any file type)
13	1	Binary	iConnectEFT Constant = P62_REQ_FAST_DOWNLOAD Enables fast download feature. Allowable values are: • 0 = Standard download speed. • 1 = Fast download speed. See also Info Note after the **Val1 – Val3** table regarding expected responses.

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

62.x File Write Response Message 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_RES_STATUS Response status: • 0 = Successful. • 1 = Out of order message. • 2 = File input/output error. • 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.

Info

If the fast download feature value in the message is set to 1, RBA will only ACK the message and will not send a response ’ 62.0’. Only one response will be sent for all 62.x messages. After the download is complete, the 62.x response status will indicate whether the download was successful or not.

8-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

7-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

4_2_44 63.x: Find File

4_2_44_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.

4_2_44_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.
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
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.

4_2_45 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
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.

4_2_46 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= '<?ivVAR3?>' 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".

4_2_46_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

4_2_46_2 Setting Button Text

The syntax for setting button text is as follows:

Bid,text

where

- id = the button ID from the K3Z form.
- text = the new button text.

Example: Set the button text for id "btn1" to "text":

Bbtn1,text

4_2_46_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:
>
> To show a button:
>
> Bid, S
>
> To hide a button:
>
> Bid, H
>
> Note the extra space following "Bid".

4_2_46_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.

Refer to 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.

4_2_47 82.x On-Guard and KME Session Key Injection Command

This command will allow the application to inject the On-Guard or KME Session Key.

This message is only used with On-Guard or KME encryption enabled.

4_2_48 83.x On-Guard and KME Enable Message

The 83.x On-Guard and KME Enable Message command is used to activate the E2EE encryption feature and update the E2ECFG file if all prerequisites are met:

- The selected cryptogram key must be in place in order for the command to be successfully implemented.
- The local storage key parameter requires that the selected key must exist in order for E2EE encryption to be enabled.

This message is only used with On-Guard or KME encryption enabled.

The purpose of this message is to enable the terminal to be preloaded with the required software and keys. If a valid version of this command is received while the terminal is already enabled, then the terminal parameters will be switched provided that all of the enabling checks for the new encryption parameters to be active have been met. The ability to switch will never cause the E2EE encryption to be disabled. When the POS and network are ready to process with encrypted data, the POS can send a single command to enable the feature. Once enabled, the only means of disabling E2EE encryption is to erase the terminal and reload all components.

Refer to the following tables which describe the 83.x On-Guard and KME Enable Request Message Format and 83.x On-Guard and KME Enable Response Message Format.

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 will return Track 2 only and support Base 24 framing. • B = Type B IngeCrypt will return 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 DUKPT

Offset	Length	Type	Description
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 should 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
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.

4_2_49 85.x On-Guard and KME Non-Payment Card Message

The 85.x On-Guard and KME Non-Payment Card message flows from the terminal to the POS. When On-Guard or KME encryption is enabled, this message will be sent in place of the standard 18.x: Non-Payment Card Message. 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 set to '1' (indicating a non-payment card type), then the terminal sends out the 85.x message. The card information will be encrypted prior to being incorporated into the message. As an expected change to the general release version, a variant of the standard 18.x message may be used in this case instead of a separate 85.x message.

This message is only used with On-Guard or KME encryption enabled.

Refer to the following table which describes the 85.x message format.

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.

4_2_50 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 in the process of pre-selecting the payment type for the customer. This message is similar to the 19.x message, with the difference being that the incorporated card data is encrypted. The 86.x message is used to determine the payment type corresponding to the swiped card. 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 RBA to indicate that a chip card has been detected. As an expected change to the general release version, a variant of the standard 19.x message may be used in this case instead of a separate 86.x message.

This message is only used with On-Guard or KME encryption enabled.

The 86.x message is also only relevant for Canadian applications using On-Guard or KME encryption. For other countries and encryption configurations, the 19.x message will supplant the 86.x. The 86.x message is deprecated, and may be removed from support in a future RBA release.

Refer to the following message formats. 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."
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.dfs file. Refer to DFS data indexes '0008_0006' ('h') and '0008_0007' ('d').
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 6 digits.

Offset	Length	Type	Description
18	1	Constant	ASCII control character – FS
19	4	Decimal	iConnectEFT Constant = P86_REQ_EXP_DATE Expiry Date (YYMM).
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."
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.

4_2_51 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. When the 87.x request message is issued by the POS, execution of the current transaction process is interrupted. The terminal then displays the "Card Swipe" screen, and waits for a card swipe on the magnetic stripe reader (MSR). When the MSR track data is available, the terminal returns an 87.x response message which includes the entered and encrypted data, returning to the interrupted process.

The 87.x message can be used even while the terminal is in an offline state.

The description of this command is similar to that of the 23.x: Card Read Request (On-Demand) message, with the difference being that this command uses encrypted card data. As an expected change to the general release version, a variant of the standard 23.x message may be used in place of a separate 87.x message.

This message is only used with On-Guard or KME encryption enabled.

87.x On-Guard and KME Card Read Data Request 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."
4	Variable	Alphanum	iConnectEFT Constant = P87_REQ_PROMPT_INDEX Prompt index number (up to 40 characters).
M	1	Constant	ASCII control character – FS. This field is optional, and is only used with the Form Name or Number.
M + 1	Variable	Alphanum	iConnectEFT Constant = P87_REQ_FORM_NAME Form Name or Number. This field is optional. When used, it is preceded by an 'FS' ASCII control character.
N	1	Constant	ASCII control character – ETX
N + 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."
4	1	Alphanum	iConnectEFT Constant = P87_RES_EXIT_TYPE Exit type: • 0 = Good read. • 1 = Bad read. • 2 = Cancelled. • 3 = Button pressed. • 7 = Encryption failed. • 9 = Declined.
5	1	Alphanum	iConnectEFT Constant = P87_RES_CARD_SOURCE Source of card read: • M = MSR • C = Contactless reader. This information is only included in the response message of '0013_0014' = '1'.
6	Variable	Decimal	iConnectEFT Constant = P87_RES_CARD_DATA Encrypted 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

4_2_52 88.x On-Guard and KME Translate Encrypted Card Data Command

This command will take transaction card data encrypted under the local storage key (AES or E2EE) and return card data encrypted under the appropriate E2EE key. This feature would generally be used only in a store-and-forward scenario.

> This message is only used with On-Guard or KME encryption enabled.

4_2_53 89.x On-Guard and KME Register BIN Record Command

This command allows the BIN table to be updated by the application. Subcommands are available which can change a single BIN table record, delete a single record, or delete all records. To change a BIN table record, the new record must be MACed using the CEFMK key.

> This message is only used with On-Guard or KME encryption enabled.

4_2_54 90.x: P2PE Data Message

The 90.x P2PE Data Message is used to send and receive cardholder P2PE data. It is also used to receive 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 the 90.x message usage.

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

4_2_54_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 requires about 10 seconds, there will be a delay before the response is received. If the public data has not been sent to RBA, no ETB will be 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”
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.

4_2_54_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.”
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.

4_2_54_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.
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 related failure).
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC check character.

4_2_54_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 will delete the public key stored under the file name provided in this request message. The file to be deleted will be located in the `/F_SECURITY_APP/RSAKEYS` directory and will contain 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 will be returned. If no key matching the Key Name is found then an error message will be returned ("No such key was found"). In either case, the encryption public key which is currently in use must not be deleted. An error response to this effect will be returned if an attempt is made to delete this key. The following tables provide the formats for 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”
5	Variable	Alphanum	Result Code. • 0 = Success. • 1 = Invalid request. • 2 = No such key was found. • 3 = Key is currently in use. • 4 = Failed to delete key (file system related failure).
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC check character.

4_2_54_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 will locate and load the public key stored under the file name provided in this request message. The file to be loaded will be located in the `/F_SECURITY_APP/RSAKEYS` directory and will contain a ".PEM" file extension.Upon successfully loading this key, RSA-OAEP encryption will be full enabled using the selected key. Security parameter '0091_0033' will be updated to preserve this key selection following reboots. If for any reason the selected key is not successfully loaded, then the key currently in use will continue to be used for card data encryption. If there is no encryption public key currently loaded then this request message is received, the RBA will enter the offline state and return an error message upon an encryption attempt.

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 matching the Key Name is found then an error message will be returned ("No such key was found"). The following tables provide the formats for 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 such key was found. • 3 = Invalid public key. • 4 = Failed to select key (file system related failure).
6	1	Constant	ASCII Control Character – ETX
7	1	Binary	LRC check character.

4_2_55 91.x: Print Message

4_2_55_1 Overview of the 91.x Printer 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 that needs to be printed. 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 of 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”. This will then cause the receipt to be printed. If the POS wants to cancel a partial receipt, a type “C” message can be sent.

Refer to the 91.x: Barcode Printing section for a description of the commands used to configure Barcode Printing parameters.

4_2_55_2 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.”
4	1	ASCII	iConnectEFT Constant = P91_RES_RESULT Result • 0 = Successful • 1 = Successful – printer paper low (not currently implemented on iWL250 printer) • 2 = Paper jam • 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.
<@BoldOn>	<@BoldOff>	Bolds text
<@ReverseOn>	<@ReverseOff>	White text surrounded by black
<@DoubleCharOn>	<@DoubleCharOff>	Double wide characters
<@BigCharOn>	<@BigCharOff>	Very Large font Three (3) 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 currently used

Number of Characters Allowed by Tag Type

Font	Default	<@DoubleCharOn>	<@BigCharOn>
Normal	40	20	6
Bold	40	20	n/a

4_2_55_3 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.

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>
<@BarCodeEAN13>	Switch to EAN-13 Format (12 digits)	<@BarCodeEAN13><@BarCodeOn>123456789012<@BarCodeOff>

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>

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>

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>
<@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>

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>

4_2_56 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
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.

4_2_57 94.x and 95.x: Barcode Configuration Messages

4_2_57_1 Barcode Configuration Messages

There are two message types which have been added for 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 RBA 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] 21'
RSA Encryption Exponent	--	--	(see Note 1)	'61.91[GS] 22'
RSA Encryption Key ID	--	--	(see Note 1)	'61.91[GS] 23'

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.

4_2_57_2 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'
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.

4_2_57_3 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

4_2_57_4 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.
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').

4_2_57_5 Barcode Encryption Messages

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.
6	2	Decimal	iConnectEFT Constant = P94_REQ_ACTION_CODE Action code: • 01 = Enabled (see Note 1).
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.

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.

Note 1: Configuration only enables encryption for 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.

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).
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
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.

4_2_57_6 Barcode Illumination Messages

Barcode Configure Illumination Mode Messages

Barcode Configure Illumination Mode Request Message

<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>iConnectEFT Constant = M94_BARCODE_SET
ASCII Message identifier, "94."</td></tr>
<tr><td>4</td><td>2</td><td>Decimal</td><td>iConnectEFT Constant = P94_REQ_TYPE_CODE
Type code:
12 = Illumination mode.</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
8	1	Binary	LRC check character.

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

<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>iConnectEFT Constant = M95_BARCODE_GET
ASCII Message identifier, "95."</td></tr>
<tr><td>4</td><td>2</td><td>Decimal</td><td>iConnectEFT Constant = P95_RES_TYPE_CODE
Type code:
12 = Illumination mode.</td></tr>
<tr><td>6</td><td>1</td><td>Alphanumeric</td><td>iConnectEFT Constant = P95_RES_STATUS
Status/Error code:
0 = No error.
1 = (reserved; do not use).
2 = Generic error.
C = Configuration error.</td></tr>
<tr><td>7</td><td>2</td><td>Decimal</td><td>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></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 there is any error, then the Action Code will not be returned.

4_2_57_7 Barcode Image Messages

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.
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.

Barcode Read Image Mode Configuration Messages

Barcode Read Image 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: • 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
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.

4_2_57_8 Barcode Lighting Messages

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."
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.

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."
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.

4_2_57_9 Barcode Power Messages

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."
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.

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.

4_2_57_10 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.
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.

4_2_57_11 Barcode Scan Messages

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.

Barcode Get Scan Messages

Barcode Get Scan 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: • 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.

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.
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.

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.
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.

4_2_57_12 Barcode Symbology Messages

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).
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.

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).
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.

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
13	Code 39
14	*** N/A ***
15	Interleaved 2 of 5
16	Standard 2 of 5
17	Matrix 2 of 5

Code	Description
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

Code	Description
39	GSI_DataBarExpanded
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

Code	Description
61	Postnet
62	Planet
63	BPO
64	Canada Post
65	Australian Post
66	Japan Post
67	Dutch Post
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

4_2_57_13 Barcode Trigger Messages

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
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.

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.

4_2_58 97.x: Reboot

This message reboots the terminal.

Reboot Request Message

Offset	Length	Type	Description
0	1	Constant	ASCII control character – STX
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.

5_Forms

The RBA uses form files to position text, buttons, and graphics on the screen, and to select colors or font types. The form file name can be 12 characters long, including the dot (.) and K3Z extension. 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.

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 "<?ivPROMPT3?>". 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.

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 "<? ivPROMPT10?>". Prompt 10 is "Amount OK? $<?ivTOTAL?>". If the transaction total is $123.45, the prompt will display as "Amount OK? $123.45".

Special Text Display

Special Text	Description
<?ivPROMPT99?>	Insert a specific prompt from the prompt file.
<?ivCARDNUMBER?>	Displays account number with all but last X digits as Xs. The number of digits shown is set by config setting '0003_0011'.
<?ivAMOUNT?>	Purchase amount.
<?ivCASHBACK?>	Cash back amount selected.
<?ivCB_MAX?>	Maximum cash back allowed.
<?ivCB_INC?>	Cash back increment amount.
<?ivCB1?>	Amount for Fast Cash Back Key 1.
<?ivCB2?>	Amount for Fast Cash Back Key.
<?ivCB3?>	Amount for Fast Cash Back Key 3.
<?ivCB4?>	Amount for Fast Cash Back Key 4.
<?ivEXPDATE?>	Expiration Date from card.
<?ivCARDHOLDERNAME?>	Cardholder's name (taken from Track 2 of payment card).
<?ivTOTAL?>	Purchase amount + cash back amount.
<?ivVAR1?> through <?ivVAR25?>	User-defined variable (1-25). Set through the file `var.dat` or the 28.x message.
<?ivTYPE?>	Card type defined in `cards.dat`.

5_1 Form Contents and Descriptions

This section includes form descriptions which are used in transactions for Telium terminals. Each form 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 table lists all forms, form file names, and a brief descriptions. All entries (form names) in this table are also linked to the actual form description pages.

- Button IDs and Images - This section includes a list of buttons, button IDs, and button images for Telium terminals.
- Status Messages (status.dat) - For status message indexes or below for Button IDs.

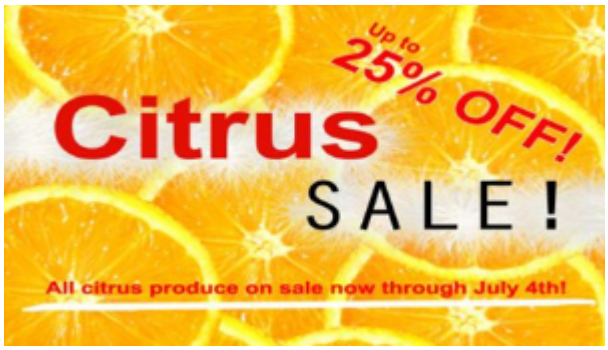
New standard forms for the iSC250, iSC350 and iSC480 are now red with grey whereas old forms are blue with white.

Each of the PayPal versions noted in this section includes a PayPal button in place of the Enter Card button.

All iSC480 forms that were copy-pasted from iSC350 and had the same 44-column line display width now are 58 characters long to fit the screen size.

5_1_1 Advertising

iPP320	iPP350 • iWL250

iSC250	iSC350
	

iSC480	iSMP • iSMPc

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

5_1_2 Approved/Disapproved

iPP320	iPP350 • iWL250

iSC250 • iSC350 • iSC480	iSMP • iSMPc

iCMP

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	

5_1_3 Card Swipe Forms

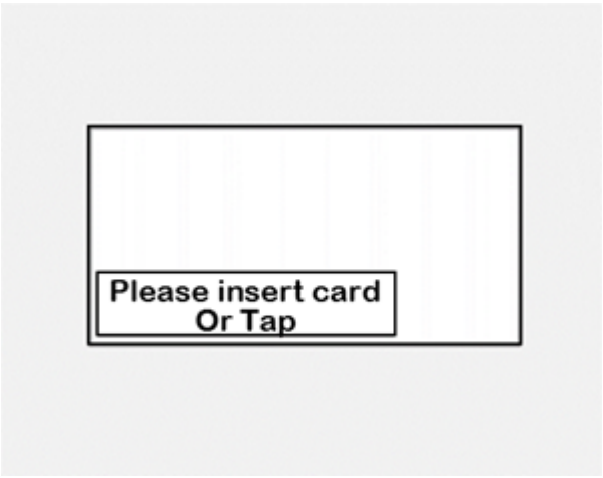
This section includes the following forms:

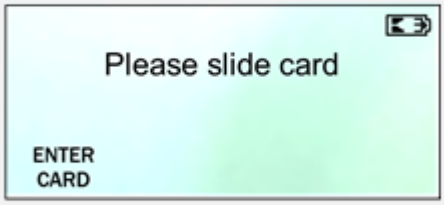
- Card Swipe
- Card Swipe with Language Selection
- Card Swipe On Demand

5_1_3_1 Card Swipe

iPP320	**iPP350 • iWL250**

iSC250	**iSC350**

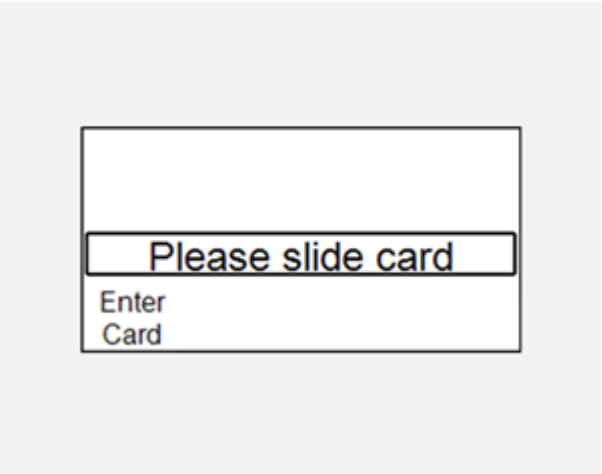
iSC480	iUP250
	

iSMP • iSMPc	iCMP
	

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.

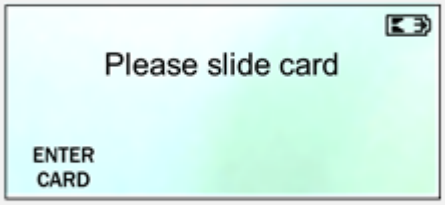
5_1_3_2 Card Swipe On Demand

iPP320	**iPP350 • iWL250**
	

iSC250	iSC350

iSC480	iSMP • iSMPc

iCMP



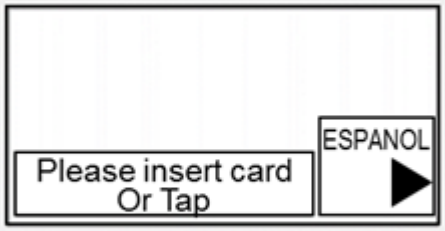
Description	Card read request (prompts user to swipe magnetic stripe card)
Initiated By	23.x Card Read Request message
Status Response upon display	11.12Input
Status Response after swipe	11.13Input Accepted
DFS Data Index	0030_0014
Form	COD.K3Z
Text	Variable
Form Buttons and IDs	iSC250/iSC350/iSC480 • "ENTER CARD"
Terminal Buttons Allowed	Cancel

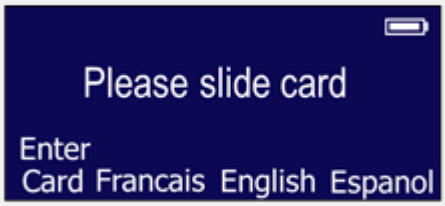
The "Enter Card" button for manual entry is hidden when the Display “Enter Card” Prompt (configuration parameter '0007_0029') is set to '0'. In order for the manual entry button to be displayed, this configuration parameter must be set to a value of '1' to '4' as described in the Main Flow (mainFlow.dat) section of this document.

5_1_3_3 Card Swipe with Language Selection

iPP320	**iPP350 • iWL250**

iSC250	**iSC350**

iSC480	iUP250
	

iSMP • iSMPc	iCMP
	

<table>
<tr><td>Description</td><td>Card swipe with language. Settings are as follows:<table><tr><th>Parameter</th><th>Setting</th><th>Description</th></tr><tr><td>0007_0004</td><td>1</td><td>Combine language and swipe screen.</td></tr><tr><td>0007_0005</td><td>0</td><td>Swipe card first, then select payment type.</td></tr></table></td></tr>
<tr><td>Initiated By</td><td>01.x Online message</td></tr>
<tr><td>Status Response</td><td>11.01SlideCard</td></tr>
<tr><td>DFS Data Index</td><td>0030_0005</td></tr>
<tr><td>Form</td><td>LSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Please slide card”</td></tr>
<tr><td>Form Buttons and IDs</td><td>iPP320/iPP350/iWL250/iMP350
“Enter Card” – M
“English” – 1
“Francais” – 3
“Español” – 2
iSC250/iSC350/iSC480
"ENTER CARD" - M
"LANGUAGE"
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.</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

The PayPal version of this form is `PLSWIPE.K3Z`. Note that the PayPal form is not available for iPP320 and iPP350 terminals.

5_1_3_4 Remove Inserted Card

This form is specific to the iUP250 terminal.

iUP250

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

5_1_4 Cash Back Forms

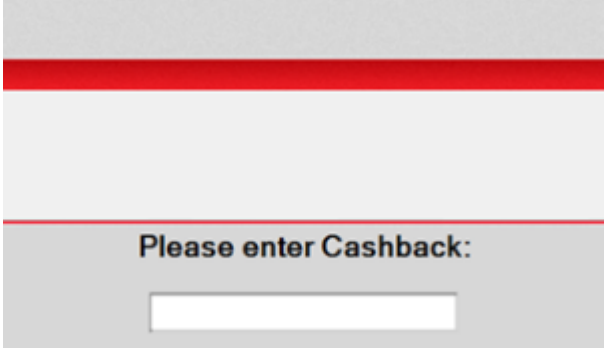
This section includes the following cash back forms:

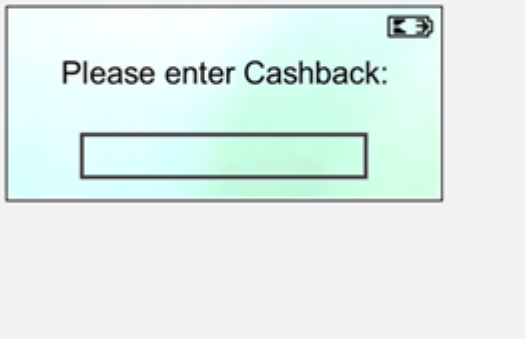
- Cash Back Other
- Cash Back Selection
- Cash Back Selection without No

- Cash Back Verification

5_1_4_1 Cash Back Other

iPP320	iPP350 • iWL250

iSC250 • iSC350 • iSC480	iSMP • iSMPc
	

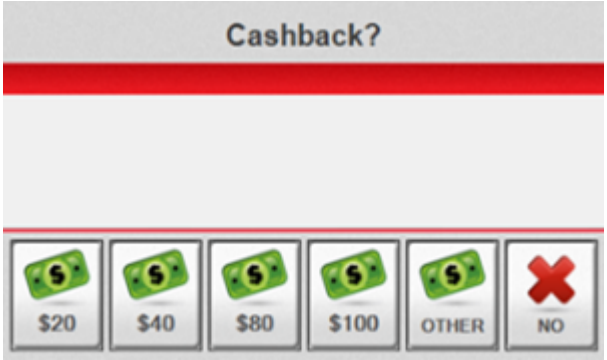
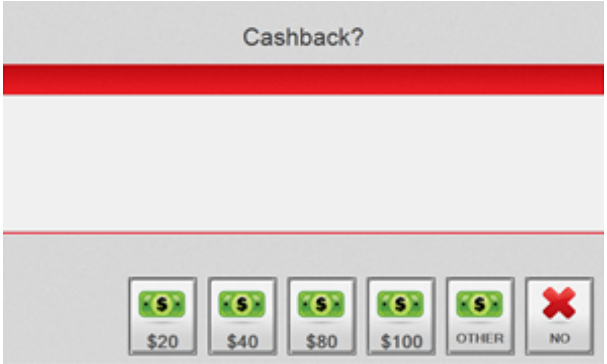
iCMP


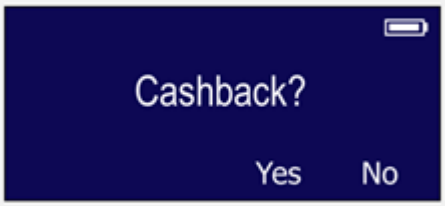
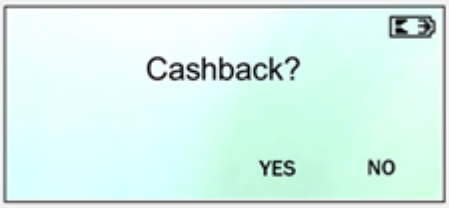
Description	Manual entry of cash back amount
Initiated By	Other button in cash back selection forms
Status Response	11.04Cash Back
DFS Data Index	0030_0018
Form	CASHBO.K3Z
Text	“Please Enter cash back:”
Form Buttons and IDs	N/A

5_1_4_2 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.

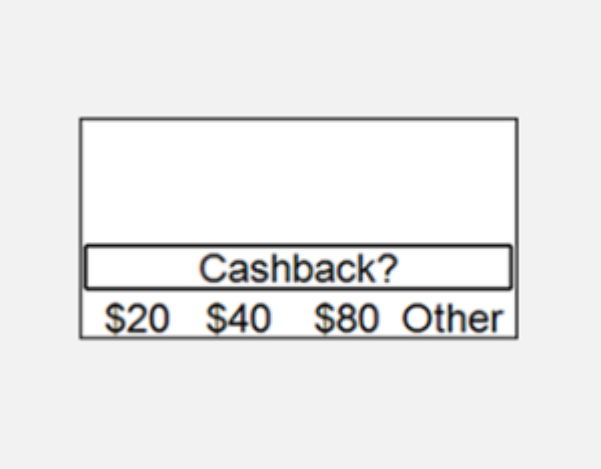
iPP320	iPP350 • iWL250

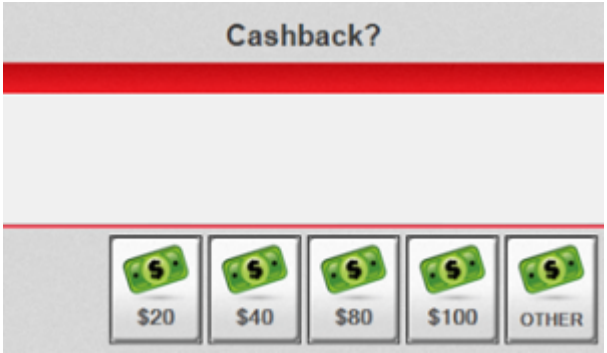
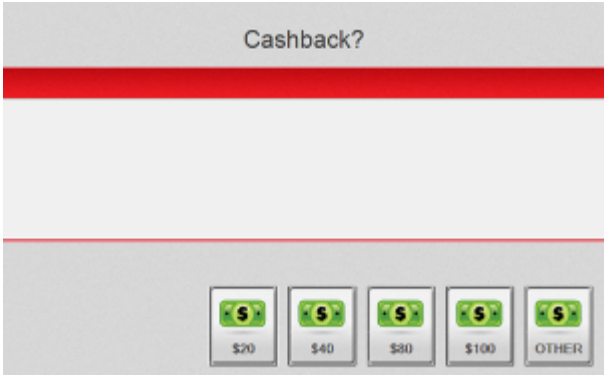
iSC250 • iSC350	iSC480
	

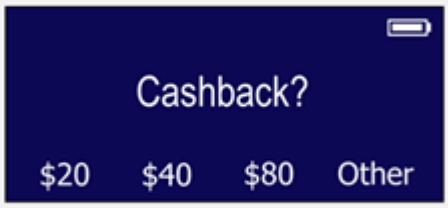
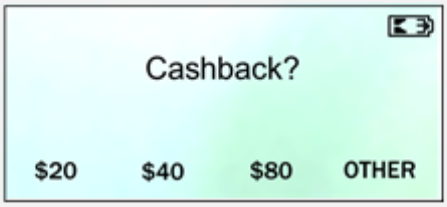
iSMP • iSMPc	iCMP
	

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	• “Yes” – Y, iPP3x0 only • “No” – N • “Other” – O, iSCx50 only • <?ivCB1?> – 1, iSCx50 only • <?ivCB3?> – 2, iSCx50 only • <?ivCB2?> – 3, iSCx50 only • <?ivCB4?> – 4, iSCx50 only
Terminal Buttons Allowed	Cancel

5_1_4_3 Cash Back Selection without No

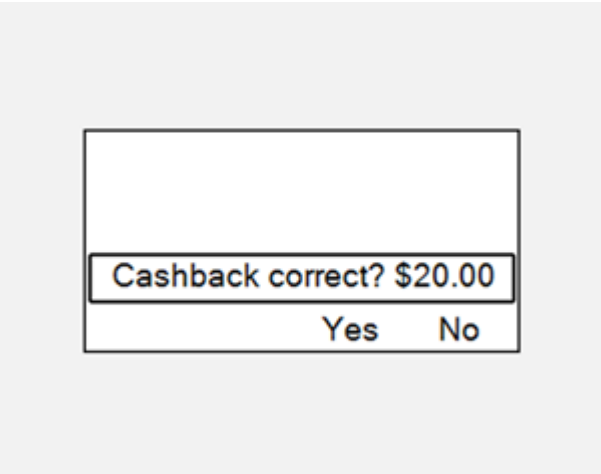
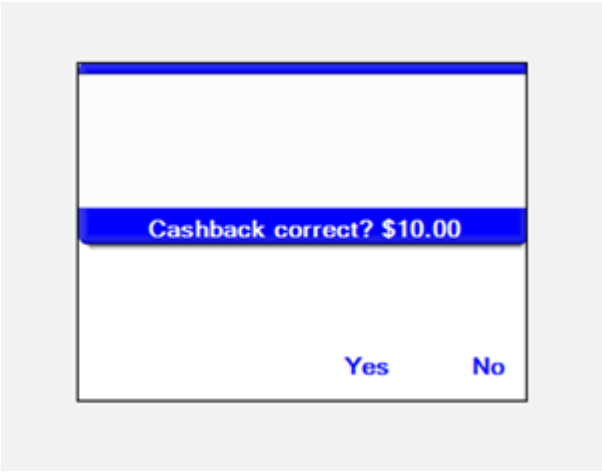
iPP320	iPP350 • iWL250
	

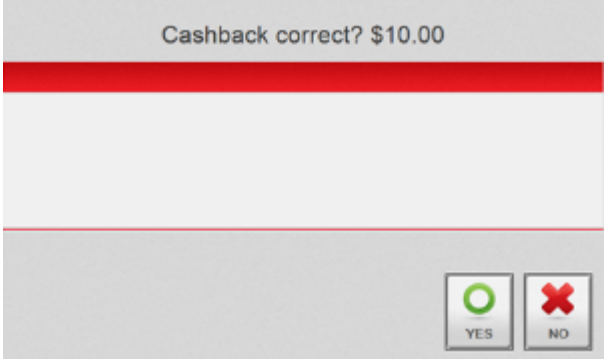
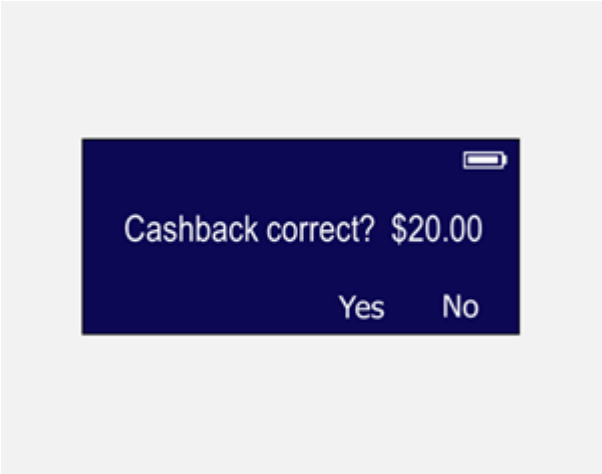
iSC250 • iSC350	iSC480
	

iSMP • iSMPc	iCMP
	

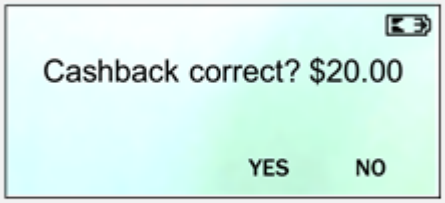
Description	Cash back selection without No button
Initiated By	Cash back being enabled, after debit or EBT selection
Status Response	11.04Cash Back
DFS Data Index	0030_0008
Form	CASHBA.K3Z
Text	“Cash back?”
Form Buttons and IDs	• “Other” • <?ivCB1?> – 1 • <?ivCB3?> – 2 • <?ivCB2?> – 3 • <?ivCB4?> – 4
Terminal Buttons Allowed	Cancel

5_1_4_4 Cash Back Verification

iPP320	iPP350 • iWL250
	

iSC250 • iSC350 • iSC480	iSMP • iSMPc
	

iCMP



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

5_1_5 Contactless Enabled Forms

This section includes the following contactless forms:

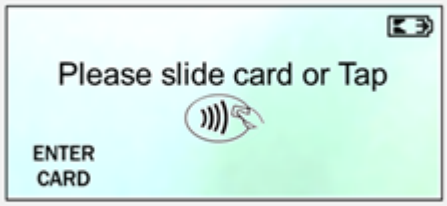
- Contactless Card Read Request
- Contactless Card Swipe
- Contactless Card Swipe with Language Selection

5_1_5_1 Contactless Card Read Request

iPP320	**iPP350 • iWL250**

iSC250	**iSC350**

iSC480	iSMP • iSMPc

iCMP


<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

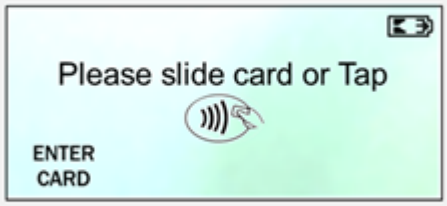
The "Enter Card" button for manual entry is hidden when the Display “Enter Card” Prompt (configuration parameter '0007_0029') is set to '0'. In order for the manual entry button to be displayed, this configuration parameter must be set to a value of '1' to '4' as described in the Main Flow (mainFlow.dat) section of this document.

5_1_5_2 Contactless Card Swipe

iPP320	iPP350 • iWL250

iSC250	iSC350

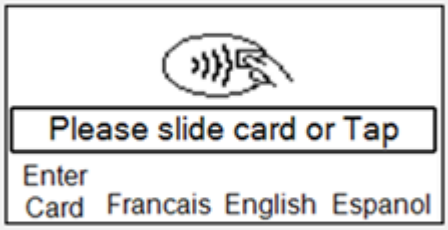
iSC480	iSMP • iSMPc

iCMP


<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</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.

5_1_5_3 Contactless Card Swipe with Language Selection

iPP320	iPP350 • iWL250
	

iSC250	iSC350

iSC480	iSMP • iSMPc

iCMP

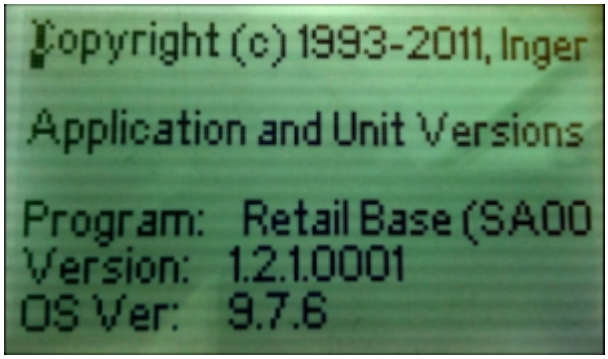
Please slide card or Tap
ENTER CARD
FRANCAIS
ENGLISH
ESPANOL

<table>
<tr><td>Description</td><td>Contactless card swipe with 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>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></table></td></tr>
<tr><td>Initiated By</td><td>01.x Online message</td></tr>
<tr><td>Status Response</td><td>11.01SlideCard</td></tr>
<tr><td>DFS Data Index</td><td>0030_0022</td></tr>
<tr><td>Form</td><td>CLSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Please slide card or Tap”</td></tr>
<tr><td>Form Buttons and IDs</td><td>iPP320/iPP350/iWL250/iMP350
“Enter Card” – M
“English” – 1
“Francais” – 3
“Español” – 2
iSC250/iSC350/iSC480
"ENTER CARD"
"Language"</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

Info

The PayPal version of this form is `CPLSWIPE.K3Z`. Note that the PayPal form is not available for iPP320 and iPP350 terminals.

5_1_6 Initialization Screen

iPP320	**iPP350 • iWL250**
	

iSC250 • iSC350 • iSC480	**iSMP • iSMPc**

iCMP

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

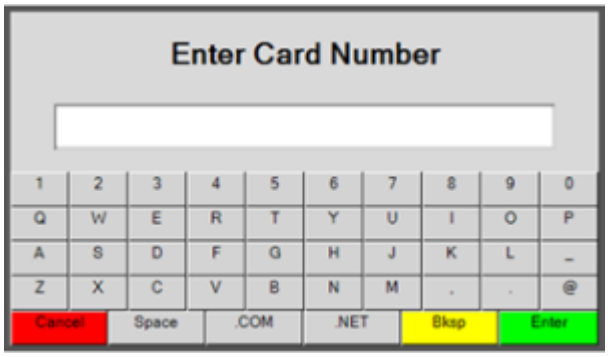
5_1_7 Input Entry Forms

This section includes the following forms:

- Alphanumeric Entry
- Input Entry

5_1_7_1 Alphanumeric Entry

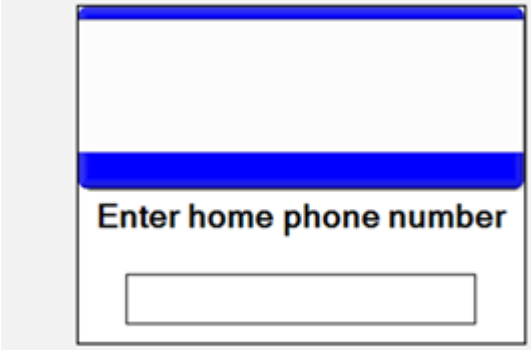
iSC250 • iSC350 • iSC480

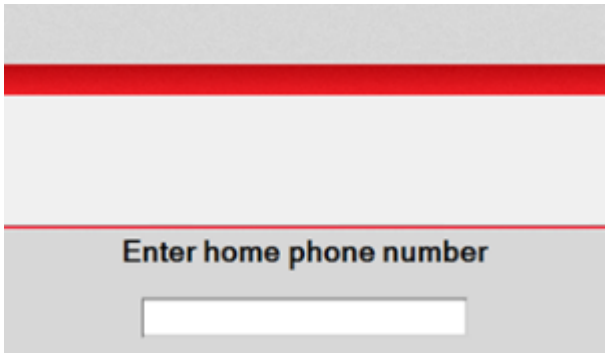
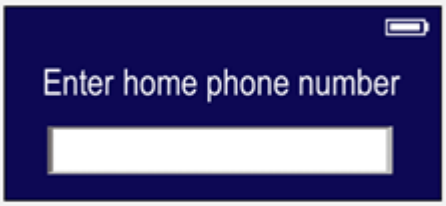


This form is not available for the iCMP, iSMP, iSMPc, iPP320, iPP350 or iWL250.

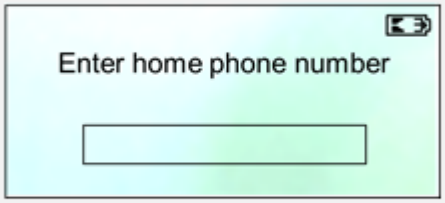
Description	Data entry form
Initiated By	27.x: Alpha Input Message (On-Demand)
Status Response upon display	11.12 Input
Status Response after data entry	11.13 Input Accepted
DFS Data Index	0030_0017
Form	ALPHA.K3Z
Text	Variable
Form Buttons and IDs	• Alphanumeric keyboard (for alpha entry) • “Cancel” • “Backspace” • “Enter”
Terminal Buttons Allowed	All

5_1_7_2 Input Entry

iPP320	iPP350 • iWL250
	

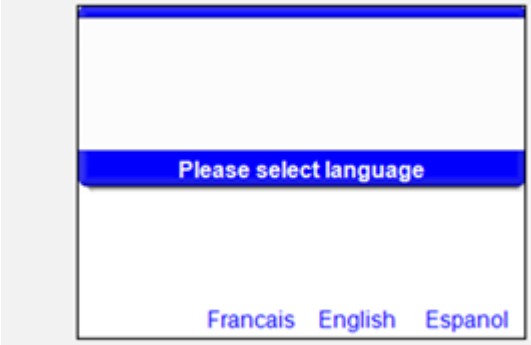
iSC250 • iSC350 • iSC480	iSMP • iSMPc
	

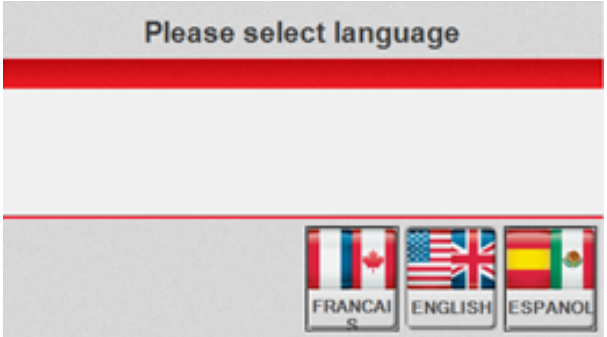
iCMP



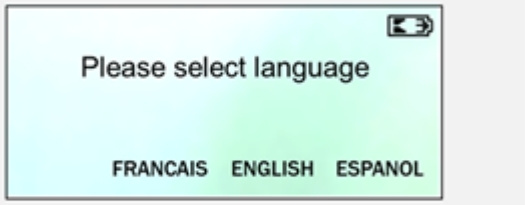
Description	Numeric input entry
Initiated By	21.x: Numeric Input Request Message (On-Demand)
Status Response upon display	11.12Input
Status Response after swipe	11.13Input Accepted
DFS Data Index	0030_0020
Form	INPUT.K3Z
Text	Variable
Form Buttons and IDs	N/A
Terminal Buttons Allowed	All

5_1_8 Language Selection

iPP320	iPP350 • iWL250
	

iSC250 • iSC350 • iSC480	iSMP • iSMPc
	

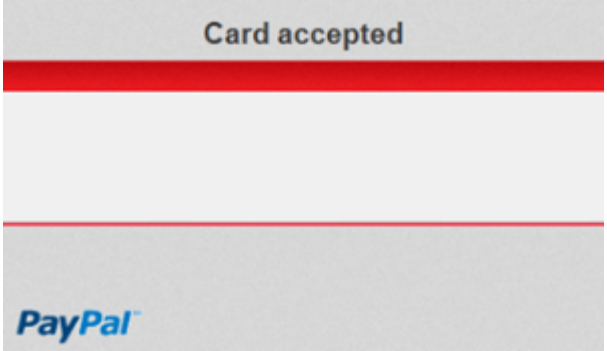
iCMP

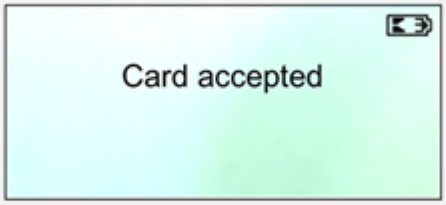


Description	Language selection
Initiated By	Flag in mainflow.dat
Status Response	
DFS Data Index	0030_0003
Form	LANG.K3Z
Text	“Please select language”
Form Buttons and IDs	• “French” or “Francais” – 3 • “English” – 1 • “Español” – 2
Terminal Buttons Allowed	Cancel

5_1_9 Message

iPP320	iPP350 • iWL250

iSC250 • iSC350 • iSC480	iUP250
	

iSMP • iSMPc	iCMP
	

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.

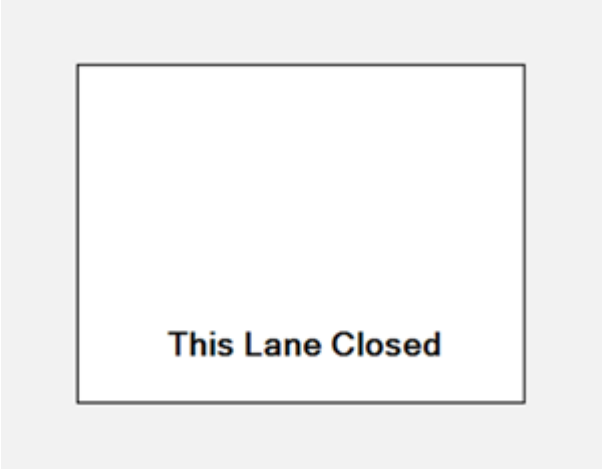
Description	Status response
Initiated By	After amount verification, waiting on host response
Status Response	11.05Processing
DFS Data Index	0030_0002
Form	MSG.K3Z
Text	“Processing …Please wait”
Form Buttons and IDs	N/A
Terminal Buttons Allowed	N/A

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

5_1_10 Offline

iPP320	iPP350 • iWL250
	

iSC250 • iSC350 • iSC480	iSMP • iSMPc

iCMP

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

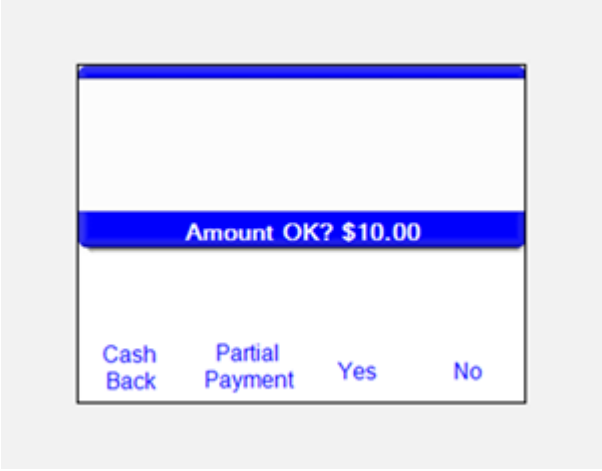
The PayPal version of this form is `PPOFFLINE.K3Z`. Note that the PayPal form is not available for iPP320 and iPP350 terminals.

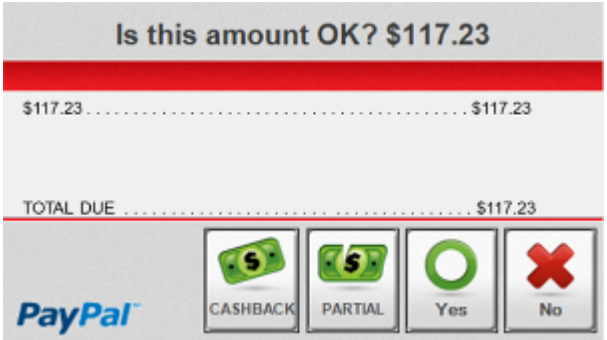
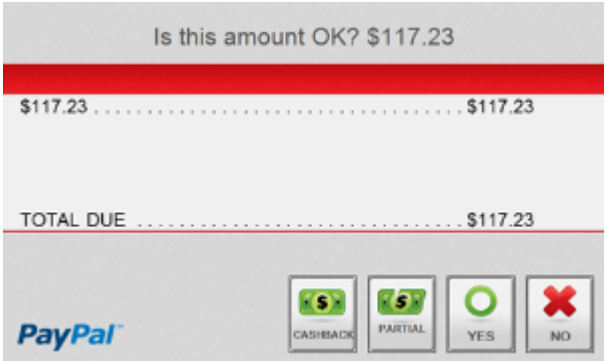
5_1_11 Payment Forms

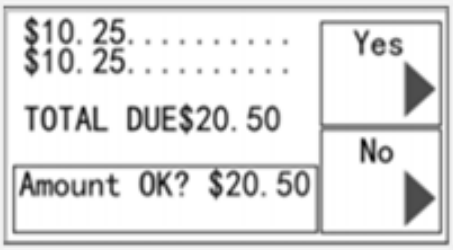
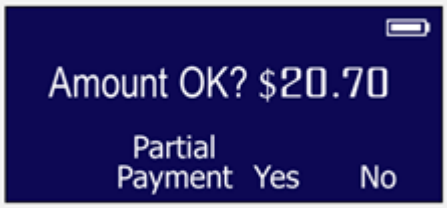
This section includes the following forms:

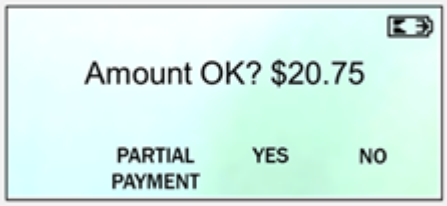
- Payment Selection
- Amount Verification

5_1_11_1 Amount Verification

iPP320	iPP350 • iWL250
	

iSC250 • iSC350	iSC480
	

iUP250	iSMP • iSMPc
	

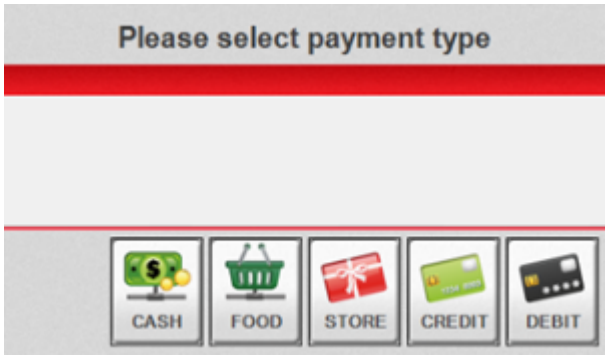
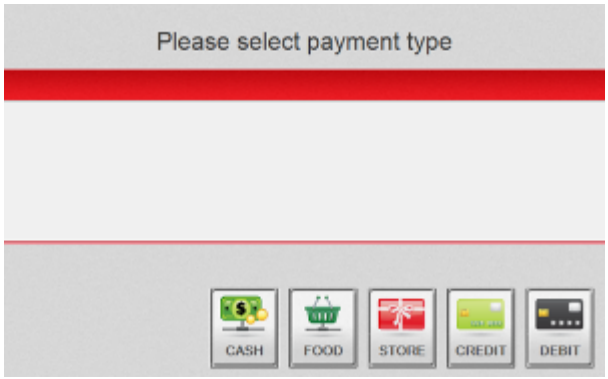
iCMP


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

5_1_11_2 Payment Selection

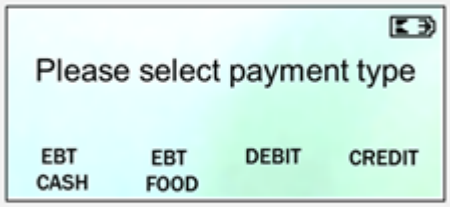
> **Info**
>
> The form format discussed below applies to the default payment selection screen (`pay1.K3Z`) as well as any custom payment screens that may be added (`pay2.K3Z` through `pay9.K3Z`).

iPP320	iPP350 • iWL250

iSC250 • iSC350	iSC480
	

iUP250	iSMP • iSMPc

iCMP



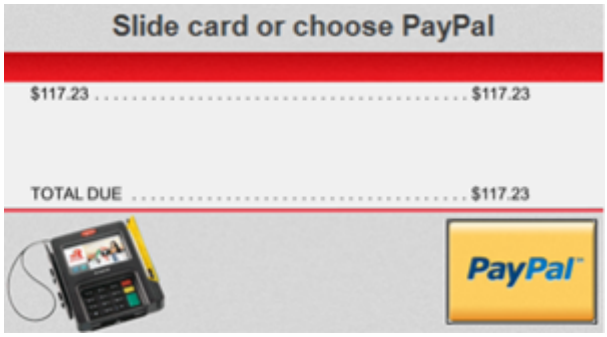
Description	Select Payment Type
Initiated By	Card swipe
Status Response	11.02Select Payment
DFS Data Index	0030_0006
Form	PAY%d.K3Z (%d Replaced by numbers 1-9)
Text	“Please select payment type”
Form Buttons and IDs	iPP320/iPP350/iWL250 • “EBT Cash” – C • “EBT Food” – D • “Debit” – A • “Credit” – B iSC250/iSC350/iSC480 • iPP320/iPP350/iWL250 • “EBT Cash” – C • “EBT Food” – D • “Store” – E • “Debit” – A • “Credit” – B
Terminal Buttons Allowed	Cancel

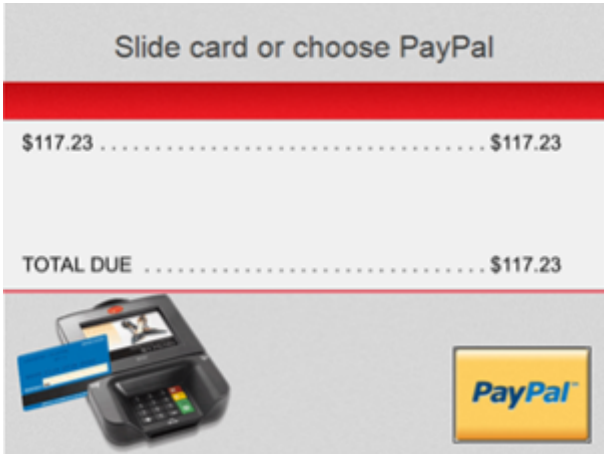
5_1_12 PayPal Forms

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

5_1_12_1 Card Swipe with PayPal

iPP350	iSC250
	

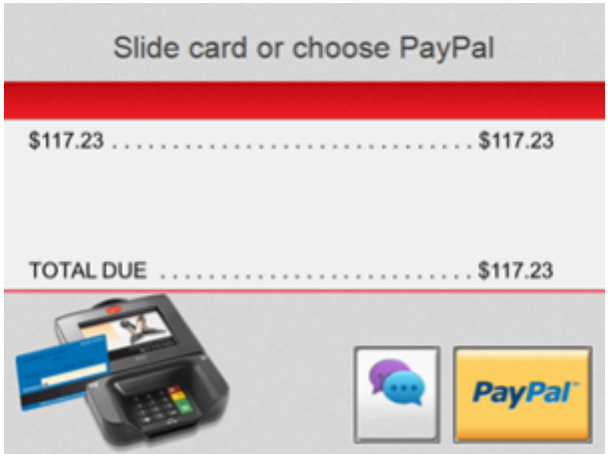
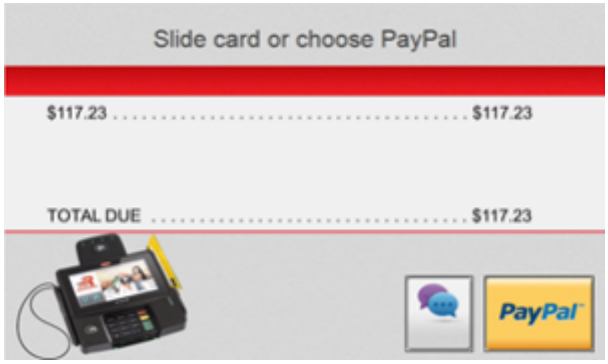
iSC350	iSC480
	

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>

5_1_12_2 Card Swipe with PayPal and Language Selection

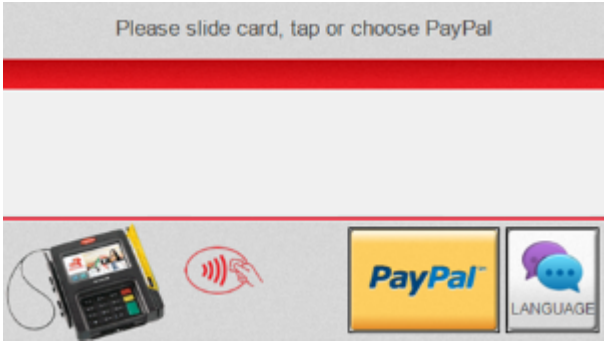
iPP350	**iSC250**

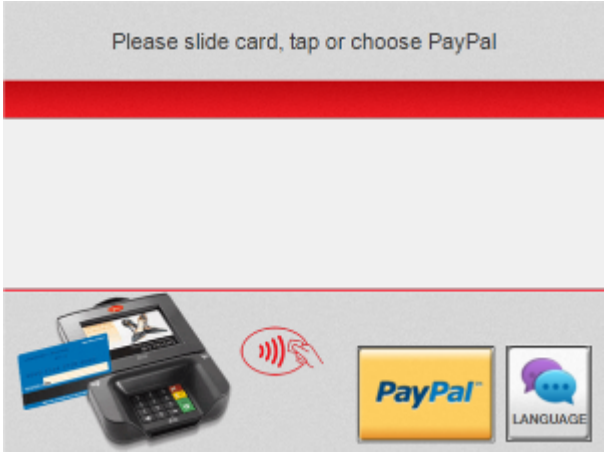
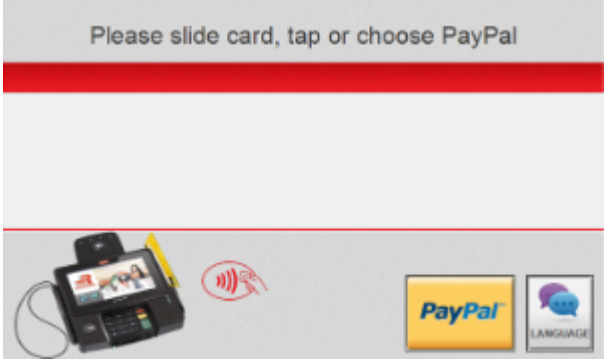
iSC350	iSC480
	

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>
<tr><td>Initiated By</td><td>01.x Online message 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_0005</td></tr>
<tr><td>Form</td><td>PLSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Slide card or choose PayPal"</td></tr>
<tr><td>Form Buttons and IDs</td><td>iSC250/iSC350/iSC480
"PayPal" - P
"Language"
iPP350
"PayPal" - P
"Francais" - 3
"English" - 1
"Espanol" - 2</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

5_1_12_3 Contactless Card Swipe with PayPal and Language Selection

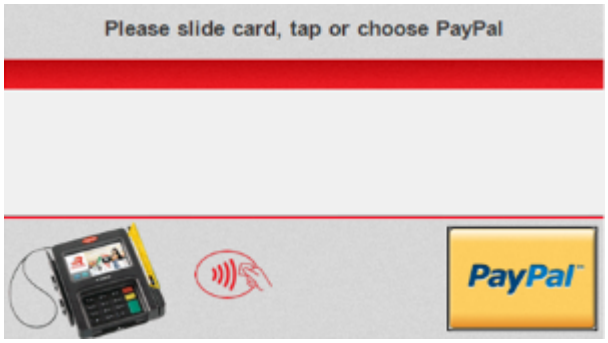
iPP350	**iSC250**
	

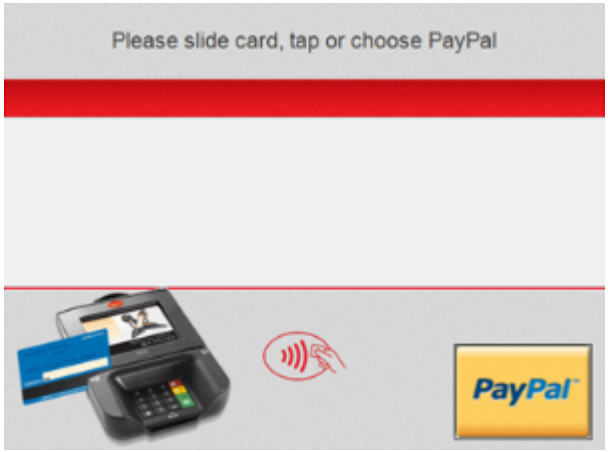
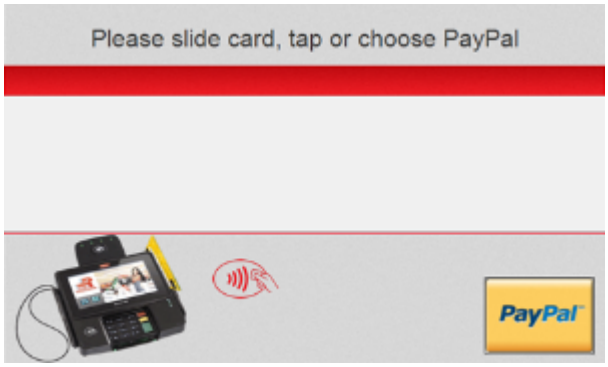
iSC350	**iSC480**
	

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>

5_1_12_4 Contactless Card Swipe with PayPal Selection

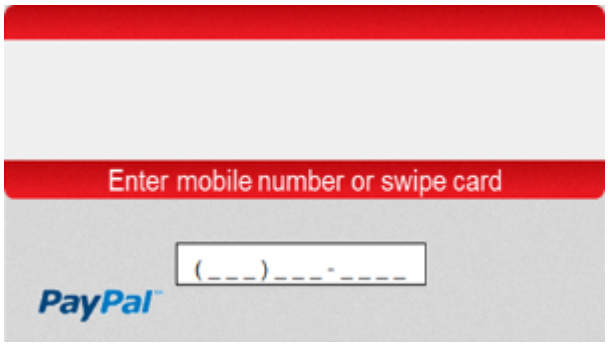
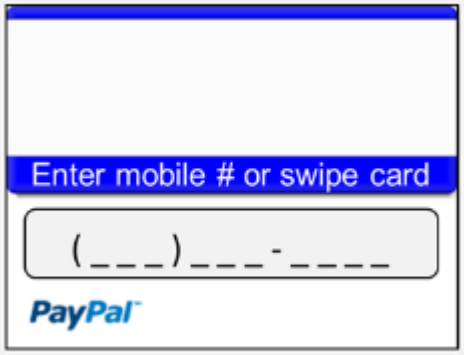
iPP350	iSC250
	

iSC350	iSC480
	

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>
<tr><td>Status Response</td><td>11.01 Slide/Insert/Tap Card</td></tr>
<tr><td>DFS Data Index</td><td>0030_0021</td></tr>
<tr><td>Form</td><td>CPSWIPE.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>"PayPal" - P</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

5_1_12_5 PayPal Data Input (On-Demand form)

iSC250 • iSC350 • iSC480	iPP350
	

This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

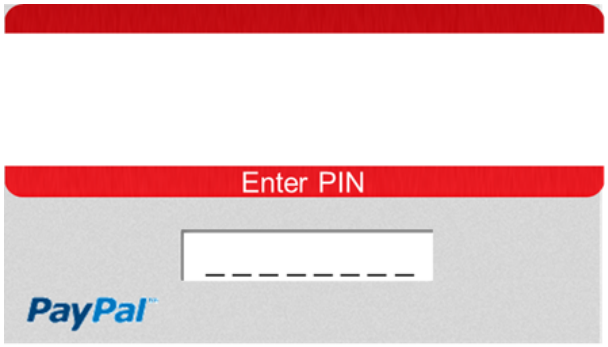
Description	PayPal data input form (zip code or phone number, for instance)
Initiated By	21.x message
Status Response	11.12 Input Accepted/Declined
DFS Data Index	0030_0027
Form	PPALINP.HTM
Text	“Enter mobile number or swipe card"
Form Buttons and IDs	N/A
Terminal Buttons Allowed	All

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.

5_1_12_6 PayPal PIN Entry

iSC250 • iSC350 • iSC480	iPP350
	

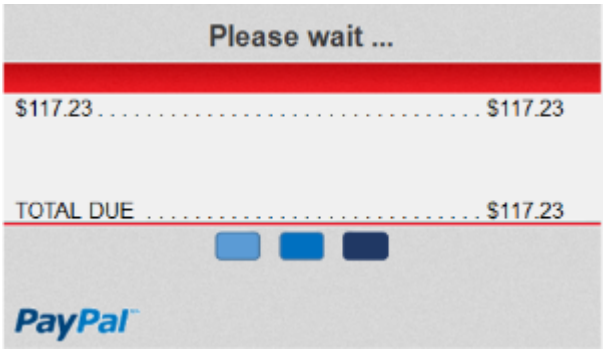
This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

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.

5_1_12_7 PayPal Please Wait

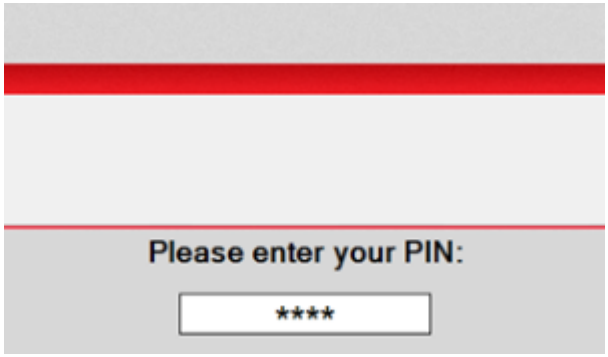
iSC250 • iSC350 • iSC480	iPP350
	

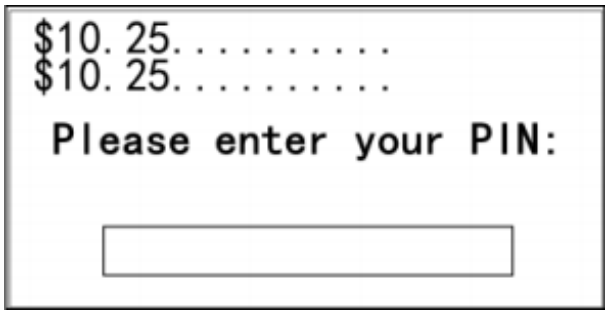
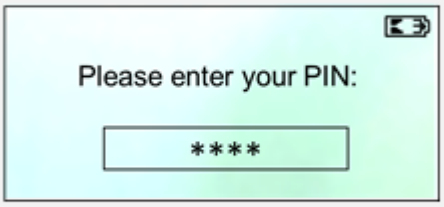
This form is not available for the iCMP, iSMP, iSMPc, iPP320, iUN250 or iWL250.

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

5_1_13 PIN Entry

iPP320	**iPP350 • iWL250**

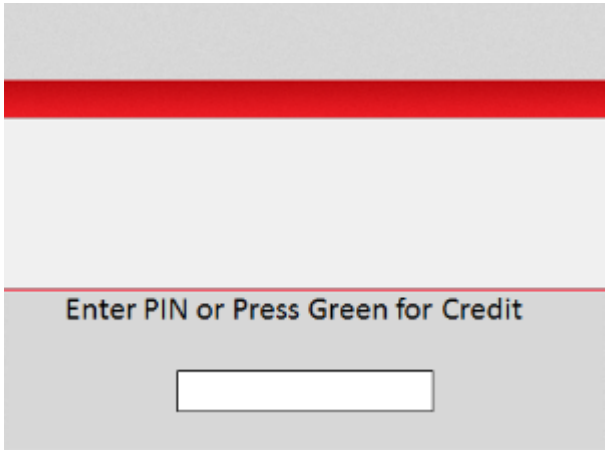
iSC250 • iSC350 • iSC480	iSMP • iSMPc
	

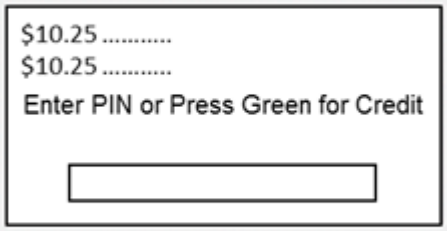
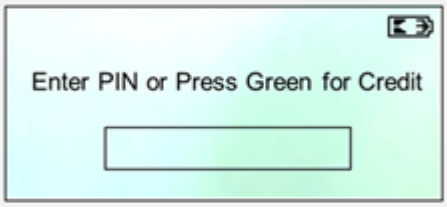
iUP250	iCMP
	

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

5_1_14 Enter PIN or Press Green for Credit

iPP320	**iPP350 • iWL250**

iSC250 • iSC350 • iSC480	iSMP • iSMPc
	

iUP250	iCMP
	

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

5_1_15 Signature Forms

This section includes the following forms:

- Post-Sign
- Pre-Sign
- Signature (On-Demand)

5_1_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," "Enter" 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."

iSC250 • iSC350 • iSC480

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 ENTER with pen"
Status Response after signature	11.11 "Signature Accepted"
DFS Data Index	0030_0011
Form	POSTSIGN.K3Z
Text	“Please sign and tap ENTER with pen”
Form Buttons and IDs	• “Enter" • "Clear
Terminal Buttons Allowed	Enter and Clear.

5_1_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," "Enter" 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.

iSC250 • iSC350 • iSC480

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 ENTER with pen"
Status Response after signature	11.11 "Signature Accepted"
DFS Data Index	0030_0040
Form	PRESIGN.K3Z
Text	“Please sign and tap ENTER with pen”
Form Buttons and IDs	• “Enter" • "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.

5_1_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," "Enter" 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.

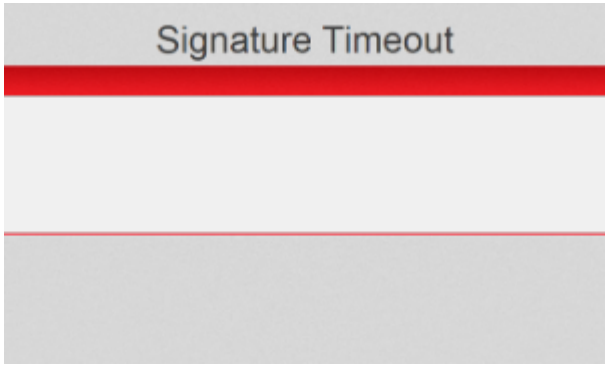
Cancel Button and Physical Key Function During Signature Capture Process

	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

Signature Timeout Form



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 ENTER with pen"
Status Response after signature	11.11 "Signature Accepted"
DFS Data Index	0030_0041
Form	SIGN.K3Z
Text	“Please sign and tap ENTER 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	• “Enter" • "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.

5_1_16 Smart Card (SMC) and EMV Forms

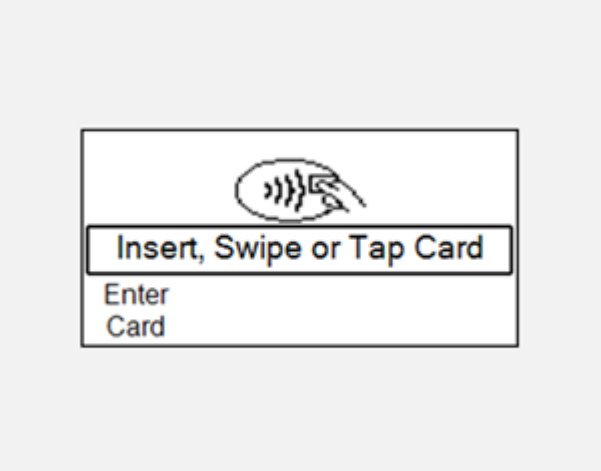
This section includes the following forms:

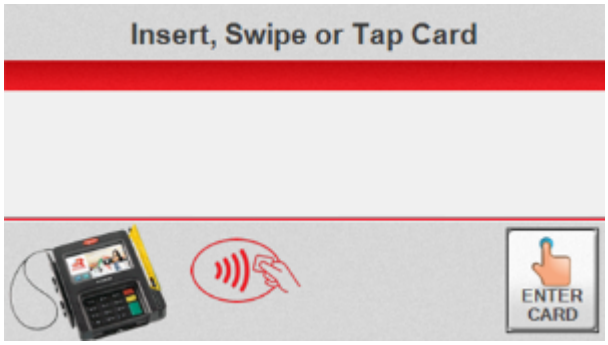
- Contactless Smart Card (EMV) and Swipe
- Contactless Smart Card (EMV) and Swipe with Language Selection
- Smart Card (EMV) and Swipe
- Smart Card (EMV) and Swipe with Language Selection

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.

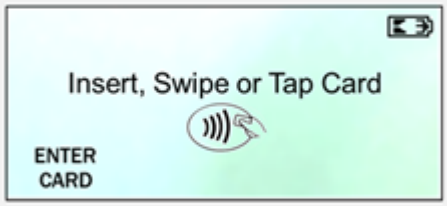
A contactless smart card features an antenna embedded in the card. When the card is tapped, the RF field generated by the contactless card reader powers the embedded chip and enables communications with the terminal.

5_1_16_1 Contactless Smart Card (EMV) and Swipe

iPP320	**iPP350 • iWL250**
	

iSC250	**iSC350**
	

iSC480	iSMP • iSMPc

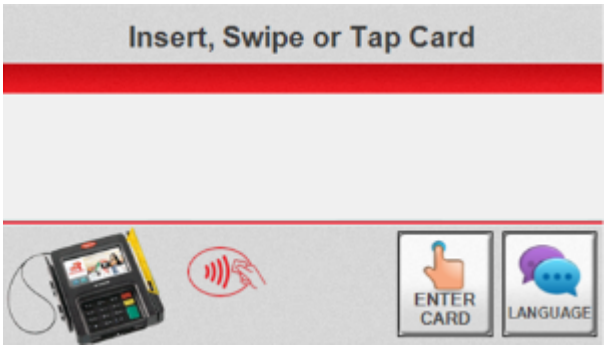
iCMP


<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:

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 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>“Insert, Swipe or Tap Card”</td></tr>
<tr><td>Form Buttons and IDs</td><td>“Enter Card” – M</td></tr>
</table>

5_1_16_2 Contactless Smart Card (EMV) and Swipe with Language Selection

iPP320	**iPP350 • iWL250**

iSC250	**iSC350**
	

iSC480	iSMP • iSMPc

iCMP

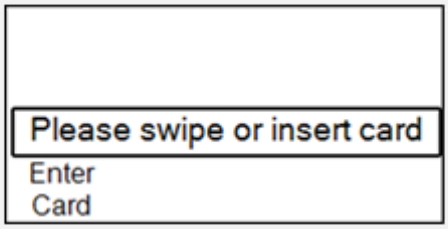
<table>
<tr><td>Description</td><td>Contactless EMV and Swipe with Language. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0019_0001</td><td>1</td><td>Enable EMV transactions.</td></tr>
<tr><td>0008_0001</td><td>1</td><td>Enable contactless.</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

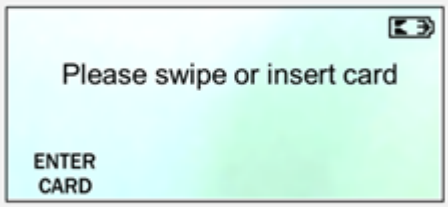
5_1_16_3 Smart Card (EMV) and Swipe

iPP320	iPP350 • iWL250
	

iSC250	iSC350

iSC480	iSMP • iSMPc

iCMP

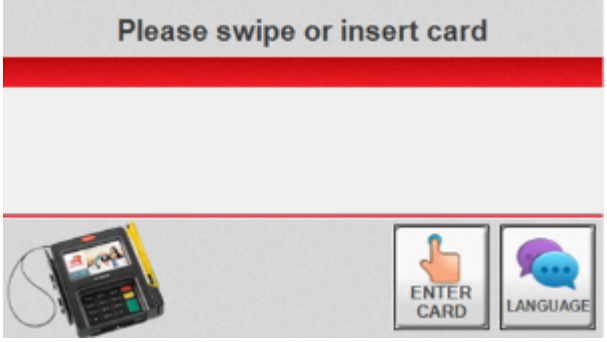
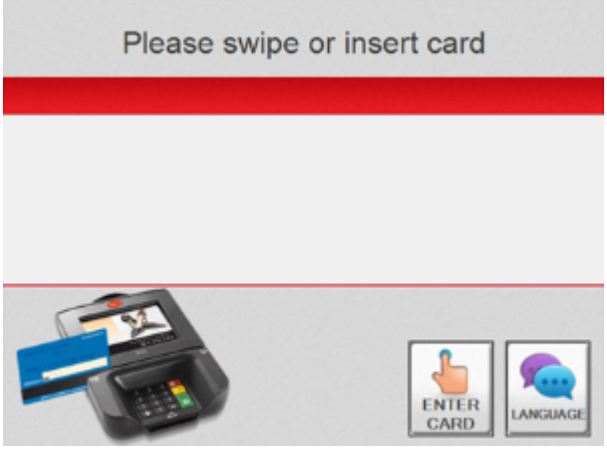


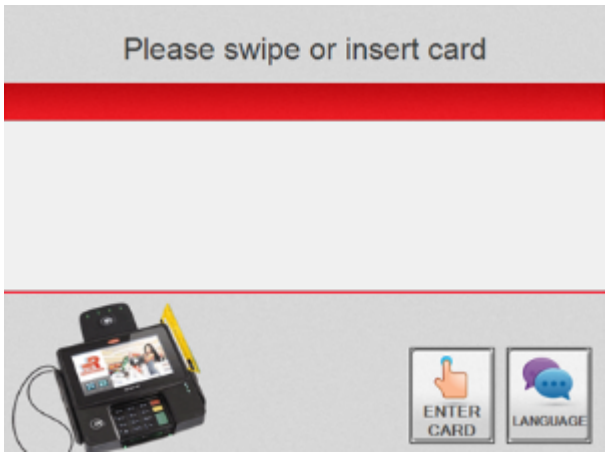
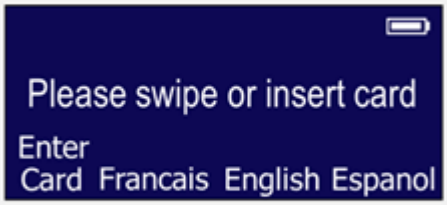
<table>
<tr><td>Description</td><td>EMV card and swipe. Settings are as follows:
<table>
<tr><th>Parameter</th><th>Setting</th><th>Description</th></tr>
<tr><td>0019_0001</td><td>1</td><td>Enable EMV transactions.</td></tr>
<tr><td>0007_0005</td><td>0</td><td>Swipe card first, then select payment type.</td></tr>
<tr><td>0007_0029</td><td>1</td><td>Display "Enter Card" button and prompt for card number, expiration date and CVV.

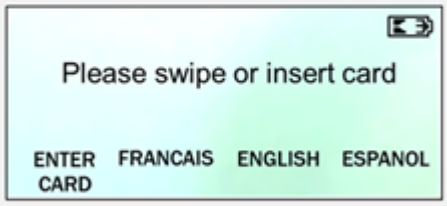
This parameter controls the display of the "Enter Card" button on the card swipe screen and enables cardholder prompts for manual entry. When not set to '0', the "Enter Card" button will be displayed with prompt options. Additional settings for parameter '0007_0029' used during manual entry include:
• 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>
</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>
<tr><td>Form Buttons and IDs</td><td>“Enter Card” – M</td></tr>
<tr><td>Terminal Buttons Allowed</td><td>Cancel</td></tr>
</table>

5_1_16_4 Smart Card (EMV) and Swipe with Language Selection

iPP320	iPP350 • iWL250

iSC250	iSC350
	

iSC480	iSMP • iSMPc
	

iCMP


<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>

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_0037</td></tr>
<tr><td>Form</td><td>ELSWIPE.K3Z</td></tr>
<tr><td>Text</td><td>“Please swipe or inert card"</td></tr>
<tr><td>Form Buttons and IDs</td><td>iPP320/iPP350/iWL250

“Enter Card” – M
"Francais"
"English"
"Espanol"

iSC250/iSC350/iSC480

"ENTER CARD"
"LANGUAGE"
</td></tr>
</table>

Terminal Buttons Allowed	Cancel

5_1_17 Survey Swipe

iSC250 • iSC350 • iSC480

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

Info

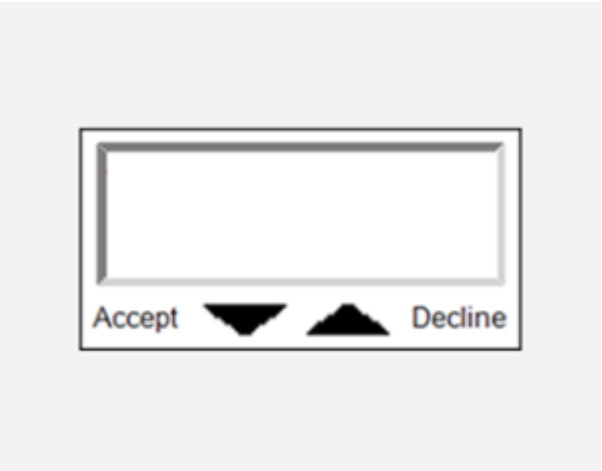
For additional information including status responses on the 40.x and '40.0' messages, see section 40.x: Survey Messages.

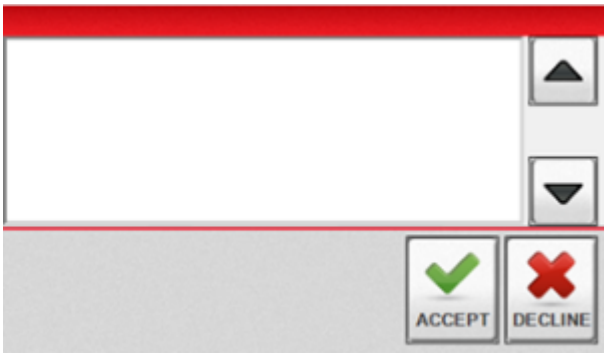
5_1_18 Terms and Conditions Forms

This section includes the following forms:

- Terms and Conditions
- Terms and Conditions Signature

5_1_18_1 Terms and Conditions

iPP320	iPP350 • iWL250
	

iSC250 • iSC350 • iSC480	iSMP • iSMPc
	

iCMP

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

5_1_18_2 Terms and Conditions Signature

iSC250 • iSC350 • iSC480

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

5_2 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, and iCMP terminals.

iPP320 and iPP350 Menu Option Selection

iSMP and iCMP 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.

5_3 Button IDs and Images

Text for dynamically generated buttons is specified in the `PROMPT.XML` file in the prompts directory. There is a separate `PROMPT.XML` for each terminal covered in this manual.

Some customers would like to add Coupons.com support with the RBA application. This involves displaying a button on the main screen which could trigger a sign-in/sign-up process. To support this, the RBA will return button IDs which are not recognized in order to enable the sign-up/sign-in process. This is implemented using the 24.x: Form Entry Request (On-Demand) message. During the normal application flow, any button which is not handled by the application will result in a 24.x message being sent to the POS. The message format will be the same as that of the response message sent when a button is pressed in response to a form display request. This is not supported by On-Demand messages. On-Demand functions have their own response to return key presses.

When setting a custom button ID in the Telium Form Builder, there are three options for the value type:

- hexadecimal
- character

- decimal

For Retail Base Applications, only the character type should be used. Refer to the Form Contents and Descriptions section for a description and images of forms used in Retail Base Applications.

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.

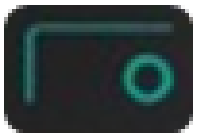
Refer to the following sections for buttons and button IDs specific to each terminal:

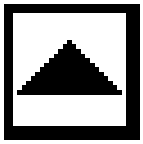
- iSMP and iSMPc Button IDs and Images
- iPP320 Button IDs and Images
- iPP350 Button IDs and Images
- iSC250 Button IDs and Images
- iSC350 Button IDs and Images
- iSC480 Button IDs and Images

Also refer to Mobile Terminal Battery Level Icons.

5_3_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

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	
Dynamically generated text	Store	117	69
Dynamically generated text	Yes	118	Y
	CANCEL		0x1B
	CLEAR		N/A
	Down		N/A
	ENTER		0x0D [CR]
	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

5_3_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

Button Images	Button Name	Button ID	Key Press Value
	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
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

Button Images	Button Name	Button ID	Key Press Value
	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

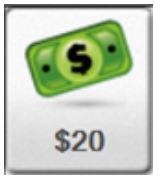
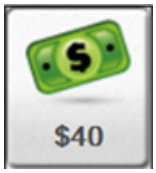
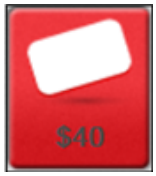
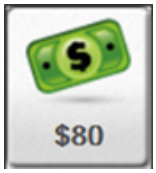
5_3_3 iPP350 Button IDs and Images

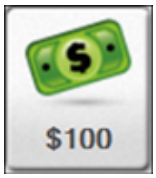
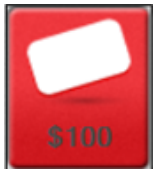
Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	20 (cash back amt 1)		1
Dynamically generated text	40 (cash back amt 2)		2
Dynamically generated text	80 (cash back amt 3)		3
Dynamically generated text	Accept	101	Y
	CANCEL		0x1B
	CLEAR		N/A
Dynamically generated text	Clear	103	0x08 [ESC]

Button Image	Button Name	Button ID	Key Press Value
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
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

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
Dynamically generated text	Store	117	69
	Up		N/A
Dynamically generated text	Yes	118	Y

5_3_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
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
Not Selected Selected	EBT Cash	107	67

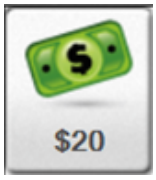
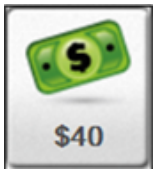
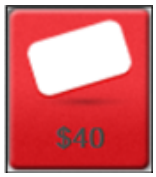
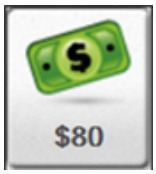
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	EBT Food	108	68
Not Selected Selected	English	109	1
	ENTER (*physical*)		0x0D [CR]
Not Selected Selected	Enter Card	110	M

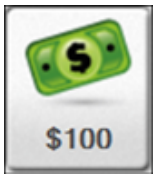
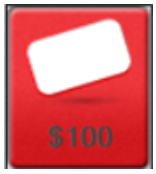
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	Español	111	2
Not Selected Selected	Français	112	3
Not Selected Selected	LANGUAGE	177	L
Not Selected Selected	No	113	N

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated	Ok	114	0x0D [CR]
Not Selected Selected	Other	115	O
Not Selected Selected	PARTIAL PAYMENT	116	
Not Selected Selected	btnp		P
Not Selected Selected	Scroll Down		N/A

Button Image	Button Name	Button ID	Key Press Value
	Scroll Thumb		N/A
Not Selected Selected	Scroll Up		N/A
Not Selected Selected	Store	117	69
Not Selected Selected	Yes	118	Y

5_3_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
 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
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
Not Selected Selected	EBT Cash	107	67

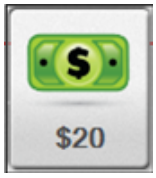
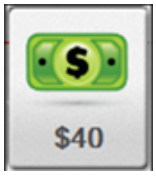
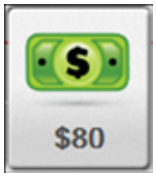
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	EBT Food	108	68
Not Selected Selected	English	109	1
	ENTER (*physical*)	114	0x0D [CR]
Not Selected Selected	Enter Card	110	M

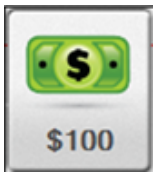
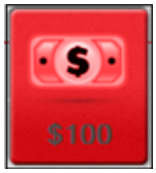
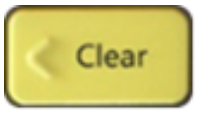
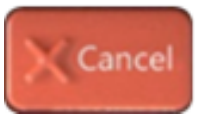
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	Español	111	2
Not Selected Selected	Français	112	3
Not Selected Selected	LANGUAGE	177	L
Not Selected Selected	No	113	N

Button Image	Button Name	Button ID	Key Press Value
Dynamically generated	Ok	167	0x0D [CR]
Not Selected Selected	Other	115	O
Not Selected Selected	PARTIAL PAYMENT	116	
Not Selected Selected	btnp		P
Not Selected Selected	Scroll Down		N/A

Button Image	Button Name	Button ID	Key Press Value
	Scroll Thumb		N/A
Not Selected Selected	Scroll Up		N/A
Not Selected Selected	Store	117	69
Not Selected Selected	Yes	118	Y

5_3_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
 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
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
Not Selected Selected	EBT Cash	107	67

Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	EBT Food	108	68
Not Selected Selected	English	109	1
	ENTER (*physical*)		0x0D [CR]
Not Selected Selected	Enter Card	110	M

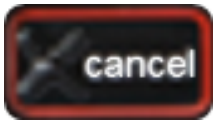
Button Image	Button Name	Button ID	Key Press Value
Not Selected Selected	Español	111	2
Not Selected Selected	Français	112	3
Not Selected Selected	LANGUAGE	177	L
Not Selected Selected	No	113	N

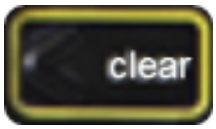
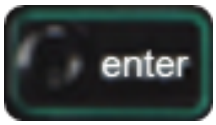
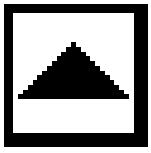
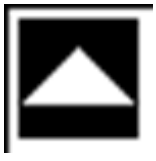
Button Image	Button Name	Button ID	Key Press Value
Dynamically generated	Ok	114	0x0D [CR]
Not Selected Selected	Other	115	O
Not Selected Selected	PARTIAL PAYMENT	116	
Not Selected Selected	btnp		P
Not Selected Selected	Scroll Down		N/A

Button Image	Button Name	Button ID	Key Press Value
	Scroll Thumb		N/A
Not Selected Selected	Scroll Up		N/A
Not Selected Selected	Store	117	69
Not Selected Selected	Yes	118	Y

5_3_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

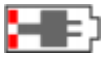
Button Image	Button Name	Button ID	Key Press Value
Dynamically generated text	40 (cash back amt 2)	147	2
Dynamically generated text	80 (cash back amt 3)	155	3
Dynamically generated text	Accept	101	Y
Dynamically generated text	Clear	103	0x08 [ESC]
Dynamically generated text	Credit	104	66
Dynamically generated 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
	Down		N/A
	ENTER		0x0D [CR]
	FUNCTION		
Not Selected Selected	Scroll Down		N/A
 Not Selected  Selected	Scroll Up		N/A
	Up		N/A

5_3_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
	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.

5_3_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

File Names	BmpButton IDs
AMTV.K3Z	btnc, btnn, btnp, btny
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

File Names	BmpButton IDs
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

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 "`<?ivPROMPT3?>`". 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.

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
 	Line break.	`Please wait Do not remove card`	Please wait Do not remove card

Any extra forward slashes (\) will appear in a prompt's displayed text.

5_4_2 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."

Info

French language support is slated for an upcoming RBA release. RBA currently supports English and Spanish only.

5_4_3 Security Prompts (SECURPROMPT.xml)

Prompts for various Ingenico terminal applications are found in the `SECURPROMPT.xml` file. The `SECURPROMPT.xml` file includes prompts in English, Spanish, and French. The `SECURPROMPT.xml` file MUST be signed by Ingenico before being loaded on an Ingenico terminal. Each prompts XML file is subdivided into sections containing several different types of prompts. All of the following security prompts are available for use in the RBA application. The following table provides the security prompts and associated IDs.

Security Prompts

Prompt ID	Default Value	Description
0	PROMPTINDEX	Prompt ID 0 (zero) indicates that the prompt number should be taken from the PROMPTINDEX variable.
PIN Prompts		
14	• "Please enter your PIN:" • "Por favor marque su PIN:" • "Entrez votre PIN SVP:"	This prompt may be used during PIN entry.
15	• "Invalid PIN. Please re-enter" • "PIN incorrecto. Favor de marcar de nuevo" • "PIN erroné. Réessayez"	This prompt may be used during PIN entry.
16	• "Please re-enter" • "Presione ENTER"	This prompt may be used during PIN entry.
17	• "Enter PIN or Press Green for Credit"	This prompt may be used during PIN entry.
Other PIN Prompts		
21	• "ENTER PIN" • "MARQUE PIN" • "ENTRER PIN"	This prompt may be used during PIN entry.
22	• "ENTER PIN & ENTER" • "MARQUE PIN & ENTER" • "ENTRER PIN & ENTRER"	This prompt may be used during PIN entry.
23	• "ENTER PIN & PRESS ENTER" • "MARQUE PIN & OPRIMA ENTER" • "ENTRER PIN & PRESSE ENTRER"	This prompt may be used during PIN entry.
24	• "ENTER PIN AND PRESS ENTER" • "MARQUE PIN Y OPRIMA ENTER" • "ENTRER PIN ET APPUYEZ SUR ENTRER"	This prompt may be used during PIN entry.

Prompt ID	Default Value	Description
25	• "REENTER PIN" • "VUELVA A MARCAR PIN" • "RETOURNER PIN"	This prompt may be used during PIN entry.
26	• "REENTER PIN & ENTER" • "VUELVA A MARCAR PIN & ENTER" • "RETOURNER PIN & ENTRER"	This prompt may be used during PIN entry.
27	• "REENTER PIN & PRESS ENTER" • "VUELVA A MARCAR PIN & OPRIMA ENTER" • "RETOURNER PIN & PRESSE ENTRER"	This prompt may be used during PIN entry.
28	• "REENTER PIN AND PRESS ENTER" • "VUELVA A MARCAR PIN Y OPRIMA ENTER" • "RETOURNER PIN ET APPUYEZ SUR ENTRER"	This prompt may be used during PIN entry.
29	• "PIN=" • "PIN=" • "PIN="	This prompt may be used during PIN entry.
30	• "Please Enter Your PIN and Press ' Enter'" • "Por Favor Marque Su PIN y Oprima ' Enter' " • "SVP entrer votre code PIN et appuyez sur 'Enter'"	This prompt may be used during PIN entry.
31	• "Enter PIN" • "Introducir el PIN" • "Entrez le code PIN"	This prompt may be used during PIN entry.
Clear Entry Prompts		

Prompt ID	Default Value	Description
1	• "Enter home phone number" • "Marque su número de teléfono" • "Entrez le numéro de téléphone à la maison"	This prompt may be used during user input entry.
16	• "Please enter Cashback:" • Por favor marque Cashback: " • "Entrez montant du retrait d'argent SVP:"	This prompt may be used during user input entry.
19	• "Please enter new amount:" • "Por favor marque la cantidad nueva: " • "Entrez le nouveau montant:"	This prompt may be used during user input entry.
39	• "Enter Cashback in $<?ivCB_INC?> increments" • "Marque Cashback en incrementos de $<?ivCB_INC?>" • "Retrait d'argent: multiples de $<?ivCB_INCS?>"	This prompt may be used during user inputentry.
151	• "Enter Card Number" • "Marque Numero de Tarjeta" • "Entrer le Nombre de Carte"	This prompt may be used during user input entry.
153	• "Enter Expiration Date" • "Marque Fecha de Expiración" • "Entrer la Date d'Expiration"	This prompt may be used during user input entry.
155	• "Enter CVV or CID from card" • "Marque CVV o CID de su tarjeta" • "Entrer le Code de Sécurité"	This prompt may be used during user input entry.
Other Clear Entry Prompts		
201	• "Enter Month and Day of Birth (MMDD)" • "Marque Mes y Día de Nacimiento (MMDD) " • "Entrez le mois et le jour de naissance (MMDD)"	This prompt may be used during user input entry.

Prompt ID	Default Value	Description
202	• "Enter Home Phone Number" • "Marque teléfono de su Hogar" • "Entrez le numéro de téléphone à domicile"	This prompt may be used during user input entry.
203	• "Enter Last 4 Digits of Social Security #" • "Marque Últimos 4 Dígitos de Seguro Social" • "Entrez 4 derniers chiffres de la SSN"	This prompt may be used during user input entry.
204	• "Annual Income (from all sources)" • "Ingreso Anual (de todas fuentes) " • "Revenu annuel (toutes source confondues)"	This prompt may be used during user input entry.
205	• "Enter Social Security #" • "Marque # de Seguro Social" • "Entrez la SSN"	This prompt may be used during user input entry.
206	• "Please enter your 10-digit HOME phone number" • "Por favor marque el numero de teléfono de 10-dígitos de SU HOGAR" • "Entrer votre numéro de téléphone"	This prompt may be used during user input entry.
207	• "Social Security Number" • "Numero de Seguro Social" • "Numéro de Sécurité Sociale"	This prompt may be used during user input entry.
208	• "Annual Income Amount" • "Cantidad de Ingreso Anual" • "Montant du revenu annuel"	This prompt may be used during user input entry.
209	• "Years lived at Current Address" • "Años Vividos en la Dirección Actual" • "Années vécues à l'adresse actuelle"	This prompt may be used during user input entry.

Prompt ID	Default Value	Description
210	• "Months lived at Current Address" • "Meses Vividos en la Dirección Actual" • "Mois vécu à l'adresse actuelle"	This prompt may be used during user input entry.
211	• "Mothers Maiden Name" • "Apellido Materno" • "Mères Nom de jeune fille"	This prompt may be used during user input entry.
212	• "Email Address" • "Dirección de Correo Electrónico" • "Adresse e-mail"	This prompt may be used during user input entry.
213	• "Date of Birth" • "Fecha de Nacimiento" • "Date de naissance"	This prompt may be used during user input entry.
214	• "Home Phone Number" • "Numero de teléfono" • "Numéro de téléphone résidentiel"	This prompt may be used during user input entry.
215	• "Zip Code" • "Código Postal" • "Code postal"	This prompt may be used during user input entry.
216	• "Use the attached pen to type mother's maiden name" • "Use el lápiz para teclar su apellido materno" • "Utilisez le stylo attaché aux mères de type nom de jeune fille"	This prompt may be used during user input entry.
217	• "Please enter your annual household income" • "Por favor marque su cantidad de ingreso anual" • "S'il vous plaît entrer votre revenu annuel du ménage"	This prompt may be used during user input entry.

Prompt ID	Default Value	Description
218	• "Please enter your date of birth" • "Por favor marque su fecha de nacimiento" • "S'il vous plaît entrer votre date de naissance"	This prompt may be used during user input entry.
219	• "Please enter your social security number" • "Por favor marque su numero de seguro social" • "S'il vous plaît entrer votre numéro de sécurité sociale"	This prompt may be used during user input entry.
220	• "Enter two-digit number of months" • "Marque numero de meses de dos dígitos" • "Entrez le numéro à deux chiffres du mois"	This prompt may be used during user input entry.
221	• "Enter two-digit number of years" • "Marque numero de años de dos dígitos" • "Entrez le numéro à deux chiffres des années"	This prompt may be used during user input entry.
222	• "Please enter your home phone number" • "Por favor marque su numero de teléfono" • "S'il vous plaît entrer votre numéro de téléphone à la maison"	This prompt may be used during user input entry.
224	"Enter CVN"	This prompt may be used during user input entry.
225	"Please Enter the Check Amount"	This prompt may be used during user input entry.
226	“Please Enter the Check Number”	This prompt may be used during user input entry.
227	“Please Enter Billing Zip Code”	This prompt may be used during user input entry.
228	“Please Enter CID”	This prompt may be used during user input entry.
229	“Please Enter CID Code”	This prompt may be used during user input entry.
230	“Please Enter CVV2”	This prompt may be used during user input entry.

Prompt ID	Default Value	Description
231	“Please Enter CVV2 Code”	This prompt may be used during user input entry.
232	“Please Enter Customer Identification Code”	This prompt may be used during user input entry.
233	“Please Enter Security Code”	This prompt may be used during user input entry.
234	“Please Enter the Last Four digits of your Card Number”	This prompt may be used during user input entry.
235	“Please Enter Invoice Number”	This prompt may be used during user input entry.
236	“Please Enter Driver ID”	This prompt may be used during user input entry.
237	“Please Enter Vehicle Number”	This prompt may be used during user input entry.
238	“Please Enter Vehicle ID”	This prompt may be used during user input entry.
239	“Please Enter Odometer”	This prompt may be used during user input entry.
240	“Please Enter Prompt Code”	This prompt may be used during user input entry.
241	“Please Enter Vehicle Card Number”	This prompt may be used during user input entry.
242	“Please Enter ID Number”	This prompt may be used during user input entry.
234	“Please Enter Reference Number”	This prompt may be used during user input entry.
244	“Please Enter Approval Code”	This prompt may be used during user input entry.
245	“Please Enter Customer Code”	This prompt may be used during user input entry.
246	“Please Enter EBT Voucher Number”	This prompt may be used during user input entry.
247	“Is this a Cash Benefit Card”	This prompt may be used during user input entry.
248	“Enter PD Seq # (Vehicle Number, 5 digits)”	This prompt may be used during user input entry.
249	• “Please Enter ZIP Code” • “Por favor marque Código Postal” • “Entrez votre Code postal”	This prompt may be used during user input entry.

Prompt ID	Default Value	Description
250	• "Enter mobile number or swipe card" • "Ingresar #móvil o deslizar la tarjeta" • "Entrez n° mobile ou glissez carte"	This prompt may be used during user input entry.
251	• "Enter ZIP code" • "Introducir código postal" • "Entrer un code postal"	This prompt may be used during user input entry.

5_4_3_1 Button Text

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

5_4_4 Transaction Prompts (PROMPT.xml)

Prompts and other display strings are also found in the `PROMPT.xml` file. The 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 the display of prompts containing character combinations representing the words “PIN” and “NIP.” The following table provides descriptions and default values for prompts.

Prompt Descriptions and Default Values

Prompt ID	Default Value	Description
0	“PROMPTINDEX”	Prompt ID 0 (zero) indicates that the prompt number should be taken from the PROMPTINDEX variable.
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 may be displayed at the beginning of each transaction. A button for each available language will appear on the screen for customer selection.
2	• "Processing... please wait" • "Procesando… favor de esperar" • "En traitement... Un moment SVP"	The Processing BIN Lookup message displays while the application is using the BIN lookup function to select the payment type for the swiped card.

Prompt ID	Default Value	Description
3	• "Please slide card" • "Por favor deslice su tarjeta" • "Glissez la carte SVP"	This parameter prompts the customer to swipe a payment card.
4	• "Expired card. Please use another." • "Tarjeta expirada. Favor de usar otra. " • "Carte expirée"	This parameter displays when date checking is enabled and the customer used a card with an expiration date that is before today's date. Another form of payment must be submitted in order to complete the transaction.
5	• "Card read error. Try again." • "Error de lectura de tarjeta. Intente de nuevo." • "Erreur de lecture. Réessayez"	The Bad Card Read message informs the customer that terminal could not read the card that was swiped.
6	• "Please select payment type" • "Seleccione tipo de pago" • "Choisissez le type de paiement"	This parameter prompts the customer to select the desired payment type.
7	• "Please wait for the cashier" • "Por favor espere por el cajero (a)" • "Attendez le cassier SVP"	The Wait for Amount Message is displayed when waiting for the purchase amount from the POS.
8	• "Cashback correct? $<? ivCASHBACK?>" • "Cashback correcto? $<? ivCASHBACK?> " • "Retrait d'argent? $<? ivCASHBACK?>"	This prompts the customer to verify the cash back amount.
9	• "Not a sale. Cashback cancelled." • "No es una venta. Cashback cancelado." • "Retrait d'argent annulé"	This informs the customer that the selected cash back amount will be ignored due to a transaction type limitation. This parameter only applies to void, return, and void return transactions.

Prompt ID	Default Value	Description
10	• "Amount OK? $<?ivTOTAL?>" • "Cantidad correcta? $<?ivTOTAL?> • "Montant OK? $<?ivTOTAL?>"	This prompts the customer to verify the purchase amount.
11	• "Processing... please wait" • "Procesando… favor de esperar" • "En traitement... Un moment SVP"	This displays 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"	The Invalid Card prompt informs the customer 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"	This informs the customer that the PIN entered was not either less than 4 or more than 12 digits.
15	• "Cashback?" • "Cashback?" • "Retrait d'argent?"	This asks the customer if he would like to receive cash back.
17	• "Cashback limit is $<?ivCB_MAX_TEXT?>" • "Limite de cashback es $<?ivCB_MAX_TEXT?>" • "Retrait d'argent limité à $<?ivCB_MAX_TEXT?>"	This informs the customer that the amount requested is greater than the maximum allowed.

Prompt ID	Default Value	Description
18	• "Invalid amount: $<? ivCASHBACK?>" • "Cantidad no permitida: $<? ivCASHBACK?>" • "Montant erroné: $<? ivCASHBACK?>"	This informs the customer that the amount requested is not a multiple of the cash back increment amount ('0002_0009').
20	• "Amount must be less than $<? ivAMOUNT? >" • "Cantidad debe ser menos que $<?ivAMOUNT?>" • "Montant dépasse le total $<? ivAMOUNT?>"	This is an error message that displays if the customer enters a payment amount greater than the purchase amount.
21	• "Approved" • "Approbation" • "Approuvé"	This informs the customer that the transaction has been approved. If the authorization message includes a prompt, the prompt from the authorization message is used.
22	• "Declined" • "Denegado" • "Refusé"	This informs the customer that the transaction has been declined. If the authorization message includes a prompt, the prompt from the authorization message is used.
23	• "Transaction cancelled" • "Transacción cancelada" • "Transaction Annulée"	This prompt informs the customer that the transaction has been cancelled.
24	• "Invalid payment type" • "Tipo de pago no permitido" • "Le type nul de Paiement"	This prompt informs the customer that they attempted to use an invalid payment type.
25	• "Card not accepted" • "Tarjeta no aceptada" • "Cette carte n'est pas acceptée"	This prompt informs the customer that the card was not accepted.
26	• "Encrypting... please wait..." • "Codificando… favor de esperar…" • "Chiffrement... patientez SVP..."	This is displayed while a PIN is being encrypted after entry.

Prompt ID	Default Value	Description
27	• "CPEM test card read" • "Lectura de tarjeta de prueba CPEM" • "Lecture de la carte test CPEM"	This prompt 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"	This prompt is displayed when the cash register cancels the card swipe on demand.
33	• "Input cancelled" • "Entrada cancelada" • "Entrée annulée"	This informs the customer that the POS has cancelled the request to read a card.
34	• "Signature cancelled" • "Firma cancelada" • "Signature annulée"	This prompt 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"	This asks the customer to hand the payment card to the cashier so the signature or account number may be verified.
36	• "Void OK? $<?ivTOTAL? >" • "Confirma anulación? $<? ivTOTAL?> " • "Annulation OK? $<? ivTOTAL?>"	This asks the customer to verify the void amount.

Prompt ID	Default Value	Description
37	• "Return OK? $<?ivTOTAL? >" • "Confirma devolución? $<? ivTOTAL?> " • "Remboursement OK? $<? ivTOTAL?>"	This asks the customer to verify the return amount.
38	• "Void return OK? $<? ivTOTAL?>" • "Confirma anulación de devolución? $<?ivTOTAL? > " • "Annuler remboursement OK? $<?ivTOTAL?>"	This asks the customer 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…"	This prompt displays 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 parameter is enabled).
91	• "Unable to authorize" • "Incapaz de autorizar" • "Incapable d'autorise"	This prompt is displayed if the terminal is unable to send an authorization request to the POS.
92	• "Signature accepted" • "Firma aceptada" • "Signature acceptée"	This prompt informs the customer that the signature was accepted.

Prompt ID	Default Value	Description
93	• "Input accepted" • "Entrada aceptada" • "Entrée acceptée"	This informs the customer that the extra input just entered was accepted.
94	• "Card accepted" • "Tarjeta aceptada" • "Carte acceptée"	This prompt informs the customer that the card swipe was accepted.
95	• “Terms accepted” • "Condiciones aceptadas" • "Conditions acceptées"	This prompt confirms that the customer has accepted terms and conditions.
96	• “Terms declined” • "Condiciones negadas" • "Conditions refusées"	This prompt confirms that the customer has declined terms and conditions.
97	• “- More -“ • "- Mas -" • “- Plus -“	This prompt is displayed to indicate there is more available.
98	• “Card read error” • "Error de lectura de tarjeta" • “Erreur de lecture”	This prompt is displayed when a card read error occurs.
100	• "Please insert card" • "Por favor, inserte la tarjeta" • "Insérer la carte SVP"	
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"	

Prompt ID	Default Value	Description
103	• "Please remove card quickly" • "Por favor, retire la tarjeta de forma rápida" • "Retirer la carte SVP"	
120	• "Accept" • "Aceptar" • "Approuvé"	This prompt is displayed to confirm the customer's acceptance.
121	• "Decline" • "Denegar" • "Refusé"	This prompt is displayed to confirm the customer's denial.
126	• "Please hand card to cashier" • "Por favor pase tarjeta al cajero (a)" • "Donnez carte au cassier SVP"	This prompt asks the customer to hand the payment card to the cashier. The cashier may then enter the card number manually. It 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', msr.dat file).
127	• "Too many PIN entry errors" • "Demasiados errores de entrada de PIN" • "Trop d'essais - NIP erroné"	This informs the customer that the transaction is being cancelled because the customer is having trouble entering a valid PIN.
128	• "Invalid PIN. Please re-enter" • "PIN incorrecto. Marque de nuevo" • "NIP erroné. Réssayez"	
130	• "Debit" • "Debito" • "Débit"	This prompt is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or customer screen selection.
131	• "Credit" • "Crédito" • "Crédit"	This prompt is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or customer screen selection.

Prompt ID	Default Value	Description
132	• "EBT Cash" • "Efectivo EBT" • "Comptant - EBT"	This prompt is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or customer screen selection.
133	• "EBT Foodstamps" • "Estampillas EBT" • "Bon alimentaire - EBT"	This prompt is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or customer screen selection.
134	• "Store Charge" • "Cargo de la Tienda" • "Carte magasin"	This prompt is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or customer screen selection.
135	• "Thank you for your loyalty" • "Gracias por su lealtad" • "Merci de votre fidélité"	This prompt is displayed to confirm the selected payment type. The payment may be selected through a POS message, BIN range checking, or customer screen selection.
136	• "PayPal" • "PayPal" • "PayPal"	
137	• "EMV" • "EMV" • "EMV"	
153	• "Invalid Account Number" • "Número de cuenta no válido" • "Numéro de compte non valide"	
154	• “Invalid Date” • "Fecha No Permitida" • “Date Nulle”	This prompt is displayed when an invalid date is entered.
155	• "Incorrect Password"	

Prompt ID	Default Value	Description
156	• “Security Code too small” • "Código de Seguridad demasiado pequeño" • "Code de Sécurité trop petit"	This prompt is displayed when the user enters a security code that is too small.
163	• "Rebooting for iOS mode..."	
164	• "Please ask for assistance" • "Por favor pida ayuda" • "Demandez de l'aide SVP"	This prompt displays when the terminal is unable to read the card after the specified number of allowed bad read attempts has been reached (parameter '0003_0001', msr.dat file). The cashier may then 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"	This prompts the customer to enter a signature, using the electronic pen attached to the terminal.
166	• "Please slide card or Tap" • "Por favor deslice o toque su tarjeta" • "Glissez la carte SVP or TAP"	This prompts the customer to slide or tap their card.
167	• "Index key is missing" • "Clave de índice que falta" • "Clé d'index introuvable"	
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"	

Prompt ID	Default Value	Description
170	• "Is this amount OK? $<?ivTOTAL?>" • "¿Importe OK? $<?ivTOTAL?>" • "Montant OK? $<?ivTOTAL?>"	
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"	
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?"	

Prompt ID	Default Value	Description
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"	
181	• "BT Name: <? ivBLUETOOTHDEVICE?> BT Pairing... PIN: <? ivBLUETOOTHPIN?>" • "Nombre BT: <? ivBLUETOOTHDEVICE?> BT Sincronización... PIN: <?ivBLUETOOTHPIN?>" • "BT Nom: <? ivBLUETOOTHDEVICE?> Pairage BT... PIN: <? ivBLUETOOTHPIN?>"	
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
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 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"	
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"	

Prompt ID	Default Value	Description
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"	
207	• "Cancelled by customer" • "Spanish 207" • "Annulée par le client"	
208	• "Card read error please try again" • "Spanish 208" • "Erreur lecture carte\nSVP Réessayer"	
209	• "Invalid card type... ask for assistance" • "Spanish 209" • "Carte Invalide\nSVP demandez de l'assistance"	
210	• "Card not read please ask for assistance" • "Spanish 210" • "Carte non lisible\nSVP demandez de l'assistance"	

Prompt ID	Default Value	Description
211	• "Please select shopping payment type" • "Spanish 211" • "SVP sélectionnez le type de paiement d'achats"	
212	• "Entry timeout Transaction cancelled" • "Spanish 212" • "Dépassement temp entrée;\n Transaction annulée"	
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"	

Prompt ID	Default Value	Description
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"	
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"	

Prompt ID	Default Value	Description
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"	
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"	

Prompt ID	Default Value	Description
284	• "Call your bank" • "Spanish 284" • "Appelez Votre Banque"	
285	• "Purchase" • "Spanish 285" • "Achat"	
286	• "Refund" • "Spanish 286" • "Remboursement"	
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"	

Prompt ID	Default Value	Description
293	• "Authorization send failed" • "Spanish 293" • "Envoie Autorisation Echoué"	
294	• "Authorization response timeout" • "Spanish 294" • "Delai Réponse d'autorisation"	
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"	

Prompt ID	Default Value	Description
301	• "Declined" • "Spanish 301" • "Refusé"	
302	• "Not approved \n Application expired" • "Spanish 302" • "Non Approuvée\nApplication Expirée"	
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"	

Prompt ID	Default Value	Description
309	• "Authorizing... Please wait" • "Spanish 309" • "Autorisation\nPatientez SVP"	
310	• "System Error" • "Spanish 310" • "Erreur du Systeme"	
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"	

Prompt ID	Default Value	Description
351	• "Accept changes?\nEnter or Cancel" • "Aceptar cambiar?\n\nEnter o Cancel" • "Accepter les changement? \nEnter or Cancel"	
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"	

Prompt ID	Default Value	Description
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"	
360	• "Application Blocked" • "Spanish 360" • "Application Bloquée"	

The following is an example of the initial portion of the default `PROMPT.XML` file.

```
-<PromptDefinition>
 -<Lang1>
    <!-- English -->
   -<NonEntry>
      <Prompt id="0" message="PROMPTINDEX" shortmessage="PROMPTINDEX"/>
      <Prompt id="1" message="Please select language" shortmessage="Please select language"/>
      <Prompt id="2" message="Processing... please wait" shortmessage="Processing... please wait"/>
      <Prompt id="3" message="Please slide card" shortmessage="Please slide card"/>
      <Prompt id="4" message="Expired card. Please use another." shortmessage="Expired card. Use another."/>
      <Prompt id="5" message="Card read error. Try again." shortmessage="Card read error. Try again."/>
      <Prompt id="6" message="Please select payment type" shortmessage="Please select payment type"/>
      <Prompt id="7" message="Please wait for the cashier" shortmessage="Please wait for the cashier"/>
      <Prompt id="8" message="Cashback correct? $<?ivCASHBACK?>" shortmessage="Cashback correct? $<?ivCASHBACK?>"/>
      <Prompt id="9" message="Not a sale. Cashback cancelled." shortmessage="Not a sale. Cashback cancelled."/>
      <Prompt id="10" message="Amount OK? $<?ivTOTAL?>" shortmessage="Amount OK? $<?ivTOTAL?>"/>
      <Prompt id="11" message="Processing... please wait" shortmessage="Processing... please wait"/>
      <Prompt id="12" message="Invalid card for payment type" shortmessage="Invalid card payment type"/>
      <Prompt id="14" message="PIN must be 4 to 12 digits" shortmessage="PIN must be 4 to 12 digits"/>
      <Prompt id="15" message="Cashback?" shortmessage="Cashback?"/>
      <Prompt id="17" message="Cashback limit is $<?ivCB_MAX_TEXT?>" shortmessage="Cashback limit is $<?ivCB_MAX_TEXT?>"/>
      <Prompt id="18" message="Invalid amount: $<?ivCASHBACK?>" shortmessage="Invalid amount: $<?ivCASHBACK?>"/>
      <Prompt id="20" message="Amount must be less than $<?ivAMOUNT?>" shortmessage="Amount must be less than $<?ivAMOUNT?>"/>
```

The default PROMPT.xml file

prompt.xml File

5_5 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 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.

French language support is slated for an upcoming RBA release. RBA currently supports English and Spanish only.

Prompt Type	Prompt ID	Default Value
Terms and Conditions - English	n/a	"I, the undersigned, certify that the person for whom the prescription was written is eligible for benefits. I have received the prescription listed and authorize the release of all information contained, the prescription to which it corresponds and subsequent claim to all parties concerned. I further certify that this medication is NOT covered by any other insurance plan other than the one I previously indicated and I hereby assign any payment due pursuant to this transaction to the pharmacy shown. I certify that I am the patient or the patient's authorized representative and I have the legal right to obtain patient records. PLEASE NOTE: In the event that your insurance company fails to pay this claim, for any reason, by accepting below you agree to reimburse the pharmacy."
Terms and Conditions - Spanish	n/a	"Yo, el abajo firmante, certifica que la persona a la que la receta fue escrita es elegible para recibir beneficios. He recibido la prescripción y la lista de autorizar la liberación de toda la información contenida, la prescripción a la que corresponde y reclamar con posterioridad a todas las partes interesadas. Además, certifico que este medicamento no está cubierto por ningún otro plan de seguro que no sea el i uno se ha indicado anteriormente y me asigno cualquier pago debido conforme a esta transacción a la farmacia se muestra. Yo certifico que soy el paciente o el representante autorizado del paciente y tengo el derecho legal de obtener registros de los pacientes. tenga en cuenta: en el caso de que su compañía de seguros no paga esta reclamación, por cualquier razón, aceptando tu conformidad a reembolsar a la farmacia."
Terms and Conditions - French	n/a	"Je, soussigné, certifie que la personne pour qui l'ordonnance a été écrit est admissible aux prestations. J'ai reçu la prescription énumérés et d'autoriser la libération de toutes les informations contenues, la prescription à laquelle il correspond et demander ultérieurement à toutes les parties concernées. Je certifie en outre que ce médicament n'est pas couvert par un régime d'assurance autre que celle que j'ai déjà indiqué et je cède tout paiement dû en vertu de cette transaction à la pharmacie montré. Je certifie que je suis le patient ou son représentant autorisé du patient et j'ai le droit d'obtenir les dossiers des patients. S'IL VOUS PLAÎT NOTE: Dans le cas où votre compagnie d'assurance ne paie pas cette revendication, pour une raison quelconque, en acceptant ci-dessous, vous acceptez de rembourser à la pharmacie."

5_6 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 RBA Form Variables. This section covers the variables that can be used to meet those needs.

5_6_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.

5_6_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.

5_6_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.

5_6_2 RBA Form Variables

In RBA there are certain variables that can be displayed on a form. Below is a table of the available variables. These variables are always shown in text preceded by "<?iv" and succeeded by ">", e.g., <?ivIP_ADDR_PORT?>.

Available RBA Form Variables

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.
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.

Variable ID	Description
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.
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).

5_6_2_1 IP Address and Port Display Variable for TCP-IP

IP Address Display Variable

In RBA releases 12.2.1 and later, a feature was added to enable terminals to display their IP address without having to reboot the terminal. Form variable <?ivIP_ADDR_PORT?> was added to display a TCP-IP (Ethernet) enabled. The RBA can be configured such that the "This Lane Closed" form will display the IP, allowing users to connect the POS via TCP-IP. Other default forms can also enable the IP address display using this variable.

Either form will display the address in the format www.xxx.yyy.zzz:ppppp (e.g., 192.168.0.103:12000) as shown below:

Lane Closed Form Displaying IP Address

If the variable is not set as active, that section of the form will simply be blank:

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' />
```

5_7 Using the iSC480 Terminal Screen to Display Contactless Status

5_7_1 Overview

The iSC480 terminal may be configured with an internal contactless reader or external contactless reader. Since there are no built-in LEDs for the internal contactless reader, provisions have been made to emulate these using the terminal display. Just as the external contactless reader has four green LEDs which are illuminated when the contactless card enters the RF field, four green LEDs will be displayed in a similar manner at the top of the terminal screen. To facilitate this new feature, forms used during contactless card detection will be adjusted so that the upper 10 pixels are reserved for displaying these LEDs.

5_7_2 Implementation

Firstly, contactless must be enabled by setting parameter '0008_0001' to '1' for PPSE payment only. Next, the application must identify which type of contactless reader is in place (internal or external). A call to `SystemFioctl(SYS_FIOCTL_WHERE_IS_CLESS, 0)` will return the following:

- '1' indicates iSC480 external contactless reader.
- '2' indicates iSC480 internal contactless reader.

In order to preserve the 10 pixel band at the top of the display, forms will need to be adjusted. Every form element placed relative to these elements must also be shifted down. Some forms will require additional adjustments as described in the following table.

Form Modifications for Simulating Contactless LEDs on Terminal Display

Form	Modification
`ads.K3Z`	• Reduce loop image from 480 pixels to 470 pixels. • Reduce ads by clipping off the top 5 and bottom 5 pixels so that it remains centered.
`alpha.K3Z`	• Top of form must be moved down to y=20 pixels in order to preserve the 10 pixel border.
`alphaNew.K3Z`	• (same as `alpha.K3Z`)
`amtv.K3Z`	• `BG-IngBase1.png` must be reduced from 480 to 470 pixels. • Should be shifted down 10 pixels.
`app.dapp.K3Z`	• (same as `amtv.K3Z`)
`OFFLINEVID.K3Z`	• `AdsVideo` element may be shifted down 5 pixels to preserve centering.
`pin.K3Z`	• Apply modifications for `amtv.K3Z`. • Reduce label PROMPTLINE1 to height of 73 pixels and shift down so that it is not in the 10 pixel band at the top of the display.
`PPOFFLINE.K3Z`	• `PPALAD01.png` may be shifted down 5 pixels to preserve vertical centering.
`presign.K3Z, sign.K3Z, tcsign.K3Z`	• Apply modifications for `amtv.K3Z`. • Shift label PROMPTLINE1 down 10 pixels.
`smcmsg`	• Apply modifications for `amtv.K3Z`. • Reduce label PROMPTLINE6 to height of 63 pixels and shift down to y-20 pixels.
`smcupdate.K3Z`	• Apply modifications for `amtv.K3Z`. • Reduce label PROMPTLINE7 to height of 63 pixels and shift down to y-20 pixels.

A new set of forms and images supporting this feature can be created and saved in the `Parms` sub-directories `iSC480i` and `iSC480ai`. Additionally, cmake files `iSC480i.cmake` and `iSC480ai.cmake` can be added to `mk/WEBCONTENT` for building the new configuration.

6_Drivers and Tools

6_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.

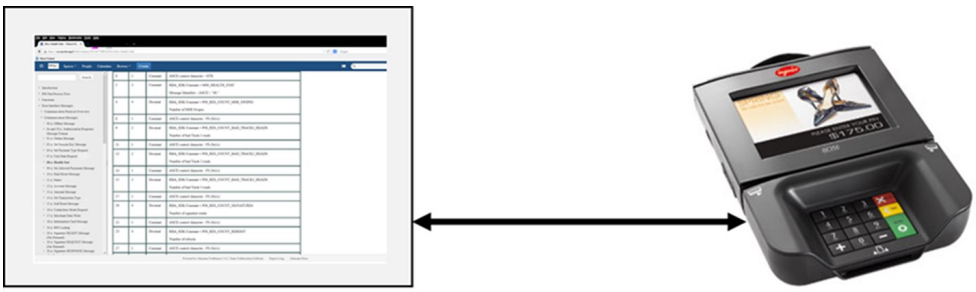
Example iConnectEFT Constants Used in 24.x: Form Entry Request (On-Demand) Message

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.
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.

6_2 Ingenico iConnectREST

6_2_1 Overview of iConnectREST

The Ingenico iConnectREST is a collection of software packages which supports the development of web applications that communicate with Ingenico Telium terminals using the RESTful Application Programming Interface (API). The iConnectREST package enables an application to interface with the RBA application running on a Telium terminal via a REST web service. Refer to the following illustration.



Using iConnectREST to Communicate with Telium Terminal

6_2_2 Supported Payment Terminals and Connection Methods

The following table describes the iConnectREST supported payment 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
	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

6_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 terminal.

Refer to the RBA Testing Tool documentation for more information.

7_Configuring the Application

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

7_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.

7_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.

7_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 */

7_1_3 File Name Line

The file name will contain 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 double quotes, for example, “`cashback.dat`”.
- 5-digit format code number: Consists of a 3-digit data identifier and a 2-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.

The following syntax rules must be followed for file names. The syntax of the file name will be checked by the application. The available characters for file names are as follows:

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

7_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. Here are some examples of the name translations:

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

In config.dfs the parameters are listed by group 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 Ingenico 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 may 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 will be 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 the double quote characters (e.g., “Please enter PIN:”). Comments are allowed in a DFS file. They must use either /* */ or // format.

7_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 5 bytes in length. The revision information is 12 bytes in length.
- The data is delimited with the binary number zero (0x0).

The DAT file should not be edited since it is difficult to determine the meaning of the data therein. Instead, edit the DFS file and run the translator from the DFS to DAT format.

7_2_1 Advertising Parameters (ads.dat)

7_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 RBA or on-demand by the POS.

7_2_1_2 Automatic Startup of Advertising

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

- When the RBA 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 RBA is offline and the '0010_0001' configuration parameter value is not '0', offline advertising is enabled and will continue until terminated by the 01.x: Online Message.
- When the RBA goes to the transaction end and advertising is enabled, advertising will continue until terminated by a 00.x, 01.x, 10.x, '15.0', '15.6', 20.x, 21.x, or 23.x message.

7_2_1_3 On-Demand Startup of Advertising

When the 30.x: Advertising Request Message (On-Demand) is received from the POS, the RBA stops execution of the current process and proceeds with the advertisements. This process is ignored when the RBA is in the offline state or when 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 RBA.

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.JPG 50 1111111 00:00 24:00 101	Advertisement 1.
	0010_0012	AD2.JPG 50 1111111 00:00 24:00 101	Advertisement 2.
	0010_0013	AD3.JPG 50 1111111 00:00 24:00 101	Advertisement 3.
	0010_0014	AD4.JPG 50 1111111 00:00 24:00 101	Advertisement 4.
	0010_0015	AD5.JPG 50 1111111 00:00 24:00 101	Advertisement 5.

Parameter Name	DFS Data Index	Default Value	Description
	0010_0016	AD6.JPG 50 1111111 00:00 24:00 101	Advertisement 6.
	0010_0017	AD7.JPG 50 1111111 00:00 24:00 101	Advertisement 7.
	0010_0018	AD8.JPG 50 1111111 00:00 24:00 101	Advertisement 8.
	0010_0019	AD9.JPG 50 1111111 00:00 24:00 101	Advertisement 9.
	0010_0020	AD10.JPG 50 1111111 00:00 24:00 101	Advertisement 10.
	0010_0021	AD11.JPG 50 1111111 00:00 24:00 101	Advertisement 11.
	0010_0022	AD12.JPG 50 1111111 00:00 24:00 101	Advertisement 12.
	0010_0023	AD13.JPG 50 1111111 00:00 24:00 101	Advertisement 13.
	0010_0024	ADV1.JPG 100 1111111 00:00 24:00 010	Advertisement 14.
	0010_0025	ADV2.JPG 100 1111111 00:00 24:00 010	Advertisement 15.
	0010_0026	ADV3.JPG 100 1111111 00:00 24:00 010	Advertisement 16.

Parameter Name	DFS Data Index	Default Value	Description
	0010_0027	ADV4.JPG 100 1111111 00:00 24:00 010	Advertisement 17.
	0010_0028	ADV5.JPG 100 1111111 00:00 24:00 010	Advertisement 18.
	0010_0029	ADV6.JPG 100 1111111 00:00 24:00 010	Advertisement 19.
	0010_0030	AD20.JPG 50 1111111 00:00 24:00 000	Advertisement 20.
	0010_0031	AD21.JPG 50 1111111 00:00 24:00 000	Advertisement 21.
	0010_0032	AD22.JPG 50 1111111 00:00 24:00 000	Advertisement 22.
	0010_0033	AD23.JPG 50 1111111 00:00 24:00 000	Advertisement 23.
	0010_0034	AD24.JPG 50 1111111 00:00 24:00 000	Advertisement 24.
	0010_0035	AD25.JPG 50 1111111 00:00 24:00 000	Advertisement 25.
	0010_0036	AD26.JPG 50 1111111 00:00 24:00 000	Advertisement 26.
	0010_0037	AD27.JPG 50 1111111 00:00 24:00 000	Advertisement 27.

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

Advertisement durations are only used if advertisements are set to recycle. In order for advertisement scheduling to work, the terminal's 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 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 */

/* |||||||||| ||| |||||| ||||| ||||| ||| */

7_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 in which 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 (currently 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 (currently 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 (currently 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>ON</td><td>OFF</td></tr><tr><td>2</td><td>OFF</td><td>ON</td></tr><tr><td>3</td><td>ON</td><td>ON</td></tr></table></td></tr>
<tr><td>Lighting Mode</td><td>0015_0006</td><td>0</td><td>Lighting mode (currently iSMP only).
0 = Shorter exposure time
1 = Longer exposure time (for shiny or reflective surfaces; see '0015_0004 : Image mode : 4')</td></tr>
</table>

Parameter Name	DFS Data Index	Default Value	Description
Trigger Mode	0015_0007	0	Trigger mode (currently iSMP only) • 0 = Physical trigger disabled • 1 = Physical trigger enabled to scan barcodes
Symbologies	0015_0008	" "	A comma-separated list of (non-negative) decimal codes corresponding to barcode symbologies to enable. Examples: • " " to disable ALL barcode symbologies • "00" to enable ALL barcode symbologies • "13,23,33,41" to enable Code39, Code128, PDF417, and QR barcode s ' ' NO barcode symbologies enabled; all symbologies disabled '0'/'00' may be used as a solitary code to enable all symbologies **List of supported (iSMP) symbologies**:

<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>'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>
<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 set.</td><td>Symbology encryption.
A comma-separated list of (non-negative) decimal codes corresponding to barcode symbologies whose data will be encrypted in 95. messages.
Use same format as '0015_0008 : Symbologies'</td></tr>
</table>

7_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 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
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

Parameter Name	DFS Data Index	Default Value	Description
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

7_2_4 BIN Processing (allBins.dat, bin0.dat - bin20.dat)

This section describes the parameters needed to determine the bank identification number (BIN) processing information. A BIN identifies the account number of the card swiped. BIN range checking is used to determine the card type (e.g., Visa, American Express) using the account number. After a card is swiped, the terminal's internal 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 included in the list, the payment type is retrieved from that list also. Here is a basic look up table.

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	16
VISA	4	13, 16

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

The RBA may also request the payment type from the host. If the payment type is not selected by the host, the customer is prompted to make the selection by pressing a button on the terminal screen.

In the config.dfs file, these parameters are listed under the heading BIN Processing. The filenames in this section are `allbins.dat`, and `bin0.dat` through `bin6.dat`. You may add additional bins, up to `bin12.dat`.
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	This parameter specifies if BIN range checking should be used. • 0 = Disable (default) • 1 = Enable (must be set to 1 if using PayPal) A BIN identifies the account number of the card swiped. BIN range checking is used to determine the card type (e.g., Visa, American Express) using the account number. Because the S1 encryption type requires the use of Mod-10, the Mod-10 flag in each BINx.dat file must be set to a value of '1'.
Number of BIN Ranges	0099_0002	12	This parameter sets the number of BIN files you want to search in config.dfs. When the value is 6, files bin1.dat to bin6.dat are searched. You may add more BIN files, up to bin12. File bin0.dat contains the default values. If this parameter is set to "0", all cards are considered valid and no BIN range parameters are required.
BIN Length is x Digits	0099_0003	6	This parameter specifies the number of digits in the account number to test for. Testing starts from the account first digit. If the number of digits to test is 4, all account digits from the 5th digit to the last account digit are not tested.

Card Transaction Codes (`bin0.dat` - `bin20.dat`)

This section describes the parameters listed under `bin0` to `bin20`.
BIN Ranges

The first part of the DFS Data Index (0100 to 0120) represents a particular card type (e.g., VISA, MasterCard). The second part of the DFS Data Index (0001 to 0004) represents the following:

- 0001 = Start BIN range.
- 0002 = End BIN range.
- 0003 = Minimum account length.
- 0004 = Maximum account length.

The BIN range is the first few digits of the account number used for checking the BIN range. The number of digits used for checking is selected by the '0099_0003' parameter (default is 4 digits). This parameter's value must be lower than the End of BIN Range parameter that has the same first four digits of the DFS data index number. If a swiped card's BIN is within this parameter and the End of BIN Range parameter, the swiped card will be considered valid. The minimum length and maximum length of the account number are used to distinguish between cards that may have similar prefixes but diverse lengths.

The following table lists the DFS Data Index for each card type and provides the default value, Start of BIN range, End of BIN range, minimum account number length and maximum account number length.

BIN Processing Parameters and Descriptions

Default Transaction Codes, bin0.dat

DFS Data Index	Default Value	Description
0100_0001	" "	reserved
0100_0002	" "	reserved
0100_0003	12	Minimum account length.
0100_0004	24	Maximum account length.
0100_0005	000110102030405060708090A0B0C0D0E0F0G01121314151617181 91A1B1C1D1E1F1G11222324252627282 92A2B2C2D2E2F2G213233343536373 8393A3B3C3D3E3F3G3	
0100_0006	" "	

PayPal Discover, bin1.dat

DFS Data Index	Default Value	Description
0101_0001	601104	Start of 1st BIN range. Due to the value of parameter '0040_0008' (PayPal/Discover BIN table number), this BIN table number will be skipped when doing a BIN lookup with PayPal disabled (parameter '0040_0006' is set to '0'). Do not alter 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 bin11 to non-payment cards.
0101_0002	601104	End of 1st BIN range.
0101_0003	14	Minimum account length.
0101_0004	16	Maximum account length.
0101_0005	G0011000000000 000070000000000 000000000000000 000000710000000 000000000000000 000000007200000 000000000000000 000000000073000 000000000000000	
0101_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

Discover 1, bin2.dat

DFS Data Index	Default Value	Description
0102_0001	601100	Start of 1st BIN range.
0102_0002	601199	End of 1st BIN range.
0102_0003	14	Minimum account length.
0102_0004	16	Maximum account length.
0102_0005	000110002000000000000000000000000000000021000000000000000000000000000000022000000000000000000000000000000023000000000000000000000000000000	
0102_0006	MCSHcm	

Discover 2, bin3.dat

DFS Data Index	Default Value	Description
0103_0001	622126	Start of 1st BIN range.
0103_0002	622925	End of 1st BIN range.
0103_0003	14	Minimum account length.
0103_0004	16	Maximum account length.
0103_0005	000110002000000000000000000000000000000021000000000000000000000000000000022000000000000000000000000000000023000000000000000000000000000000	
0103_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

Discover 3, bin4.dat

DFS Data Index	Default Value	Description
0104_0001	644000	Start of 1st BIN range.
0104_0002	649999	End of 1st BIN range.
0104_0003	14	Minimum account length.
0104_0004	16	Maximum account length.
0104_0005	000110002000000000000000000000000000000021000000000000000000000000000000000022000000000000000000000000000000000023000000000000000000000000000000	
0104_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

Discover 4, bin5.dat

DFS Data Index	Default Value	Description
0105_0001	650000	Start of 1st BIN range.
0105_0002	659999	End of 1st BIN range.
0105_0003	14	Minimum account length.
0105_0004	16	Maximum account length.
0105_0005	000110002000000000000000000000000000000021000000000000000000000000000000000022000000000000000000000000000000000023000000000000000000000000000000	
0105_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

MasterCard, 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	000110102000000000000000000000000000011210000000000000000000000000000012220000000000000000000000000000000013230000000000000000000000000000	
0106_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

VISA, bin7.dat

DFS Data Index	Default Value	Description
0107_0001	400000	Start of BIN range.
0107_0002	499999	End of BIN range.
0107_0003	13	Minimum account length.
0107_0004	16	Maximum account length.
0107_0005	000110102000000000000000000000000000011210000000000000000000000000000012220000000000000000000000000000000013230000000000000000000000000000	
0107_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

AMEX - Range 1, bin8.dat

DFS Data Index	Default Value	Description
0108_0001	340000	Start of BIN range.
0108_0002	349999	End of BIN range.
0108_0003	14	Minimum account length.
0108_0004	15	Maximum account length.
0108_0005	000110002000000000000000000000000000000021000000000000000000000000000000000022000000000000000000000000000000000023000000000000000000000000000000	
0108_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

AMEX - Range 2, bin9.dat

DFS Data Index	Default Value	Description
0109_0001	370000	Start of BIN range.
0109_0002	379999	End of BIN range.
0109_0003	14	Minimum account length.
0109_0004	15	Maximum account length.
0109_0005	000110002000000000000000000000000000000021000000000000000000000000000000000022000000000000000000000000000000000023000000000000000000000000000000	
0109_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

Loyalty Cards, bin10.dat

DFS Data Index	Default Value	Description
0110_0001	700100	Start of BIN range.
0110_0002	700199	End of BIN range.
0110_0003	10	Minimum account length.
0110_0004	20	Maximum account length.
0110_0005	000110000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000 00000000000000000000000000000000	
0110_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

PayPal Physical Cards, bin11.dat

DFS Data Index	Default Value	Description
0111_0001	622119	Start of BIN range. Due to the value of parameter '0040_0009' (PayPal/Discover BIN table number), this BIN table number will be skipped when doing a BIN lookup with PayPal disabled (parameter '0040_0006' is set to '0'). Do not alter 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 bin11 to non-payment cards.
0111_0002	622119	End of BIN range.
0111_0003	16	Minimum account length.
0111_0004	16	Maximum account length.
0111_0005	G00110000000000000700000000000000000000000000000071000000000000000000000000000000072000000000000000000000000000000730000000000000000000	
0111_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

All Other Cards, Customer Definable, bin12.dat

DFS Data Index	Default Value	Description
0112_0001	000000	Start of BIN range.
0112_0002	999999	End of BIN range. There are up to 30 BIN ranges. All BIN ranges can be configured by the customer.
0112_0003	10	Minimum account length.
0112_0004	20	Maximum account length.
0112_0005	000110102030405060708090A0B0C0D0E0F0G0112131415161718191A1B1C1D1E1F1G1122232425262728292A2B2C2D2E2F2G213233343536373839 3A3B3C3D3E3F3G	
0112_0006	MCSHcm	Card source. • M = MSR • C = Contactless • S = Smart card • c = Coupon • m = Mobile • H = Hand entry

7_2_4_1

7_2_4_2 Transaction Codes Table

<table>
<tr><th>DFS Data Index</th><th>Description</th></tr>
<tr><td>

- 0100_0005 (default transaction code, `bin0.dat`)
- 0101_0005 (PayPal Discover card, `bin1.dat`)
- 0102_0005 (Discover1, `bin2.dat`)
- 0103_0005 (Discover2, `bin3.dat`)
- 0104_0005 Discover3, `bin4.dat`)
- 0105_0005 (Discover4, `bin5.dat`)
- 0106_0005 (MasterCard, bin6.dat)
- 0107_0005 (Visa, `bin7.dat`)
- 0108_0005 (American Express, Range 1, `bin8.dat`)
- 0109_0005 (American Express, Range 2, `bin9.dat`)
- 0110_0005 (sample loyalty card)
- 0111_0005 (PayPal Physical Card)
- 0112_0005 to 00120_0005 (customer-definable cards, `bin12.dat` to `bin20.dat`)

</td><td>

This parameter contains a list of valid transaction codes for each payment type included in the authorization message. Each transaction code is 2 digits long. See previous tables for each's default value.

This table is used to convert the payment type (e.g., debit) and the transaction type (e.g., void) into the transaction number. Each payment type has a unique transaction number. The string included in this parameter includes the following values:

<table>
<tr><th>Character</th><th></th></tr>
<tr><td>1</td><td>Card type.
Value = A through P.</td></tr>
<tr><td>2</td><td>(Obsolete)</td></tr>
<tr><td>3</td><td>Mod 10 digit checking flag.
0 = Disable.
1 = Enable.
Set to "0" for S1 encryption.</td></tr>
<tr><td>4</td><td>Prompt for expiration date during manual entry.
0 = Disable.
1 = Enable.</td></tr>
<tr><td>5</td><td>Prompt for CVV during manual entry.
0 = Disable.
1 = Enable.</td></tr>
<tr><td>Remaining characters</td><td>Transaction codes string.
If a transaction code is set to 00, the payment type will not be allowed for that BIN range.
If a transaction code is not set to 00, the payment type will be allowed for that BIN range.</td></tr>
</table>

</td></tr>
</table>

7_2_4_3 Card Source

The following table lists the card source types. The default value "MCSHcm" allows for the following card sources:

- M = MSR
- C = Contactless
- S = Smart Card
- c = coupon
- m = Mobile
- H = Hand entry (manual entry)

Available Card Source Types

Card Source	Description
H	Manual (hand) entry.
M	MSR.
C	Contactless (MSR or EMV).
S	Smart card (e.g., EMV, WIC, memory).
A	Account message entry.
c	Coupon or key card.
m	Mobile.
?	Unknown or invalid card type.

Card Source DFS Data Index Table

DFS Data Index	Card Source
0101_0006	PayPal Discover card, `bin1.dat`
0102_0006	Discover1, `bin2.dat`
0103_0006	Discover2, `bin3.dat`
0104_0006	Discover3, `bin4.dat`
0105_0006	Discover4, `bin5.dat`
0106_0006	MasterCard, `bin6.dat`
0107_0006	Visa, `bin7.dat`
0108_0006	American Express, Range 1, `bin8.dat`
0109_0006	American Express, Range 2, `bin9.dat`
0110_0006	Sample loyalty card
0111_0006	PayPal Physical Card
0112_0006	All other cards

7_2_5 Card Configuration (cards.dat)

The Card Configuration parameters are found in the `config.dfs` file under the heading 'Cards' and filename `cards.dat`.

Parameters of this file are used to control data flow in the Retail Base Application (RBA), individually per card type. RBA supports 16 card types, which are called A, B, C, etc., up to P. The order of the card options executed by the RBA are listed in columns, starting from the left side and going to the right. The order may be altered by some of the configuration parameters listed in different configuration files, for example, `mainFlow.dat`.

Parameter Name: Card Configuration for Cards A through P

DFS Data Index: '0011_0001' to '0011_0016'

Default Value: See Card Configuration Table.

Description: Each record specifies the processing of a different card type. Most types are defined as payment cards. These are configured to process appropriately for debit, credit, and other payment types. 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 want 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                            */
            |||                                                  */
'0011_0001' 80E"  /* Card type A (key ID = 65) - Debit          */
'0011_0002' C0E"  /* Card type B (key ID = 66) - Credit         */
'0011_0003' C06"  /* Card type C (key ID = 67) - EBT Cash       */
'0011_0004' C06"  /* Card type D (key ID = 68) - EBT Foodstamps */
'0011_0005' C06"  /* Card type E (key ID = 69) - Store Charge   */
'0011_0006' C06"  /* Card type F (key ID = 70) - Loyalty        */
'0011_0007' C02"  /* Card type G (key ID = 71) - PayPal         */
'0011_0008' 888"  /* 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.

7_2_5_1 Use of the Verify Amount Flag During EMV Transactions

EMV transactions now utilize the Amount Verify flag in the cards.dat section of the config.dfs file as of RBA release version 14.0.x. This provides a means of suppressing the amount verification prompt during the EMV transaction flow. In situations where the customer wishes to prompt the user 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 their 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.

So 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'.

7_2_5_2 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.

The following image is extracted from Card Configuration Table (`config.dfs` file). The two most commonly used combinations, where the hexadecimal representations are either '002' or '406', are outlined, below, as examples.

```
Card sources allowed (hex rep. of bit mask)      */
|||  (0000 0000 0000)                            */
|||   |||| |||| ||||                              */
|||   |||| |||| |||(Reserved)                     */
|||   |||| |||| ||MSR                            */
|||   |||| |||| |Contactless                      */
|||   |||| |||| Contactless EMV (not implemented)*/
|||   |||| |||SmartCard Memory (not implemented) */
|||   |||| ||SmartCard Generic (not implemented) */
|||   |||| |SmartCard WIC (not implemented)      */
|||   |||| SmartCard EMV (not implemented)       */
|||   |||Coupon (not implemented)                 */
|||   ||Mobile (not implemented)                  */
|||   |Hand Entry                                 */
|||   (Reserved)                                  */
|||                                               */
002"   /* Card type A (key ID = 65) - Debit
406"   /* Card type B (key ID = 66) - Credit
```

Extract of Card Sources Allowed from the config.dfs File (Table 71. Card Configuration Table)

Example Binary-to-Hex Conversions

Hex Representation	Binary version	Description
		Binary digits: • 0 = Not allowed. • 1 = Allowed.
002	0000 0000 0010	One card source is allowed: MSR. In the example table image provided, above, MSR is the only card source allowed for use with debit cards `('0011_0001').`
406	0100 0000 0110	Three card sources are allowed: Hand Entry, Contactless and MSR. In the example table image provided, above, these three card sources are allowed for use with credit cards `('0011_0002').`

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
C	1100
D	1101
E	1110
F	1111

7_2_5_3 Card Configuration Table

The columns in the card configuration table are:

Card Configuration Table

Parameter	Position from Left	Description

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.
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.

Parameter	Position from Left	Description
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: Authorization 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.
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. • 0 = Cancel the payment and start over. • 1 - 9 = Loads the payment menu 1 - 9 respectively. • A - P selects the specific payment type.

Parameter	Position from Left	Description
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.
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.

7_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.
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).

Parameter Name	DFS Data Index	Default Value	Description
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.
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

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

7_2_7 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.

7_2_8 Compatibility Flags (compat.dat)

This section is for customers who have coded their registers to work with features that were in the RBA of legacy terminals. It allows the user to select the old behavior or new behavior.

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

compat.dat Parameters

11.x Status Response Format	0013_0001	1	This parameter 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, response with: • 0 = <STX><11.><00><text><ETX> (response with status) • 1 = <STX><00.><offline code><ETX> (response with off-line message)
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 1/10ths 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 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	Sets when to send a reset message at the end of a transaction. Only 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
Reset on Decline	0013_0008	1	Sets the application to reset after a decline transaction: • 0 = Do not reset • 1 = Reset

Parameter Name	DFS Data Index	Default Value	Description
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	This setting controls whether to 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	This setting controls whether to go online after a successful EFT download. • 0 = Start in offline mode • 1 = Start in online mode
Field Size for RAM and Flash Memory Size	0013_0016	1	This setting sets the field size for the terminal's RAM and Flash memory. • 0 = 4-byte fields (compatible with RBA versions prior to 3.0.0) • 1 = Variable size fields (default). 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 transaction. • 0 = Disable • 1 = Enable

Parameter Name	DFS Data Index	Default Value	Description
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

7_2_9 Contactless Reader Configuration (cless.dat)

This section describes the configuration parameters pertaining to the internal contactless card reader.

It is important to note that when contactless is enabled using the '412' variable with the 28.x: Set Variable Request message, contactless mode will only remain enabled until the terminal is rebooted. It will be disabled following a reboot of the terminal. To enable contactless mode permanently (following terminal reboot), use the '0008_0001' configuration parameter (below) with 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). To retain the value 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 table for contactless configuration parameters.

Contactless Configuration Parameters

Parameter Name	DFS Data Index	Default Value	Description

Parameter Name	DFS Data Index	Default Value	Description
Contactless Reader Mode	0008_0001	0	This parameter defines whether the contactless card reader in the terminal is enabled, and which supported mode is selected. • 0 = Disabled. • 1 = Payment (PPSE) only. • 2 = Payment (PPSE) first, Google Wallet second. • 3 = Google Wallet first, payment (PPSE) second. • 4 = Google Wallet only. • 5 = Payment (PPSE) first, Isis SmartTap only. • 6 = Isis SmartTap first, payment (PPSE) second. • 7 = Isis SmartTap only. • 8 = Key Card only. • 9 = EMV only. After the contactless mode setting is changed, the terminal will reset the transaction and the screen will refresh.
Beep When Contactless Card is Read	0008_0003	1	This parameter allows the application to sound a tone when a valid Contactless card read is received. • 0 = Do not sound a tone. • 1 = Sound a tone.
Contactless Both Tracks Indicator	0008_0005	b	This parameter 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	This parameter 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	This parameter 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	This parameter defines the display duration for a "Bad read" error. • 0 = Disabled. • >0 = Display duration in 1/10ths seconds.

Parameter Name	DFS Data Index	Default Value	Description
Contactless Event Delay	0008_0010	7	This parameter 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 1/10ths of a second
Contactless Floor Limit	0008_0011	200	This parameter defines the contactless floor limit.

7_2_10 EMV Flags (emv.dat)

This section describes the EMV flags in the emv.dat file which are used for EMV transaction.

emv.dat Parameters

Parameter Name	DFS Data Index	Default Value	Description
EMV Transactions Supported	0019_0001	0	Determines if the register supports EMV transactions. • 0 = Register does not support EMV transactions. • 1 = Register supports EMV transactions.
Language Auto-Select During EMV Transaction	0019_0002	1	Determines if the terminal should auto-select the language during an EMV transaction. • 0 = Terminal should not auto-select the language during an EMV transaction. • 1 = Terminal should auto-select the language during an EMV transaction.
Application Auto-Select during EMV Transaction	0019_0003	1	Determines if the terminal should auto-select the application during an EMV transaction. • 0 = Terminal should not auto-select the application during an EMV transaction. • 1 = Terminal should auto-select the application during an EMV transaction.
Asynchronous Transmission of Status Message	0019_0004	0	Determines if the terminal should asynchronously transmit the Status message to the register on change of status. • 0 = Terminal should not asynchronously transmit the Status message to the register on change of status. • 1 = Terminal should asynchronously transmit the Status message to the register on change of status.

Parameter Name	DFS Data Index	Default Value	Description
Load EMV Test Environment	0019_0005	1	Determines if the terminal should load the EMV test environment as well as the production environment. • 0 = Terminal should only load the production environment. • 1 = Terminal should load the test environment as well as the production environment.
Wait After Authorization Confirmation Sent to Register	0019_0006	0	Determines if the terminal should wait after the Authorization Confirmation message has been sent to the register. Typically, the terminal pauses for 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 (this is as originally implemented in the CA implementation). • 1 = Wait after the confirmation message is assumed in the EMV transaction.
Interac Application Selection	0019_0007	1	Determines if the Interac application selection is enabled. • 0 = Interac application selection is disabled. • 1 = Interac application selection is enabled.
Domestic VISA Debit Application Selection	0019_0008	1	Determines if the Domestic VISA Debit application selection is enabled. • 0 = Domestic VISA Debit application selection is disabled. • 1 = Domestic VISA Debit application selection is enabled.
Allow CVM modification by POS	0019_0009	0	Allow CVM modification by POS. • 0 = Do not allow CVM modification by POS (QPS, EPS, etc.) • 1 = Allow CVM modification by POS
Configurgation file for contact EMV to load at boot time	0019_0010	EMVCLESS.XML	This parameter can be overidden by the '33.08' message. The name and path of the last file loaded can be retreived by the `600` variable. If left blank, EMVCONTACT.XML will be loaded. The source folder for this file is determined by '0091_0031'
Configurgation file for contactless EMV to load at boot time	0019_0011	EMVCLESS.XML	This parameter can be overidden by the '33.08' message. The name and path of the last file loaded can be retreived by the `601` variable. If left blank, EMVCLESS.XML will be loaded. The source folder for this file is determined by '0091_0031'

7_2_11 Form Files (forms.dat)

The first 18 forms are the standard forms described in the 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.
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.

Parameter (Form) Name	DFS Data Index	Default Value	Description
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).
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.
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).

Parameter (Form) Name	DFS Data Index	Default Value	Description
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.
EMV Select Account	0030_0031	EACCOUNT.K3Z	EMV select account.
EMV Select Language	0030_0032	ELANG.K3Z	EMV select language.
EMV Confirmation	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.

Parameter (Form) Name	DFS Data Index	Default Value	Description
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.
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.

7_2_12 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'.

7_2_13 Main Flow (mainFlow.dat)

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

<table>
<tr><th>Parameter Name</th><th>DFS Data Index</th><th>Default Value</th><th>Description</th></tr>
<tr><td>Duration to Display Results</td><td>0007_0001</td><td>30</td><td>This parameter is used in conjunction with parameters '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>
<tr><th>Parameter</th><th>Message</th></tr>
<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>
</table>
The value of this parameter 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 1/10th of a seconds.
In order to enable configuration parameters '0007_0037' through '0007_0043' to individually control the display duration for these messages, this parameter ('0007_0001') must be set to '2'.</td></tr>
<tr><td>Default Language</td><td>0007_0002</td><td>1</td><td>This parameter specifies the default language. For example:
1 = English
2 = Spanish (Espanol)
The value must be less than or equal to the number of supported languages defined in Number of Supported Languages parameter '0007_0003'.</td></tr>
</table>

Parameter Name	DFS Data Index	Default Value	Description
Number of Supported Languages	0007_0003	2	This parameter specifies the number of languages the customer may choose from. • Maximum value = "3." If more than one language is specified, the "Select Language" prompt may be displayed at the beginning of a transaction.
Combine Language with Card Swipe Screens	0007_0004	1	This parameter can only be enabled if the '0007_0009' Customer Action parameter is set to '0'. When this parameter is enabled, the terminal will display a single form which combines the "Slide Card" and "Select Language" prompts, which provides the cardholder with the option of selecting the language prior to 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	This parameter 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	This parameter 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 cash back process was not used during the transaction, the RBA goes back to waiting for the amount message from host.
Clear Line Display on Reset	0007_0007	1	This parameter specifies if 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	This parameter 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 1/10th of a second.

Parameter Name	DFS Data Index	Default Value	Description
Reset Response	0007_0009	2	This parameter 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	This parameter specifies if 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	" "	This parameter specifies a custom application name to display on the application splash screen. • Maximum of 25 characters.
Splash Screen Version Number	0007_0012	" "	This parameter specifies a custom application version number to display on the application splash screen. • Maximum of 25 characters.
Splash Screen Custom Part Number	0007_0013	" "	Custom part number for splash screen.
Beep Volume	0007_0014	25	This parameter 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 parameter is only supported on the following terminals: • iSC250 • iSC350 • iSC480 • iWL250 • iSMP

Parameter Name	DFS Data Index	Default Value	Description
Alternate On Demand Mode	0007_0015	0	This parameter turns on or turns off the alternate demand mode, which enables or disables 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.
Terms and Conditions Form: Accept Button	0007_0019	0	This parameter determines when the user will be able to see and press 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	This parameter toggles an option to hide the <Up> button when at the top of the text, and hide the <Down> button when 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	This parameter 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 1/10th of a second.

Parameter Name	DFS Data Index	Default Value	Description
After Approval /Decline Display	0007_0023	1	This parameter 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' should be '0' or '1'.
Wait to Send Authorization Request	0007_0024	0	This parameter 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	This parameter 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	0	This parameter 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	This parameter 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). 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	This parameter controls 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 P2P encryption 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, however, is the exception to this, and will leave the expiration date and the CVV blank if not entered manually.
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 RBA application. Must unpair via the Telium Manager. This is otherwise equal to a customized string that enables Bluetooth unpairing when the configured string is scanned via the barcode scanner.

Parameter Name	DFS Data Index	Default Value	Description
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. • 0 = Disabled. • 1 - 200 = number of minutes of inactivity before putting the terminal into sleep mode. This parameter is ignored if contactless and/or smart card readers are active.
Automatic Power Down	0007_0035	60	Automatic power down for portable terminals. • 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.
Display Backlight Intensity	0007_0036	70	Display backlight intensity 0-100% . • 0 = Off. • 1 - 100 = brightness level (%). Very low levels may cause backlight to appear to be off, depending on the terminal model. Set display timeout for on-demand message result individually with parameters '0007_0037' to '0007_0043'. These parameters are only used when parameter '0007_0001' is set to '2'.
20. Signature Request Message Result Display Duration	0007_0037	30	Duration to display result of 20.x: Signature Message (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 1/10ths seconds.

Parameter Name	DFS Data Index	Default Value	Description
21. Numeric Input Request Message Result Display Duration	0007_0038	30	Duration to display result of 21.x: Numeric Input Request Message (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 1/10ths seconds.
23. Card Read Request Result Display Duration	0007_0039	30	Duration to display result of 23.x: Card Read 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 1/10ths seconds.
24. Form Entry Request Result Display Duration	0007_0040	30	Duration to display result of 24.x: Form Entry 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 1/10ths seconds.
25. 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 1/10ths seconds.
27. Input Message Result Display Duration	0007_0042	30	Duration to display result of 27.x: Alpha Input Message (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 1/10ths seconds.
31. PIN Entry Message Result Display Duration	0007_0043	30	Duration to display result of 31.x: PIN Entry Messages (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 1/10ths seconds.

Parameter Name	DFS Data Index	Default Value	Description
Terminal Country	0007_0044	0	Terminal country. • 0 = USA. • 1 = Canada.
Enable 24 Hour Reboot	0007_0045	0	Automatic reboot after 24 hours continuous PIN pad run time. • 0 = Disable. • 1 = Enable. Once this setting has been enabled on a PCI v4 compatible terminal, it cannot be disabled. The terminal will reboot once, the first time this feature is enabled. The counter is reset at every reboot. Once this feature is enabled, the unit must be returned to Ingenico if you choose to have it disabled. This functionality is only supported in production terminals. The timer will not be enabled on mock-up terminals.
Daily Reboot Time	0007_0046	0	Daily reboot time in 24-hour format. This is the set time at which the terminal will reboot. • 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 will reboot when it enters the next idle state.
Status Message Configuration	0007_0047	1	This parameter is configured whenever the 09.x Card Status Message is sent. This is only used when '0019_0001' (EMV Support) and '0020_0001' (WIC Support) are both set to '1'. • 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.

7_2_14 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
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
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	1	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
Append Track 2 for Non-Standard Cards	0003_0018	0	Appends Track 2 to 23.x and 50.x for non-standard cards to prevent terminal reboots and card read errors. CVV and expiration date are entered by POS. • 0 = Disable • 1 = Enable

7_2_15 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 D 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

7_2_16 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	This parameter specifies how long to wait for the customer to enter a PIN before timing out. • 0 = 60 seconds (maximum value) • 1 - 600 = Time in 1/10th of a second
First Key Timeout for PIN Entry in Seconds	0006_0002	0	This parameter 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 1/10th of a second
Between Keys Timeout for PIN Entry in Seconds	0006_0003	0	This parameter 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 1/10th of a second
0-Length PINs	0006_0004	0	Allow 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	This parameter specifies how long to display the “Ask for Assistance” message before timing out. • 0 - 65,000 = Time in 1/10th of a second
PIN Encryption Method	0006_0007	1	This parameter 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	This parameter specifies a Master/Session key to use for encrypting the PIN information. • 0 - 9 indicates the key index to use
Minimum Number of Digits for PIN Entry	0006_0011	4	This parameter 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	This parameter 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'.

7_2_17 Security BIN (secbin.dat)

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

> **Info**
>
> If your organization requires changes to the 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.

> **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.

> **Info**
>
> This policy applies to all MSR encryption methods.

Descriptions for the various card types are same as prior versions.

> **Note**
>
> The start and end BINS are compared up to 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 will be 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 is dependent 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 will be encrypted.

Refer to the following table which provides BIN information for various card types.

Card Type BIN Information

Parameter Name	DFS Data Index	Start BIN Range	End BIN Range	Acct. Num Min Length	Acct. Num Max Length	Encrypt (0=no, 1=yes)
Discover 1 card	0092_0002 (bin1.dat)	601100	601199	14	16	1
Discover 2 card	0092_0003 (bin2.dat)	622126	622925	14	16	1
Discover 3 card	0092_0004 (bin3.dat)	644000	649999	14	16	1
Discover 4 card	0092_0005 (bin4.dat)	650000	659999	14	16	1
MasterCard card	0092_0006 (bin5.dat)	510000	559999	14	16	1
VISA card	0092_0007 (bin6.dat)	400000	499999	13	16	1
AMEX – Range 1	0092_0008 (bin7.dat)	340000	349999	14	15	1
AMEX – Range 2	0092_0009 (bin8.dat)	370000	379999	14	15	1
All Other Cards	0092_0010 (bin10.dat)	000000	999999	10	20	0

Info

'0092_0011' through '0092_0030' are reserved for expansion (up to `bin30.dat`). Follow the same format as the others.

7_2_18 Security Parameters (security.dat)

This section describes the security parameters in the `security.dat` file. Refer to the following table.

Security Parameters in the security.dat File

Parameter Name	DFS Data Index	Default Value	Obsolete DFS Data Index (if applies)	Description
Enable Track Data Encryption	0091_0001	0	0003_0012	Encrypt track 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 TEP3 (Not yet supported) • 7 = Monetra CardShield • 8 = Mercury Payment Systems (MPS) • 9 = RSA-OAEP • 10 = TransArmor • 11 = TDES DUKPT Generic • 12 = S1 • 13 = ECC ⚠ **Note 1** To set data value to '2' for On-Guard encryption, a separate E2ECFG.PGN must be loaded to make this change. This cannot be updated with security.PGZ as the others are updated.
Track Data Encryption Key Index	0091_0002	4	0003_0006	Encryption key index for encrypting track data in messages. • Only valid if '0091_0001' is set to 1, 2, 3, 7, 8, 9, 10, 11, 12 or 13. RSA-OAEP/TransArmor ignores the value of this parameter. Encryption key slots are not used.

Parameter Name	DFS Data Index	Default Value	Obsolete DFS Data Index (if applies)	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. • Maximum = 6. RSA-OAEP/TransArmor ignores the value of this parameter. The default value of 6 is hard coded for RSA-OAEP/TransArmor.
Voltage-specific Settings: (0091_0005 through 0091_0011, 0091_0017)				
Max Number of Transactions with Same Key	0091_0005	0		Maximum number of transactions with the same key. This parameter is mutually exclusive with parameter '0091_0006' which has precedence. • 0 = Don’t change keys based on transaction count • 1 -65000 = Change key after this many transactions with the same key
Periodically Change Keys	0091_0006	0		Periodically change keys (Requires setting the device’s date and time). This parameter has precedence over parameters '0091_0005' and '0091_0007'. All letters must be entered 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 XXth 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 non-zero. • 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	Obsolete DFS Data Index (if applies)	Description
Append ETB to Authorization Request (50.x) Message	0091_0008	0		Append Encryption Transmission Block (ETB) to the 50.x: Authorization Request message. • 0 = Do not append. • 1 = Append.
Identity String	0091_0009	id@sample.com		Identity String. • Provided by authorizer. "id@sample.com" is sample data – not for production.
Identity State	0091_0010	*		Identity State. • Use format mmddyyyy. • If set to "*" the device's current date will be used. Be sure to set the date and time via the 28.x: Set Variable Request message.

Parameter Name	DFS Data Index	Default Value	Obsolete DFS Data Index (if applies)	Description

Parameter Data Encoded in base64	0091_0011			Sample data – not for production (actual data must be obtained from your authorizer):
				`AkkI5lUTQBRABHW+Q8WeFmGOQOXIe6Av2WvRh`
				`sSVLm1BWKza4u/1JCnS6dPW5L1pC8xd6H/7Uc`
				`0vAWh8Na07Ge5lsfCW3wbPRDouotjVd7QJoOD`
				`Lqe1CGQpNJSN/Wbx5j3E5iYnWuEhSifDqy9M2`
				`zZzXWAHdzd6vkRrFkLRyavYGUsaJa4fzg1y/B`
				`EOYWaURa9I4uUD0exrG4VWrRdQ5GWJGQ3oNQC`
				`/U9Bd6FaOxfnAKttNGiidLN5mNB6ZxjuBGMY8`
				`jM3WxJKvmceyyIW18t4ddUILrTptQ2ZU1BkoZ`
				`jlbJvLuggdGv2lDZP4CMtznrvC0c5II8QaoRD`
				`7+fIpfYZZR+789TqSo3YeI5BB5rdv9iR/ul0ct`
				`/+DOUJSUERhVjTps2N5sJ5BX4OJy3dnDBDiL`
				`ulQK2VBLtiJ1QNJyPmfZPUBGT5xGlMrbXw4e5`
				`WX87bmc9jAkXjZkgoP5pJxm/MXuD1vtCcNVCrFnZ`
				`/Qi5SIPEvxxYnZkfCfzWGJMPgwA1L2T3VF`
				`BfTQeTBLBLTV8us5rsGUce/nyzuDs4ZFRbOeq`
				`VcWEdrnZTGWSUC97YA0AFsxT6GkbFR1qv7OAP`
				`Izeyw1W6WpciGC4bgATdeIxSD76mm+kPy4Dwc`
				`kTFdVKiDdo4N6rIBUjph1eV5/StpNXQ4vmwTM`
				`hbo2v9b+5WgcGboiWcAcc8AiuDzeYxZL2iJUY`
				`077U3dYKpTsYLZuWaLFYiLuwgdBQ+lsZrTYxPEjx`
				`/bp4JHvxMl8tBtU5l3DRj0rk4H7H/syGDtb7z`
				`//////////////v////////////fPmI+Yr`
				`YeuVmH8NJpSgBYtALj7Vu2ypnWI65c0CPZRCW`
				`lYxkCtTybOIGej5cYKTxLFyP12yWn017VRN+N`
				`BnozJlF6nMoOEw+GWdSzo7kpi+Et8Yyt0LEA+`
				`BPu7vGvUFUmD67v21PL7i+UYjHMN7erBhbGIB`
				`+e8ywafDBEQqaIOFza3libHVlLnZvbHRhZ2UuY29t`

End Voltage-specific Settings (above)				
Masking Character	0091_0012	0		Defines the character to use for masking the PAN. For 0091_0001 only ‘0’ (zero) and ‘*’ (asterisk) are valid. RSA-OAEP/TransArmor ignores the value of this parameter. The default value of 0 (zero) is hard coded for RSA-OAEP/TransArmor.

Parameter Name	DFS Data Index	Default Value	Obsolete DFS Data Index (if applies)	Description

Public Key encoded in Base64	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.
				
				`MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB`
				`CgKCAQEAvyYRQA3cS4wV9yk+6bFzA7KLDmE+D`
				`/SOP+Q5bNOPG9nUDkAPalRBz12KA5SDxTw2`
				`vO1BIeSFUQlYTpzEDb/XkfNNm5e6nqf12M4k`
				`dHP1F2EXW4WArilUZegAVw/Y7FvAkA8PQFbfgmBirSa5GS`
				`/fuAHjemqEf0DxIgq552IDeFw3`
				`nB0vccK6ePue5sVB9Sm2vWpKm/lj2UE4P6z2`
				`ngZr5V31cSAVN08USxHvz+MEnoUBKt6aKvfR`
				`UAp4iFyIpxlp4eylxY8zizPekS29lcRMsI9`
				`hGug2CoKFhhUJ1gD8G280zIoWCxysNvl40k`
				`/l8OTtPKrnlhAzQcIyy/RB0lwb6QIDBlU1`

Exponent Value for RSA-OAEP /TransArmor	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) This parameter may need to be modified – check with your key authority if you are unsure.
Key ID	0091_0015	12345678901		For TransArmor encryption, this value is used to map the public key. Its length should be 11-bytes. This is not required for RSA-OAEP encryption.

Parameter Name	DFS Data Index	Default Value	Obsolete DFS Data Index (if applies)	Description

Encryption Target	0091_0016	2		This parameter is used to select 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 currently ignored for RSA-OAEP encryption.

Length of encrypted CVV returned by Voltage	0091_0017	8		This parameter is for use with Voltage P2PE encryption types only. It defines the length of the encrypted CVV returned and applies to Voltage TEP1 and TEP2 encryption. It is used only for manual entry. • Valid lengths are 7 – 23 digits. ⓘ If an invalid length is assigned (e.g., a length of ‘54’), the length will default back to 8 digits.

Terminal Authentication	0091_0018	0		Device 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 the 93.x: Terminal Authentication Message for a description of the device communication authentication process.

Generic Message Encryption	0091_0019	9		Encryption Type for Barcode. • 0 = Disable. • 9 = RSA encryption.

Parameter Name	DFS Data Index	Default Value	Obsolete DFS Data Index (if applies)	Description

Parameter Data Encoded in base64 Barcode Key	0091_0020	(392-character string)		Parameter data encoded in base64 Barcode Key:
				`MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB`
				`CgKCAQEAnScvfGCiutFOzQxjN14ZS+LbeexE`
				`VJnypJG2M7rHEmf4C0xS/Q/+HFyudCdB1J7s`
				`XO2wI4xXrmMQmnFqXECGnwDtdUDx1zQVRXt/`
				`R2pPipmoaa+qmeTRRUZhtJLpg9TBhWMNfwML8FIEUwPe1JPL`
				`/nyJJHb5ke2pJiqT1VEqg5KDivpgP`
				`/w2DsR0xfLPzHqxtrxYwG7U3czpJmaT`
				`WTRFQ6VNgmxbNdPBeYEBA0VZMzYlZHis1X48`
				`djoUrlvMY51+pyCp954gXVsxeNjMnQiz6WKE`
				`QE5GzOYYLzpeqd5ulQjWAiCDX5S88jSnRPI7mbPucW`
				`//m8aOmWElVY9EHqPwIDAQAB`

Exponent For TransArmor Barcode	0091_0021	010001		This value is the Binary equivalent of 65537. Do not change this value.
Public Key ID	0091_0022	12345678901		This value is mapped to the public key and its length should be 11 bytes.

<table>
<tr><th>Parameter Name</th><th>DFS Data Index</th><th>Default Value</th><th>Obsolete DFS Data Index (if applies)</th><th>Description</th></tr>
<tr><td>RBA Escape Sequence</td><td>0091_0023</td><td>GYRYG</td><td></td><td>This parameter is currently only supported by the iUN250. This is the sequence of keys used to exit the RBA from the boot screen. The sequence of keys must be exactly 5 characters in length. Should an invalid sequence be entered, the sequence will default 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>ECC Universal Initial Public Key</td><td>0091_0024</td><td>" "</td><td></td><td>ECC Universal Initial Public Key.</td></tr>
<tr><td>ECC Public Encryption Key</td><td>0091_0025</td><td>" "</td><td></td><td>ECC Public Encryption Key.</td></tr>
</table>

Parameter Name	DFS Data Index	Default Value	Obsolete DFS Data Index (if applies)	Description
Input Messages Encryption	0091_0026	0		Input messages encryption (uses '0091_0020' key). • 0 = Disabled. • 1 = Enabled.
Barcode Messages Encryption	0091_0027	1		Barcode messages encryption (uses '0091_0020' key). • 0 = Disabled. • 1 = Enabled.
Driver's License Encryption	0091_0028	0		Driver's license encryption (if '0091_0001' is set to RSA/TransArmor). • 0 = Disabled. • 1 = Enabled.
Non-Standard Card Encryption	0091_0029	0		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 the RBA that 12.x: Account Messages are to be blocked when encrypting MSR 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 configuration. • 0 = XML files are unsigned and stored in the HOST directory. • 1 = XML files are signed and stored in the System directory.
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.

Parameter Name	DFS Data Index	Default Value	Obsolete DFS Data Index (if applies)	Description
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.

If your organization requires changes to the `security.dat` file, you must follow the procedure to obtain approval and signature from Ingenico prior to your implementation. See section 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 once implemented.

This policy applies to all MSR encryption methods.

Also refer to the RSA-OAEP and TransArmor Encryption section for additional information specific to RSA-OAEP and TransArmor encryption types.

7_2_19 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	This parameter 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 1/10th of a second
Send Message When Signature Ready	0009_0002	0	This parameter 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.
Max Time to Allow Before Signing (pen down timeout)	0009_0004	0	This parameter specifies how long to wait for a customer to begin the signature before timing out. • 0 = Timeout disabled • 1 - 65,000 = Timeout in 1/10th of a second
Signature encoding Format	0009_0005	9	This parameter determines the signature encoding. • 9 = 3-byte ASCII • 11 = Ingenico iSCxxx terminals will encode the signature data in the Legacy format and encode using base64.
Save State on Signature Capture Request (20.x message)	0009_0006	0	This parameter 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. To understand the impact of this parameter on other messages, see also sections On-Demand, Communication Messages, and 20.x: Signature RESPONSE Message (On-Demand) for additional details.

Parameter Name	DFS Data Index	Default Value	Description
Minimum Acceptable Signature Size	0009_0007	50 bytes	This parameter specifies the smallest signature (in bytes) that is acceptable. If a signature below this limit is entered, the signature is deleted and a new signature is requested.
Display Signature Until Download	0009_0008	0	Display signature until download starts. • 0 = Disable • 1 = Enable
Number of Bytes in Signature Block	0009_0012	200	This parameter specifies the number of bytes in the signature block. • 1 - 1,000 = Number of bytes
Automatic Signature Acceptance Timeout	0009_0013	0	This parameter 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 1/10th of a second
Maximum Allowable Signature Size	0009_0014	750	Maximum acceptable 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	Type of message response to send when the signature is ready. This parameter only applies if the “Send message when signature ready” ('0010_0003') 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.

7_2_20 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.

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 completed.
Error	0097_0015	“Error”	Error.
Input	0097_0016	“Input”	Input.
Input Accepted	0097_0017	“Input Accepted”	Input done.

Parameter Name	DFS Data Index	Default Value	Description
Select Language	0097_0018	“Select Language”	Select language.
Advertising	0097_0019	“Advertising”	Advertising via 30.x: Advertising Request 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.

7_2_21 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`.

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.
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.

7_2_22 User Defined Variables (var.dat)

This section defines 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.)	These are the default values of the user defined variables as set by the 28.x: Set Variable Request.

7_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 (‘ % ’) *MAY* 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.

7_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.

7_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.
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.

7_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

7_3_4 Unknown Format Specifiers

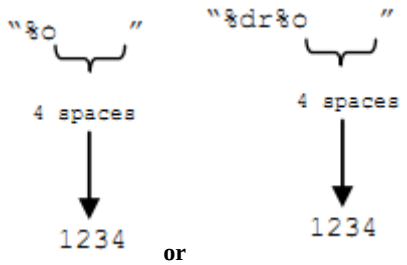
Unknown format specifier characters are now ignored. In the following examples, the format specifier characters enclosed in top-and-bottom asterisks are ignored:

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

Typically, when a format specifier or prompt is unavailable, an entire-screen text box is displayed indicating an error. When UIA encounters a missing format specifier, the message “missing specifier” is displayed in the clear entry input box. A missing format specifier most likely indicates that the format specifier was invalid. Examples of invalid format specifiers are non-numeric (like “%fhello”) or negative min/max values. Another example is “%m%M4” where no value is provided to the Minimum attribute.

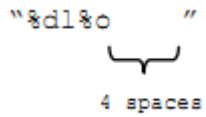
7_3_5 Examples of Format Specifiers



or

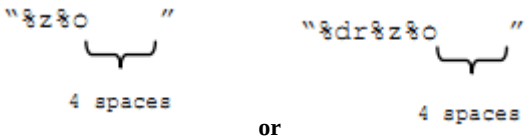
Input Example for Overwrite Format Specifier (see Note)

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

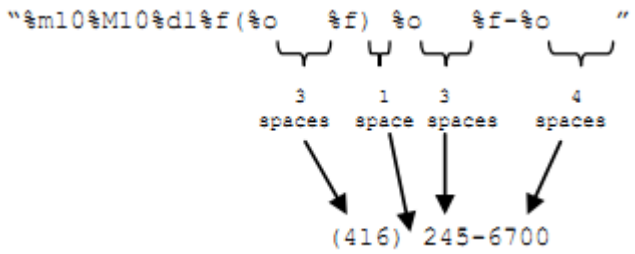
Input Example for Overwrite and Leading Zeroes Recognition Format Specifiers (see Note)

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

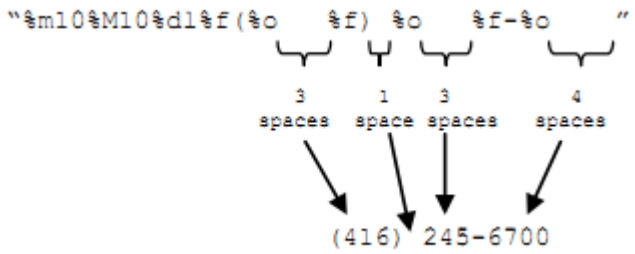
Input Example for Overwrite, Maximum, and Minimum Characters Format Specifiers (see Note)

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.



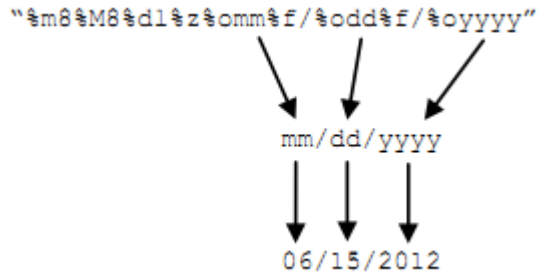
Using Specifiers to Prompt for a US Phone Number

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



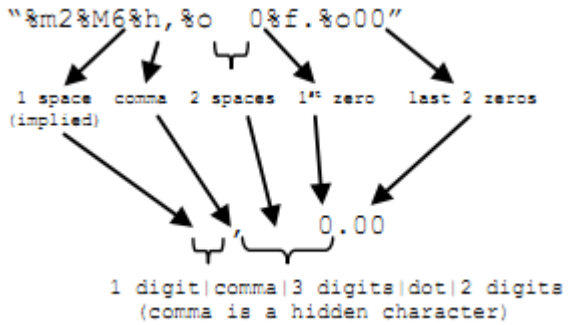
Using Specifiers to Prompt for a Social Security Number

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



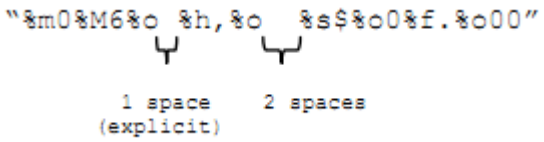
Using Specifiers to Prompt for a Date

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”	
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"	
5	" 123.45"	
6	"1,234.56"	The hidden comma appears.
7	"1,234.56"	Long/bad beep because maximum digits entered.



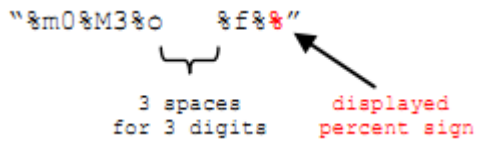
Using Specifiers to Prompt for a Dollar Amount, Method 2

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.
	Three (3) digits may be entered to the left of the displayed percent sign.

7_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 Message (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`	
<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`	

8_EMV Implementation

This section provides detailed information on 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 now managed by EMVCo, a company jointly owned by MasterCard, Visa, American Express, Discover, JCB, and UnionPay. EMV transaction specifications have been developed by EMVCo to ensure global compatibility between EMV cards, payment terminals, and ATM’s.

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

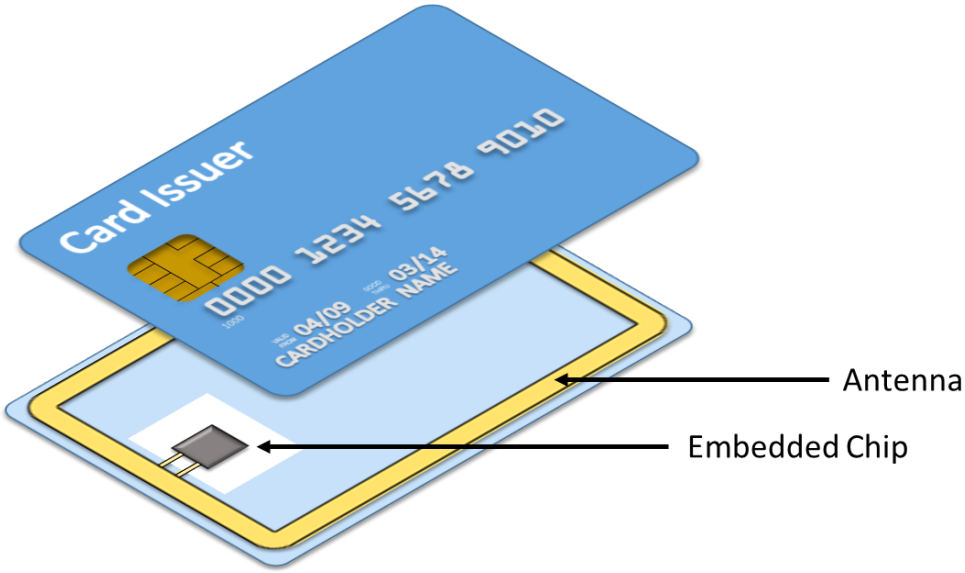
EMV cards, also referred to as smart cards or chip cards, contain an embedded microchip which is configured to provide a more secure transaction than a standard magnetic stripe card transactions. Whereas a magnetic stripe card contains secure payment information which does not change, an EMV card can encrypt data differently with each transaction. See the following figure for an illustration of an EMV card.

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. This 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 when the POS sends the '14.01' Set Transaction Type message to the terminal. This message defines the transaction as a purchase or refund. The next action is to insert the EMV card which must be detected by the terminal in order to proceed with the transaction. 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 will be prompted to “INSERT CARD IN CHIP READER,” and the terminal will wait for the card to be correctly inserted. For a contactless card, tapping the card (holding the card close enough to the contactless car reader) will serve the same purpose.

If EMV is enabled in the configuration file, then subsequent transactions will be configured EMV transactions each time the application enters the online idle state. This allows the purchase transaction to start immediately once the card is inserted in the terminal. Once the card is inserted, the terminal may request the cardholder to select the language, or to select the desired application. As a note, the first digit of the service code for EMV cards will be ‘2’ or ‘6’ with the BIN range indicating that EMV transactions are supported.

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 which 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 has been selected. Refer to the following table for 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 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', then 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'
- '2' = AID is selected by the cardholder via menu without cardholder confirmation.
  - View flow diagram Application Selection for '0019_0003' = '2'
- '1' = cardholder is prompted to confirm AID automatically starting with card's highest-priority AID.
  - View flow diagram Application Selection for '0019_0003' = '1'
- '3' = cardholder is prompted to confirm AID automatically starting with card's highest-priority AID but the RBA auto-selects the lowest-priority application without cardholder confirmation if all other applications are declined.
  - View flow diagram Application Selection for '0019_0003' = '3'

For the list of available AIDs used in EMV transactions, refer to Available EMV Application IDs. This section includes Tag `T8A` values for bad PIN entry and the DFS Data Index associated with each application listing.

Refer to Canadian Specific Requirements for more information pertaining to Canadian applications.

If an EMV card is swiped instead of being inserted, the terminal displays “INSERT CARD IN CHIP READER” and waits for the card insertion if the service code is set for chip.

The first digit of the service code for EMV cards will be '2' or '6' and their BIN range indicates that EMV is supported.

8_2_3_3 Canadian Specific Requirements

The Canadian Application Selection flag process requirement has been incorporated to facilitate compliance with Interac Direct Payment (IDP) specifications, the national EMV specifications for Canada. This enables the card issuer to encode multiple applications on one card. This also enables the card issuer to determine which application will be the primary when the card is used at a POS, and which application will be the primary when the card is used at an ATM.

As an example, a card may be encoded with a VISA Plus AID as the primary AID for POS transactions, and an Interac AID as the primary AID for ATM transactions. When the card is inserted into a retail device (POS) in Canada, only the VISA Plus AID will be included in the list of applications available for the transaction. When the card is inserted into an ATM in Canada, only the Interac AID will be included in the list of applications available for the transaction. When the card is used outside of Canada, both AIDs will be included to indicate that both applications are available for POS and ATM transactions.

The user of the application may activate or deactivate the application selection for Interac Application Selection and Domestic VISA Debit. This feature is controlled by two parameters in the EMV.DTA file:

- Interac Application Selection Flag (ASF) '0019_0007'.
- Domestic VISA Debit Application Selection Flag '0019_0008'.

A “0” value indicates that the corresponding application selection will be disabled while a “1” value indicates that the application selection will be enabled. Also refer to the EMV.DAT Configuration File and MAC Messages (Canada Only) for more information.

8_2_3_4 Available EMV Application IDs

This section lists the available Application IDs for EMV transactions.

Available EMV Application IDs

The "Allow PIN Bypass" parameter determines whether or not the customer is allowed to bypass PIN entry by pressing the "Cancel" key. This option is only available if the card allows it. PIN bypass is not be allowed if the Application ID is not found. If this option is enabled but not allowed by the card, then the transaction will be declined. To summarize the requirements for PIN bypass:

- The Application ID must be found.
- The "Allow PIN Bypass" parameter must be set to '1'.
- The card must allow PIN bypass.

The "Force Casback" boolean field allows for override of an issuer/AID's cashback setting to force/deactivate cashback regardless of AUC cashback bits:

- '0' = override AUC cashback bits (e.g. for Interac)
- '1' = require AUC cashback bits (default)

Refer to the following table which lists and describes the available EMV Application IDs. The AID Brand value for each `emvaid.dat` item corresponds to a parameter in `emvbrand.dat`, as shown in the second table below. For example, the '0021_0002' parameter's AID Brand value of '01' refers to `emvbrand.dat` parameter '0022_0001'.

Available EMV Application IDs (in emvaid.dat)

DFS Data Index	Application ID	8A Tag for Bad PIN	Allow PIN Bypass Parameter '0' = No '1' = Yes	AID Brand	US Common Debit AID '0' = Non Debit or Global Debit AID '1' = US Common Debit AID	Force Cashback? '0' = override '1' = require	Card
0021_0001	A0000000031010	55	0	02	0	0	VSDC
0021_0002	A0000000041010	55	0	01	0	0	MasterCard
0021_0003	A0000002771010	55	0	03	0	1	Interac
0021_0004	A0000000032010	55	0	02	0	0	VISA
0021_0005	A00000002501	55	0	04	0	0	AMEX
0021_0006	A0000001523010	55	0	05	0	0	Discover
0021_0007	A0000000980840	55	0	02	1	0	VISA US Common Debit
0021_0008	A0000000042203	55	0	04	1	0	MasterCard US Common Debit
0021_0009	A0000001524010	55	0	05	1	0	Discover US Common Debit
0021_0010	A0000006200620	55	0	02	1	0	DNA US Common Debit
0021_0011	" "						
0021_0012	" "						
0021_0013	" "						

DFS Data Index	Application ID	8A Tag for Bad PIN	Allow PIN Bypass Parameter '0' = No '1' = Yes	AID Brand	US Common Debit AID '0' = Non Debit or Global Debit AID '1' = US Common Debit AID	Force Cashback? '0' = override '1' = require	Card
0021_0014	" "						
0021_0015	" "						
0021_0016	" "						

EMV Brand Parameters

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

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

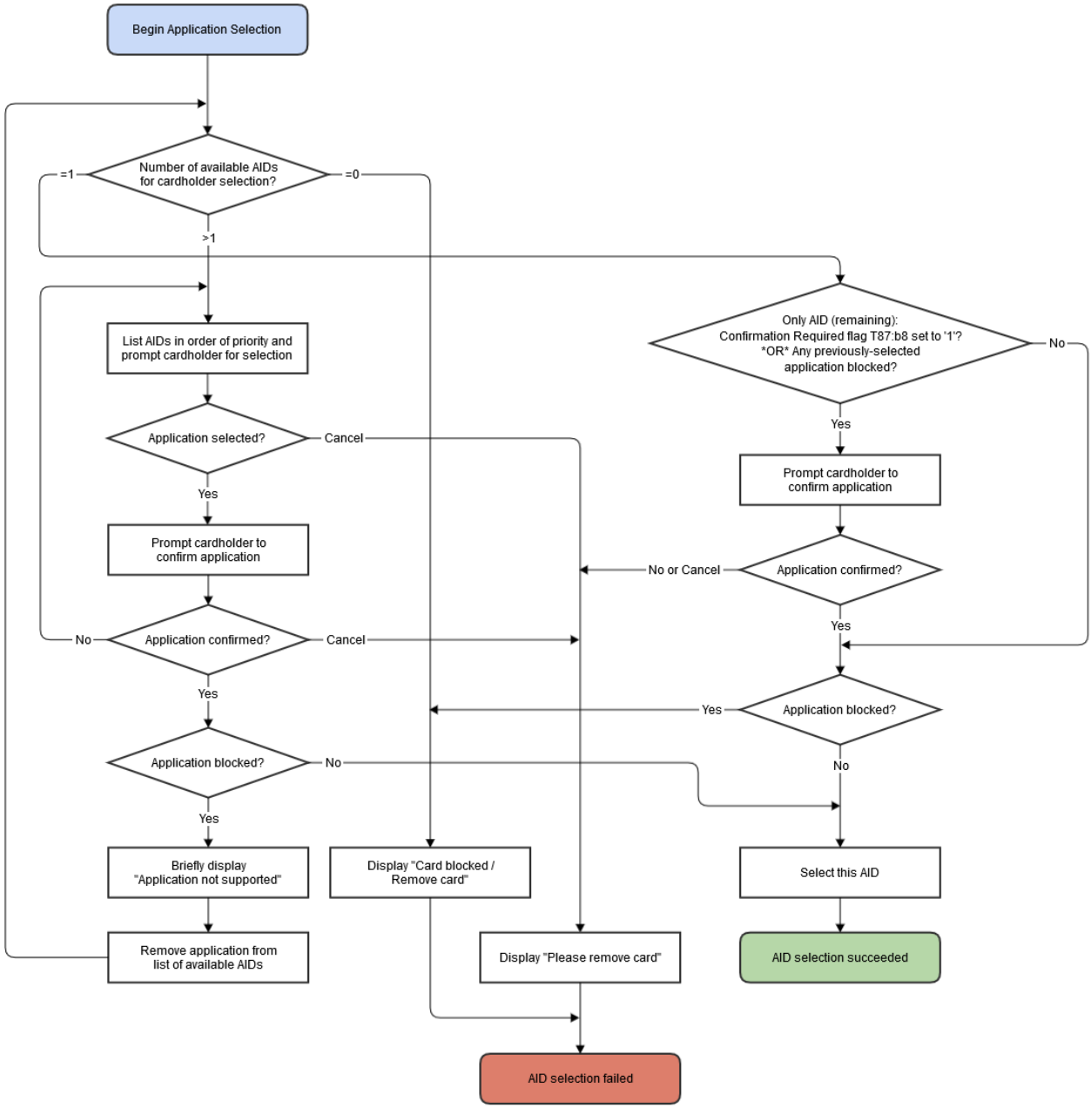
emvbrand.dat Parameters

DFS Data Index	EMV Brand	US Common/Global Debit AID Preference '0' = Do not Prefer US Common or Global Debit AID '1' = Prefer US Common Debit AID '2' = Prefer Global Debit AID	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)

If the merchant wishes to prefer the US common debit AIDs, then set each brand preference to '1'.

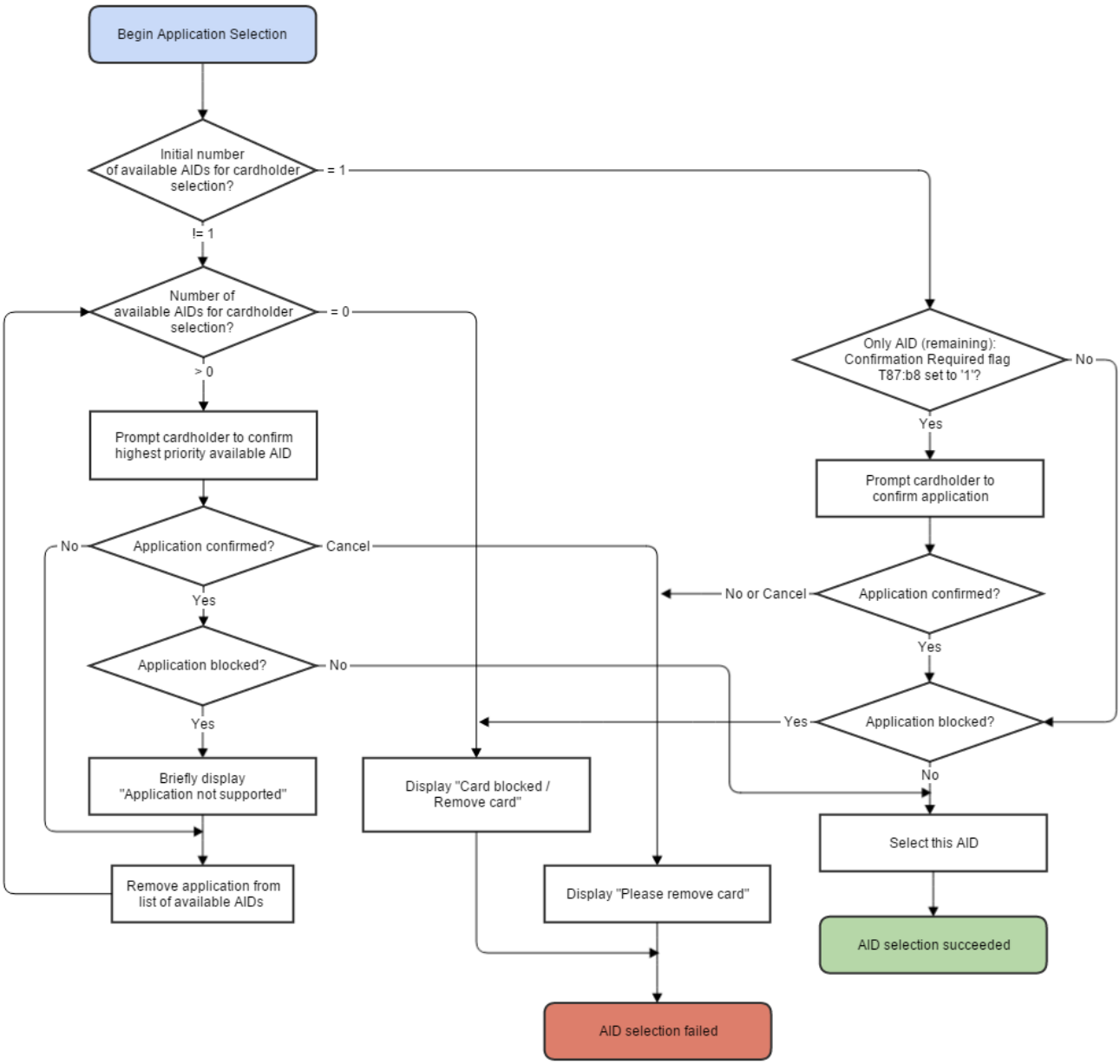
If '0019_0003' = '1' ("pin pad should auto-select the application (AID) during an EMV transaction") then the US Common debit AID will be auto-selected. The customer will not be prompted to select.

8_2_3_5 Application Selection for '0019_0003' = '0'



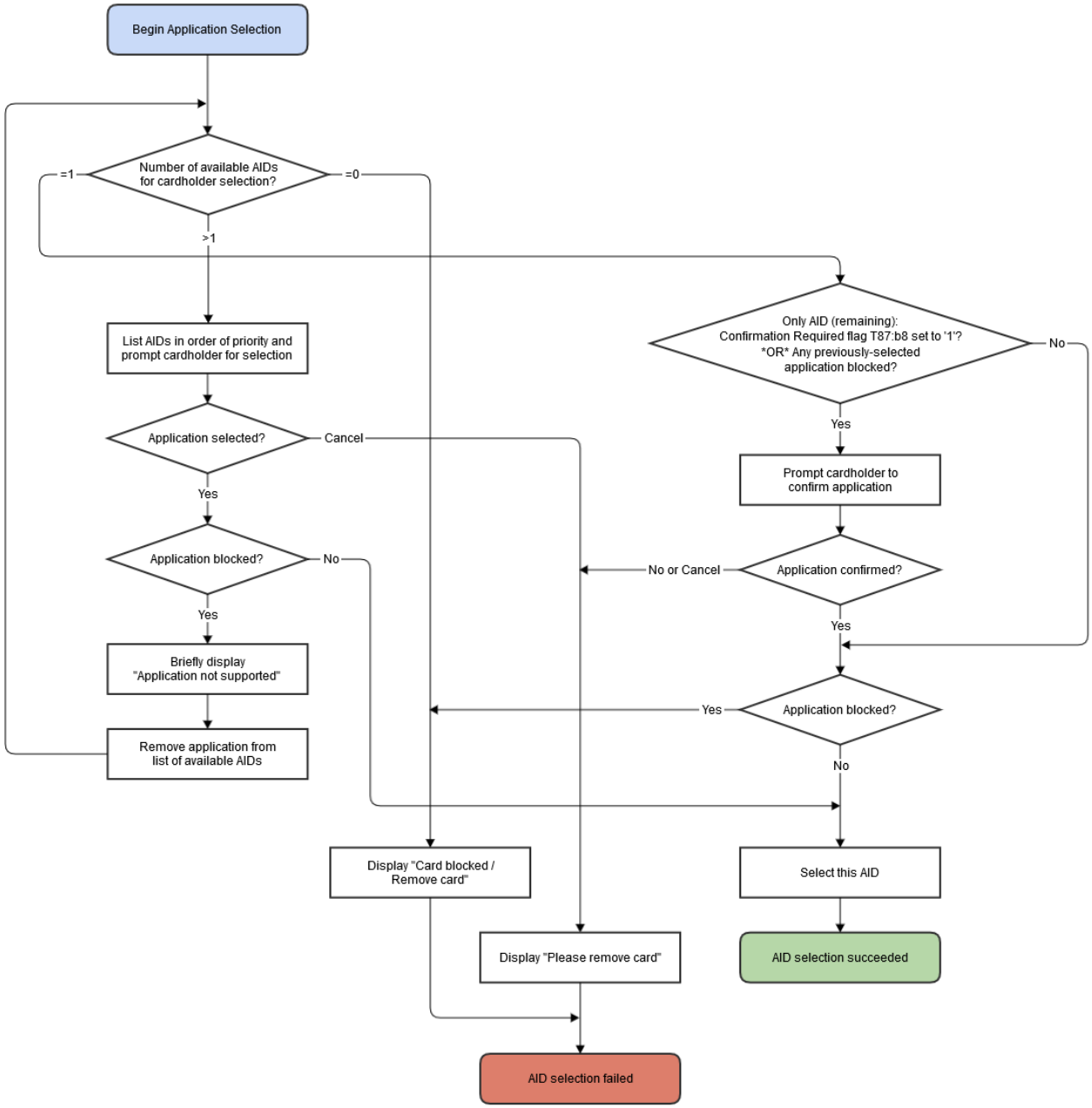
Application Selection for '0019_0003' = '0'

8_2_3_6 Application Selection for '0019_0003' = '1'



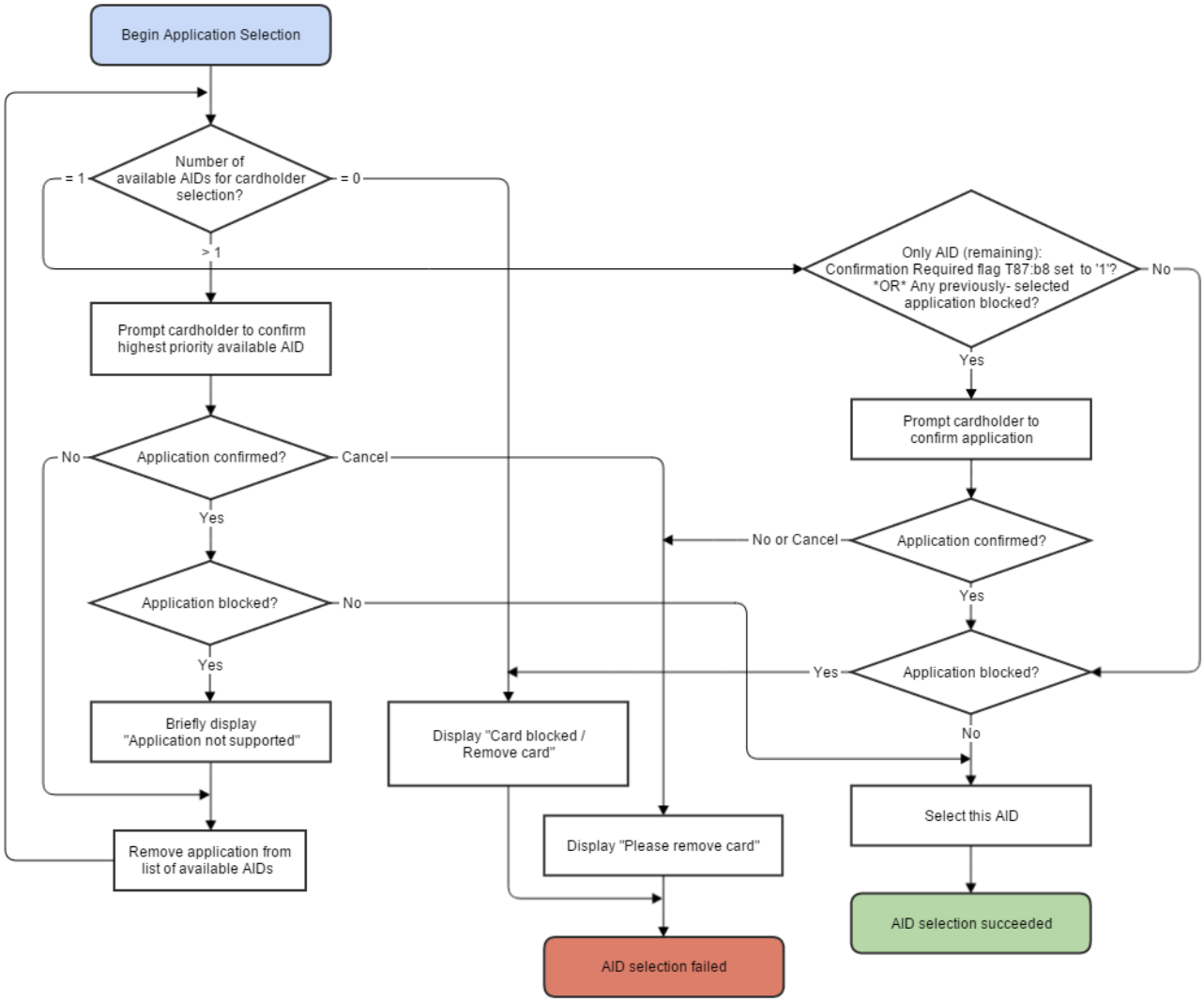
Application Selection for '0019_0003' = '1'

8_2_3_7 Application Selection for '0019_0003' = '2'



Application Selection for '0019_0003' = '2'

8_2_3_8 Application Selection for '0019_0003' = '3'



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

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.

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_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

Once the transaction is completed, the terminal sends a 10.x Hard Reset Message to the POS indicating that it is returning to the online idle state and awaits the next transaction.

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 which is embedded in the message. Subcommand identifiers are used to identify the following message types:

- EMV '33.00.x' Transaction Initiation Message
- EMV '33.01.x' Status Message
- 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 '33.07.x' Terminal Capabilities Message
- EMV '33.08.x' Set Variables Message
- EMV '33.09.x' Set Tag Data Message
- EMV '33.10.x' Get Tag Data Message

Refer to the following table for a description of Subcommand Identifier codes.

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. = Transaction Initiation. • 01. = Status. • 02. = Track 2 Equivalent Data. • 03. = Authorization Request. • 04. = Authorization Response. • 05. = Authorization Confirmation. • 07. = Terminal Capabilities. • 08. = Set Variables. • 09. = Set Tag Data. • 10. = Get Tag Data.
7	...	...	...
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. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 1) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note 1** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).

Offset	Length	Format	Description
EMV Request Header			
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.
10	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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	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.
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.

Offset	Length	Format	Description
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.
1	3	Constant	iConnectEFT Constant = M33_00_EMV_TRANSACTION_INITIATION Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 00. = Transaction initiation.

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_00_RES_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 2) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE ⓘ Failure codes can have values from 1 to 99. ⚠ **Note 2** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always '0' as only one packet will be part of an EMV response message.
10	1	Alphanum	iConnectEFT Constant = P33_00_RES_EMVH_PACKET_TYPE Packet type. Currently, EMV response messages always return '0', indicating the first and last packet.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.

Offset	Length	Format	Description
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 should use the EMV '33.00.x' Transaction Initiation Message both to request status messages for specific EMV transaction steps, and to suspend the transaction during specified steps in the EMV flow (refer to the Transaction Step List). The terminal will reply using the EMV '33.01.x' Status Response message as it reaches the steps requested by the '33.00.x' message. The EMV '33.01.x' Status Response message provides more detailed transaction status to the POS.

The EMV '33.01.x' Status Response message sent from the terminal to the POS indicates the status of the EMV transaction. This message also notifies the POS when the RBA has suspended 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 will show only dashes except for the "card inserted" field. This field will always contain 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 chip card, and only indicates whether a card is present, even if it is only an MSR card. Deactivating this EMV functionality conserves battery life.

8_3_3_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.

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_01_REQ_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 1) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note 1** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_01_REQ_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.

Offset	Length	Format	Description
10	1	Alphanum	iConnectEFT Constant = P33_01_REQ_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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 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 RBA 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, may be empty). this field enables the POS to update this list if necessary. The list contains the transaction steps where the RBA should suspend the flow and await the resume message from the POS. This list should contain one or more of the steps listed in the Transaction steps, concatenated to one another. 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 RBA 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.			
N + 6	1	Constant	FS – 0x1C.

Offset	Length	Format	Description
N + 7	1	Constant	EXT–0x03.
N + 8	1	Binary	LRC check character.

8_3_3_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.

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_01_RES_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 2) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note 2** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
9	1	Alphanum	iConnectEFT Constant = P33_01_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.

Offset	Length	Format	Description
10	1	Alphanum	iConnectEFT Constant = P33_01_RES_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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	1	Constant	FS – 0x1C.
12	2	Alphanum	iConnectEFT Constant = P33_01_RES_TRANSACTION_CODE Transaction code. • 00 = Purchase. • 01 = Refund.
14	1	Alphanum	iConnectEFT Constant = P33_01_RES_F1_CHIP_CARD Flag 1 - Indicates if 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 if the EMV process is started. • – = EMV process is not started. • S = EMV process is started.

Offset	Length	Format	Description
16	1	Alphanum	iConnectEFT Constant = P33_01_RES_F3_EMV_COMPLETED Flag 3 - Indicates if 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. ⓘ 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 if the language is selected. • – = Not selected. • M = Manually selected. • A = Automatically selected.
18	1	Alphanum	iConnectEFT Constant = P33_01_RES_F5_APP_SELECTED Flag 5 - Indicates if 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 if 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 if a rewards '05.' request is received. • – = Rewards request is not received. • R = Rewards request is received. • S = Rewards response sent.

Offset	Length	Format	Description
21	1	Alphanum	iConnectEFT Constant = P33_01_RES_F8_PAYMENT_TYPE_RECEIVED Flag 8 - Indicates if 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 if 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 if this is the last PIN try for the card. • – = This is not the last PIN try. • L = This is the last PIN try.
24	1	Alphanum	iConnectEFT Constant = P33_01_RES_F11_OFFLINE_PIN_ENTERED Flag 11 - Indicates if an offline PIN is entered. • – = Offline PIN is not entered. • P = Offline PIN is entered.
25	1	Alphanum	iConnectEFT Constant = P33_01_RES_F12_ACCOUNT_TYPE_SELECTED Flag 12 - Indicates if an account type is selected. • – = Account type is not selected. • C = Checquing 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 if the authorization request is sent. • – = Authorization request is not sent. • S = Authorization request is sent. • F = Authorization request failed to send.

Offset	Length	Format	Description
27	1	Alphanum	iConnectEFT Constant = P33_01_RES_F14_AUTH_RES_RECEIVED Flag 14 - Indicates if 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 if the confirmation response is sent. • – = Confirmation response is not sent. • S = Confirmation response is sent. • F = Confirmation response failed to send.
29	1	Alphanum	iConnectEFT Constant = P33_01_RES_F16_TRANSACTION_CANCELLED Flag 16 - Indicates if 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 if 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 if a card or application block is detected. • – = Card or application block is not detected. • B = Card or application block is detected.
32	1	Alphanum	iConnectEFT Constant = P33_01_RES_F19_ERROR_DETECTED Flag 19 - Indicates if 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.

Offset	Length	Format	Description
33	1	Alphanum	iConnectEFT Constant = P33_01_RES_F20_PREMATURE_CARD_REMOVAL Flag 20 - Indicates if completion occurred from premature card removal. • – = No premature card removal was detected. • R = Premature card removal was detected.
34	1	Alphanum	iConnectEFT Constant = P33_01_RES_F21_CARD_NOT_SUPPORTED Flag 21 - Indicates if 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 if the post confirmation wait has been started. • – = Post confirmation wait has not been started. • S = Post confirmation wait has started. 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 if 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.
38	1	Alphanum	iConnectEFT Constant = P33_01_RES_F25_TRANSACTION_PREPARATION_SENT Flag 25 - Indicates if 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.

Offset	Length	Format	Description
39	1	Alphanum	iConnectEFT Constant = P33_01_RES_F26_EMV_FLOW_SUSPENDED Flag 26 - Indicates if 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 if Online PIN entry is requested. • – = Initialized state at start of transaction. • R = PIN entry request. • C = PIN entry cancelled (failed due to invalid PIN). • A = PIN block is accepted and valid.
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_CLESS_TRANSACTION_STARTED Flag 29 - Indicates if a contactless transaction has started. • – = Initialized state at start of transaction. • 0 = Contactless transaction is stopped. • 1 = Contactless transaction has started.
43	1	Alphanum	iConnectEFT Constant = P33_01_RES_F30_CLESS_ERROR_FLAG Flag 30 - Contactless error flag. • – = Initialized state at start of transaction. • C = Collision was detected. • R = Re-tap is required.

Offset	Length	Format	Description
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 ECR). 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. ECR prompt for cashback. 3. ECR 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.
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.

Offset	Length	Format	Description
4	3	Constant	Subcommand identifier. • 02. = Track 2 Equivalent Data.
EMV Request Header.			
7	2	Alphanum	iConnectEFT Constant = P33_02_RES_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
9	1	Alphanum	iConnectEFT Constant = P33_02_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.

Offset	Length	Format	Description
10	1	Alphanum	iConnectEFT Constant = P33_02_RES_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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.			
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.			
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

The EMV '03.'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.

EMV '03.' 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.			

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_03_REQ_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
9	1	Alphanum	iConnectEFT Constant = P33_03_REQ_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.

Offset	Length	Format	Description
10	1	Alphanum	iConnectEFT Constant = P33_03_REQ_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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.			
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.
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.

Offset	Length	Format	Description
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. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note below) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 01 to 99.
9	1	Alphanum	iConnectEFT Constant = P33_04_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.
10	1	Alphanum	iConnectEFT Constant = P33_04_RES_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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.			
M	1	Constant	FS – 0x1C.

Offset	Length	Format	Description
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.
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. • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. Currently this message only returns '01' and '07' error status codes.
9	1	Alphanum	iConnectEFT Constant = P33_04_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always '0' as only one packet will be part of an EMV error response message.
10	1	Alphanum	iConnectEFT Constant = P33_04_RES_EMVH_PACKET_TYPE Packet type. Currently, EMV response messages always return '0'.
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.
1	3	Constant	iConnectEFT Constant = M33_05_EMV_AUTHORIZATION_CONFIRMATION Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 05. = Authorization Confirmation.

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_05_RES_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
9	1	Alphanum	iConnectEFT Constant = P33_05_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.

Offset	Length	Format	Description
10	1	Alphanum	iConnectEFT Constant = P33_05_RES_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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.
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.
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

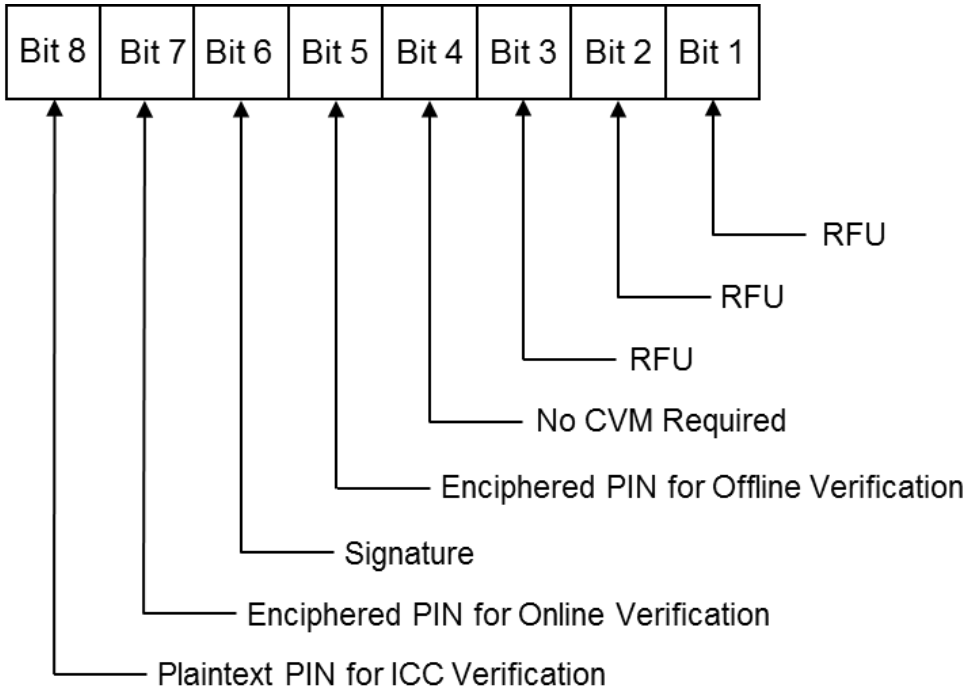
8_3_8_1 Overview of the Terminal Capabilities Request Message

The EMV '33.07.x' Terminal Capabilities Request message is sent from the terminal to the POS to implement MasterCard Quick Payment Service (MasterCard QPS) and VISA Easy Payment Service (VEPS). This message allows the POS to bypass PIN and signature card verification methods for low-level EMV transactions by modifying the EMV tag. The terminal must first be configured to allow CVM modification by the POS by setting parameter '0019_0009' to '1'. Two EMV tags are included in the request message; tag `T84` and tag `T9F33`. Tag `T84` is the Application ID of the application currently being used for the payment transaction. Tag `T9F33` indicates the Terminal Capabilities including the CVM.

After the Application has been selected, the terminal sends a new message to the POS which includes the selected Application ID. 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 should be bypassed for the transaction. If so, the POS returns a modified `T9F33` Terminal Capabilities tag 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 Message

The EMV '33.07.x' Terminal Capabilities Response message is sent in response to the EMV '33.07.x' Terminal Capabilities Request Message. Depending on the Application ID, transaction amount, and possibly additional criteria, the POS sends this response message with a modified `T9F33` Terminal Capabilities tag 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."



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.			

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_07_RES_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 1) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note 1** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
9	1	Alphanum	iConnectEFT Constant = P33_07_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.

<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.
0 = Indicates that the packet is the first and last.
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.</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>T84</td><td>Application ID of application being used for payment.</td><td>Hex ASCII</td></tr><tr><td>T9F33</td><td>Terminal Capabilities.</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>
</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.			

<table>
<tr><th>Offset</th><th>Length</th><th>Format</th><th>Description</th></tr>
<tr><td>7</td><td>2</td><td>Alphanum</td><td>iConnectEFT Constant = P33_07_RES_STATUS
Status code.
00 = OK
01 = FAIL
02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 2)
03 = INVALID_MSG_HEADER
04 = UPDATE_LIST_FAIL
05 = SUSPEND_LIST_FAIL
06 = UPDATE_TIMER_FAIL
07 = TAG_LIST_FAIL
08 = INVALID_TARGET
09 = INVALID_LOAD_SRC
10 = INVALID_FILENAME
11 = FILE_NOT_FOUND
12 = INVALID_CMD_TYPE
Failure codes can have values from 1 to 99.
Note 2
The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).</td></tr>
<tr><td>9</td><td>1</td><td>Alphanum</td><td>iConnectEFT Constant = P33_07_RES_EMVH_CURRENT_PACKET_NBR
Current packet number. Always '0' as only one packet will be part of an EMV response message.</td></tr>
<tr><td>10</td><td>1</td><td>Alphanum</td><td>iConnectEFT Constant = P33_07_RES_EMVH_PACKET_TYPE
Packet type. Currently, EMV response messages always return '0'.</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>
</table>

<table>
<tr><th>Offset</th><th>Length</th><th>Format</th><th>Description</th></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 for 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 of application being used for payment.</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>
<tr><td>P</td><td>1</td><td>Constant</td><td>ETX – 0x03.</td></tr>
<tr><td>P + 1</td><td>1</td><td>Binary</td><td>LRC check character.</td></tr>
</table>

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'. The terminal sends the following message to the POS requesting modification to the card verification method:

'33.07.0000[FS]T84:07:hA0000000031010[FS]T9F33:03:hE0 B0 C0'

The value of tag T9F33 is 'E0B0C0'. The middle byte indicates the CVM method. Note that the current settings have enabled Offline PIN verification, signature, and plaintext PIN entry for ICC verification. The POS returns the following message:

'33.07.0000[FS]T9F33:03:hE0 08 C0'

Note that the value of the middle byte (CVM) has changed. Offline PIN verification, signature, and plaintext PIN entry for ICC verification have been disabled. The "No CVM Required" flag has been set, indicating that 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	→	
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

8_3_9_1 Overview

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'.
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_2 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.

<table>
<tr><th>Offset</th><th>Length</th><th>Format</th><th>Description</th></tr>
<tr><td>7</td><td>2</td><td>Alphanum</td><td>iConnectEFT Constant = P33_08_REQ_STATUS
Status code.
00 = OK
01 = FAIL
02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 1)
03 = INVALID_MSG_HEADER
04 = UPDATE_LIST_FAIL
05 = SUSPEND_LIST_FAIL
06 = UPDATE_TIMER_FAIL
07 = TAG_LIST_FAIL
08 = INVALID_TARGET
09 = INVALID_LOAD_SRC
10 = INVALID_FILENAME
11 = FILE_NOT_FOUND
12 = INVALID_CMD_TYPE
Failure codes can have values from 1 to 99.
Note 1
The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).</td></tr>
<tr><td colspan="4">EMV Request Header</td></tr>
<tr><td>9</td><td>1</td><td>Alphanum</td><td>iConnectEFT Constant = P33_08_REQ_EMVH_CURRENT_PACKET_NBR
Current packet number.
Currently always '0'.</td></tr>
<tr><td>10</td><td>1</td><td>Alphanum</td><td>iConnectEFT Constant = P33_08_REQ_EMVH_PACKET_TYPE
Packet type.
Currently always '0' indicating that the packet is the first and last.</td></tr>
<tr><td colspan="4">Search for FS character to locate the next field.</td></tr>
</table>

Offset	Length	Format	Description
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	Variable	Alphanum	iConnectEFT Constant =P33_08_REQ_TYPE_AND_DATA Parameter type and data. The parameter type, which is the first byte sent, defines the data format. • F = Parameter type file located under SYSTEM disc. ◦ FileName = Name of the uploaded file. • H = Parameter type file located under HOST. ◦ FileName = Name of the uploaded file under HOST drive. If parameter '0091_0031' is set to '1' then configuration files cannot be loaded from HOST. Loading from HOST is only allowed if this parameter setting is '1'. 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.			
M	1	Constant	FS – 0x1C.
M + 1	1	Constant	ETX – 0x03.
M + 2	1	Binary	LRC check character.

8_3_9_3 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.
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. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 2) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note 2** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
EMV Request Header			

Offset	Length	Format	Description
9	1	Alphanum	iConnectEFT Constant = P33_08_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Always '0' as only one packet will be part of an EMV response message.
10	1	Alphanum	iConnectEFT Constant = P33_08_RES_EMVH_PACKET_TYPE Packet type. Currently, EMV response messages always return '0'.
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.

Offset	Length	Format	Description
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. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 1) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note 1** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
EMV Request Header			

Offset	Length	Format	Description
9	1	Alphanum	iConnectEFT Constant = P33_09_REQ_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.
10	1	Alphanum	iConnectEFT Constant = P33_09_REQ_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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	1	Constant	FS – 0x1C.

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.

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_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.

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_09_RES_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 2) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE ⓘ Failure codes can have values from 1 to 99. ⚠ **Note 2** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_09_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Currently, EMV response messages always return '0'
10	1	Alphanum	iConnectEFT Constant = P33_09_RES_EMVH_PACKET_TYPE Packet type. Currently, EMV response messages always return '0'.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.

Offset	Length	Format	Description
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 message to change currency to U.S. Dollars:

   '33.09.0000[FS]R[FS]T9F33:03:hE048C8[FS] T5F2A :02:h 0840 [FS]T5F36:01:h02'

   > Per ISO 4127 (International Standard for currency), the currency code for U.S. Dollars is '08 40'.

2. When the POS follows up with a '33.10.x' Get EMV Tag Data Message, the terminal will return the following message confirming the currency change to U.S. Dollars:

   '33.10.0000[FS] T5f2a :02:h 0840 [FS]T57:13:h6510000000000174D17122011000050600000F[FS]T95:05:h0000000000[FS] T9f33:03:hE048C8[FS]T5f36:01: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.

Offset	Length	Format	Description
1	3	Constant	iConnectEFT Constant = M33_10_EMV_GET_DATA Message identifier. • 33.
4	3	Constant	Subcommand identifier. • 10. = Get EMV Tag Data.
EMV Request Header			
7	2	Alphanum	iConnectEFT Constant = P33_10_REQ_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 1) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note 1** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).

Offset	Length	Format	Description
9	1	Alphanum	iConnectEFT Constant = P33_10_REQ_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.
10	1	Alphanum	iConnectEFT Constant = P33_10_REQ_EMVH_PACKET_TYPE Packet type. • 0 = Indicates that the packet is the first and last. • 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	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.

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] 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_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.

Offset	Length	Format	Description
7	2	Alphanum	iConnectEFT Constant = P33_10_RES_STATUS Status code. • 00 = OK • 01 = FAIL • 02 = INVALID_PACKET_NUMBER_FOR_TYPE (see Note 2) • 03 = INVALID_MSG_HEADER • 04 = UPDATE_LIST_FAIL • 05 = SUSPEND_LIST_FAIL • 06 = UPDATE_TIMER_FAIL • 07 = TAG_LIST_FAIL • 08 = INVALID_TARGET • 09 = INVALID_LOAD_SRC • 10 = INVALID_FILENAME • 11 = FILE_NOT_FOUND • 12 = INVALID_CMD_TYPE Failure codes can have values from 1 to 99. **Note 2** The packet number provided is inconsistent with the package type provided (e.g., the packet type indicates that this is the first and only packet, where the packet is actually the third packet).
EMV Request Header			
9	1	Alphanum	iConnectEFT Constant = P33_10_RES_EMVH_CURRENT_PACKET_NBR Current packet number. Currently, EMV response messages always return '0' as only one packet will be part of an EMV response message.
10	1	Alphanum	iConnectEFT Constant = P33_10_RES_EMVH_PACKET_TYPE Packet type. Currently, EMV response messages always return '0'.
Search for FS character to locate the next field.			
11	1	Constant	FS – 0x1C.

Offset	Length	Format	Description
12	Variable	Alphanum	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.
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.			
M	1	Constant	FS – 0x1C.
M + 1	1	Constant	ETX – 0x03.
M + 2	1	Binary	LRC check character.

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. The POS sends the following message to change currency to U.S. Dollars:

   '33.09.0000[FS]R[FS]T9F33:03:hE048C8[FS] T5F2A :02:h 0840 [FS]T5F36:01:h02'

   Per ISO 4127 (International Standard for currency), the currency code for U.S. Dollars is '08 40'.

2. When the POS follows up with the EMV '33.10.x' Get Tag Data Message, the terminal will return the following message confirming the currency change to U.S. Dollars:

   '33.10.0000[FS] T5f2a :02:h 0840 [FS]T57:13:h6510000000000174D17122011000050600000F[FS]T95:05:h0000000000[FS]T9f33:03:hE048C8[FS]T5f36:01:h02[FS]'

8_3_12 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.

Transaction Step List

Step	Step Name	Description

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 not required).
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. The flow must be suspended in order to enable synchronization.
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 using tag `T9F02`.
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 (tag `T9F02`). In this case, the RBA should be suspended and the EMV '33.09.x' Set Tag Data message can be used with command "T" (Tag data only) and corresponding tags.
L	Amount confirmation.	Non-EMV step to confirm the amount.

Step	Step Name	Description
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).
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 in progress.	This step is used when authorization is requested and in progress (used for POS information, suspend not required).
S	EMV stop.	Transaction has ended (used for POS information, suspend not required).

8_3_13 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"
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_14 EMV and Non-EMV Tags Transmitted in Host Interface Messages

The following table lists and describes EMV and non-EMV tags transmitted in the following 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 '33.07.x' Terminal Capabilities Message Flow

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'	'33.07.x'
T42	Issuer Identification Number.	Hex-ASCII					X
T4F	Application Identifier (AID).	Hex-ASCII	X	X		X	
T50	Application Label.	ASCII	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		
T82	Application Interchange Profile.	Hex-ASCII		X		X	
T84	Dedicated File (DF) Name.	Hex-ASCII		X		X	X
T8A	Authorization Response Code.	ASCII		X	X	X	
T8E	Card Holder Verification Method (CVM) List.	Hex-ASCII		X		X	
T91	Issuer Authentication Data.	Hex-ASCII		X	X	X	

Tag ID	Data Transmitted	Format	'33.02. x'	'33.03. x'	'33.04. x'	'33.05. x'	'33.07. 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				
T5F28	Issuer Country Code, 3 digit numeric.	Hex-ASCII	X				
T5F2A	Transaction Currency Code.	Hex-ASCII		X		X	
T5F2D	Preferred languages.	ASCII	X				
T5F30	Service Code.	Hex-ASCII	X				
T5F34	Application PAN Sequence Number.	Hex-ASCII	X	X		X	
T5F55	Issuer Country Code, 2 digit alpha.	ASCII	X				X
T5F56	Issuer Country Code, 3 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	

Tag ID	Data Transmitted	Format	'33.02. x'	'33.03. x'	'33.04. x'	'33.05. x'	'33.07. x'
T9F07	Application Usage Control.	Hex-ASCII	X			X	
T9F08	Application Version Number (ICC).	Hex-ASCII		X		X	
T9F09	Application Version Number (Terminal).	Hex-ASCII		X		X	
T9F0B	Cardholder Name Extended.	ASCII	X				
T9F0D	Issuer Action Code Default.	Hex-ASCII		X		X	
T9F0E	Issuer Action Code Denial.	Hex-ASCII		X		X	
T9F0F	Issuer Action Code Online.	Hex-ASCII		X		X	
T9F10	Issuer Application Data.	Hex-ASCII		X		X	
T9F11	Issuer Code Table Index	Hex-ASCII	X				
T9F12	Application Preferred Name.	ASCII	X	X		X	
T9F17	PIN Try Count.	Hex-ASCII		X		X	
T9F1A	Terminal Country Code.	Hex-ASCII	X	X		X	
T9F1E	Interface Device (IFD) Serial Number.	ASCII	X	X		X	X
T9F21	Transaction Time.	Hex-ASCII		X		X	
T9F26	Application Cryptogram (AC).	Hex-ASCII		X		X	

Tag ID	Data Transmitted	Format	'33.02. x'	'33.03. x'	'33.04. x'	'33.05. x'	'33.07. x'
T9F27	Cryptogram Information Data (CID).	Hex-ASCII		X		X	
T9F33	Terminal Capabilities.	Hex-ASCII		X		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			
T9F40	Additional Terminal Capabilities.	Hex-ASCII		X			
T9F41	Transaction Sequence Counter.	Hex-ASCII		X		X	
T9F53	Transaction Category Code (VISA only).	Hex-ASCII		X		X	
T9F5B	Issuer Script 1 results.	Hex-ASCII		X		X	
T9F6E	Third Part Data.	Hex-ASCII				X	
TDF03	Terminal Action Code Default.	Hex-ASCII		X		X	
TDF04	Terminal Action Code Denial.	Hex-ASCII		X		X	

Tag ID	Data Transmitted	Format	'33.02. x'	'33.03. x'	'33.04. x'	'33.05. x'	'33.07. x'
TDF05	Terminal Action Code Online.	Hex-ASCII		X		X	
D1000	Account Type (Interac only).	ASCII		X		X	
D1001	PIN Entry Required Flag.	ASCII		X		X	
D1002	Signature required Flag.	ASCII		X		X	
D1003	Confirmation response Code.	ASCII		X		X	
D1004	Host response available.	Hex-ASCII			X		
D1005	Transaction Type.	ASCII		X		X	
D1006	Authorization request MAC (Interac only).	ASCII					
D1007	Authorization response MAC (Interac only).	ASCII			X		
D1008	Authorization response Terminal Serial Number (Interac only).	ASCII			X		
D1009	Authorization response Base 24 response code (Interac only).	Hex-ASCII			X		
D100A	Authorization response approval code (Interac only).	Hex-ASCII			X		
D100B	Authorization response retrieval code (Interac only).	Hex-ASCII			X		
D100C	Authorization response MAC session key (Interac only).	Hex-ASCII			X		
D100D	Authorization response PIN session key (Interac only).	Hex-ASCII			X		
D100E	Selected Transaction Language.	ASCII	X	X		X	
D100F	PIN Entry Success Flag.	ASCII		X		X	

Tag ID	Data Transmitted	Format	'33.02.x'	'33.03.x'	'33.04.x'	'33.05.x'	'33.07.x'
D1010	Error Response Code.	ASCII		X		X	
D1016	U.S. Common AID Flag.						X

The above list only represents a partial list. A complete list of tags cannot be provided here as it is dependent on the host and issuer data required by the card and application. Further information on EMV tag definitions can be found at this site: http://www.emvlab.org/emvtags/all/

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.

It should be noted that every time the application enters the online idle state it immediately configures the next transaction to be an EMV purchase transaction if EMV is enabled in the configuration file. So when a chip card is entered into the terminal, the purchase transaction immediately starts. The following transaction types are discussed in this section:

- 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
- MSD Contactless Transaction Flow

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. 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. The terminal then sends a 10.x Hard Reset message to the POS indicating that it is returning to the online idle state in readiness for the next transaction.

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		→

Sequence	Message	POS	Terminal
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.	13.x: Amount Message		
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, followed by a hard reset to return the terminal to the online idle state.	EMV '33.05.x' Authorization Confirmation Response Message		
	10.x: Hard Reset Message		
Return to idle state.			

8_4_2 EMV Contactless Transaction Flow

The POS sends two messages to the terminal at the start of a purchase or refund transaction. The first message is a 14.x Transaction Type message which identifies the transaction as a purchase ('14.01') or refund ('14.03'). This message is followed by a 13.x: Amount Message which specifies the amount of the purchase or refund. If the 14.x message identifies the transaction as a purchase and the 13.x message contains a negative number, then the transaction will be processed as a refund.

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.

Once 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. This message 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 has been received by the POS, it responds with an EMV '33.04.x' Authorization Response message which contains encrypted data that is used by the embedded microchip in the card to approve or decline the transaction. Once the decision has been made, an EMV '33.05.x' Authorization Confirmation Response message is sent to the POS to convey this decision. This message includes the necessary tag information used by the POS to print the transaction receipt. The terminal then returns to the idle state and sends a 10.x Hard Reset message to the POS.

> In order 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.

Refer to the below table for an illustration of the EMV contactless transaction sequence.

EMV Contactless 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		
EMV transaction is initiated as 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, followed by a hard reset to return the terminal to the online idle state.	EMV '33.05.x' Authorization Confirmation Response Message		
	10.x: Hard Reset 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. At this time, only Interac cards meet this criteria. Based on the EMV Application IDs loaded in the terminal, this may change.

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.

The transaction amount must be sent using the 13.x message at any time to be able to continue the transaction. 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.

With the final transaction information received from the POS, the cardholder will be prompted to confirm the refund amount. The cardholder will 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 to the POS, which will return the EMV '33.04.x' Authorization Response message. Included in the '33.04.x' message will be tag D1003 which will indicate the approval status ("A" for Approved and "D" for Declined). The terminal will then display the transaction status as "Approved" or "Declined" based on this tag value. An EMV '33.05.x' Authorization Confirmation message is then returned to the POS confirming the refund transaction state and including the necessary tag information for printing. 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 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		
The cardholder is prompted to confirm the refund amount, 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.			
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.

The transaction amount must be sent using the 13.x message at any time to be able to continue the transaction. 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.

With the final transaction information received from the POS, the cardholder will be prompted to confirm the refund amount.

An EMV '33.05.x' Authorization Confirmation 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 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		→
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		←
The cardholder is prompted to confirm the refund amount. <Yes>, <No>, and <Cancel> buttons are displayed.			
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.

The transaction amount must be sent using the 13.x message at any time to be able to continue the transaction. 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.

With the final transaction information received from the POS, the cardholder will be prompted to confirm the refund amount.

An EMV '33.05.x' Authorization Confirmation 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	→	
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		←
The cardholder is prompted to confirm the refund amount. <Yes>, <No>, and <Cancel> buttons are displayed.			
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 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.

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_7 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 are 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. Additionally, non-EMV tag numbers are intentionally non-compliant with EMV standards for tag numbers in order to avoid conflict with EMV tags in the future.

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. As an example, 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' indicating error or incompletion, tag `D1010` is provided in the confirmation response to indicate the type of error which as occurred (e.g., Authorization Request Sent Failed, Card Data Invalid). As another example, tag `D1005` is issued in both the authorization request and the confirmation response to indicate which type of transaction (purchase or refund) is requesting authorization.

The table below gives definitions for 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 = Checquing. • 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).
D1004	ECR Response Available	Tag data to indicate if the ECR 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 ECR 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.
D1006	Authorization Request MAC Value	Authorization Request MAC value provided to ECR by the terminal. • MAC value is typically 8 bytes in hex ASCII format. For Interac transactions only. This value is provided to the ECR by the terminal during an Interac debit authorization request.

Tag Number	Tag Name	Non-EMV Tag Definition and Values
D1007	Authorization Response MAC Value	Authorization Response MAC value provided to terminal by ECR. • MAC value is typically 8 bytes in hex ASCII format. For Interac transactions only. The MAC is typically 8 bytes hex ASCII, and is provided to the terminal by the ECR during an Interac debit authorization response.
D1008	Authorization Response Terminal Serial Number	Serial number of the terminal the Authorization Response is being sent to. • Serial number is 8 bytes in ASCII format (e.g., "AU000006").
D1009	Authorization Response Base 24 Response Code	Base 24 response code returned in the authorization response message. • Response codes is 3 bytes for EMV transactions.
D100A	Authorization Response Approval Code	Response approval code in the authorization response message. • Response approval code is 8 bytes for EMV transactions.
D100B	Authorization Response Retrieval Reference Number	Retrieval reference number in authorization response message. • Retrieval reference number is 10 bytes for EMV transactions.
D100C	Authorization Response MAC Session Key	MAC session key returned in the authorization response message. • MAC session key is 16 bytes in hex ASCII format.
D100D	Authorization Response PIN Session Key	PIN session key returned in the authorization response message. • PIN session key is 16 bytes in hex ASCII format.

Tag Number	Tag Name	Non-EMV Tag Definition and Values
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.
D100F	Pin Entry Successful Flag	EMV CV indication of PIN entry success in the current transaction. • 0 = Not successful. • 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>ARRT</td><td>Authorization Response Received Timeout – t he 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/Application 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 – t here 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 would not actually 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>
<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 • 10 = Partial Authorization • 01 = Voice Referral Currently there are no other values, and for all other cases only the T8A tag value on its own for approval or decline etc. is provided.
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_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 EMV with P2PE Enabled

8_6_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 will be used in place of the 50.x Authorization Request message. See EMV Tag Encryption for handling of the tags which should be included in this message.

Additionally, the following new Ingenico-specific tags will be added:

- TFF1D
- TFF1E
- TFF1F
- TFF20
- TFF21

The specific tags that are provided will depend on the card data and encryption type. As an example, tag `TFF20` will only be present 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_6_2 Encryption of 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 MSR Encryption .

8_6_3 EMV Tag Encryption

The below table shows how certain tags are handled during encryption for an EMV transaction.

8_6_3_1 EMV Tag Handling During Encryption

When E2EE encryption is enabled, the following applies 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.
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.

Currently, there are multiple tags which return contact and contactless Track 2 Equivalent Data (tag `T57`), PAN (tag `T5A`), and Expiry Date (tag `T5F24`). If E2EE encryption is enabled and the BIN is not present in the validated BIN table, then the Track 2 tag data will be masked.

Ingenico-specific EMV tags are described in the following table.

Ingenico-specific EMV Tags

Tag	Content	Description
TFF1D	PAN	PAN associated with the card (encrypted)
TFF1E	Track 1 Data	Track 1 data (encrypted)
TFF1F	Track 2 Data	Track 2 data (encrypted)
TFF20	ETB	ETB (Voltage only)
TFF21	Track 3 Data	Track 3 data (encrypted)

The below tags vary by encryption type.

Tags Encrypted by Type

	TFF1D (PAN)	TFF1E (Track 1)	TFF1F (Track 2)	TFF20 (ETB)	TFF21 (Track 3)
On-Guard	N/A	N/A	Encrypted Track 1 and Track 2	N/A	N/A
Voltage TEP1 /TEP2	Encrypted PAN	Encrypted Track 1	Encrypted Track 2	ETB	N/A
RSA-OAEP	N/A	N/A	Base64-encoded encrypted data (344 bytes).	N/A	N/A
TransArmor	N/A	N/A	N/A	N/A	Encrypted Track 3 (see Note below)
S1	N/A	N/A	N/A	N/A	Encrypted Track 3
TDES	N/A	N/A	N/A	N/A	Encrypted Track 3
EPS	N/A	Encrypted Track 1	Encrypted Track 2	N/A	N/A
Mercury	N/A	N/A	N/A	N/A	Encrypted Track 3
Magtek	Encrypted PAN	Encrypted Track 1	Encrypted Track 2	N/A	N/A
Monetra	N/A	N/A	N/A	N/A	Encrypted Track 3

TransArmor's Track 3 consists of three items separated by colons:

- The 344-byte Base64 string
- One digit indicating which data were encrypted (1 = Track 1, 2 = Track 2, 3 = PAN for manually-entered data)
- KEY ID from the security.dat file

8_7 EMV Configuration Parameters

The following sections provide more detailed information on parameters and tags used in EMV transactions:

- Notes on EMV Configuration Parameters
- EMV.DAT Configuration File
- Application ID (AID) Parameters in EMVCONTACT.XML
- Application ID (AID) Parameters in EMVCLESS.XML
- Certificate Authority Public Keys in EMVCONTACT.XML
- ICS Tags in EMVCONTACT.XML
- ICS Tags in EMVCLESS.XML

8_7_1 Notes on EMV Configuration Parameters

8_7_1_1 Overview

Some of the EMV configuration parameters listed are reserved for future use. Additionally, not all parameters are implemented in any given software release. When in doubt, use the default settings in the sample files provided with the software.

8_7_1_2 Merchant and Acquirer Responsibilities and Parameter Management

Please be aware that 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 provides and uses sample configuration files. Once 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 (sometimes referred to as their “payment system” below). 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 (Visa, MasterCard, etc.) and to determine and set EMV parameters that are appropriate for their business model.

8_7_1_3 Configuration Files

For flexibility, Ingenico Retail applications separate the configuration for contact EMV and contactless EMV payment into two different files; `EMVCONTACT.XML` and `EMVCLESS.XML`.

Defaults:

- Parameters which 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. If these parameters are not found in the `EMVCLESS.XML` file, the application will check for these in the `EMVCONTACT.XML` file.
- Within a given XML file, some parameters must be defined for each individual Application ID. Many parameters, however, can be defined in the ICS section and will default to the ICS values unless otherwise specified for a specific 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 provided in the following sections indicate which parameters are assigned an ICS Default value (refer to the 2nd column in each table).
    - Application ID (AID) Parameters in EMVCLESS.XML
    - Application ID (AID) Parameters in EMVCONTACT.XML
    - ICS Tags in EMVCLESS.XML
    - ICS Tags in EMVCONTACT.XML

8_7_1_4 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**. Please make sure the parameters are set appropriately whenever the Contactless reader will be used.

8_7_1_5 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 is dependent on the tag itself. For a list of the different formats, please refer to EMV Book 3, “Data Element Format Conventions.” The following table describes the main formats used in this document. The descriptions are as described in EMV Book 3.

Data Element Formats

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).
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, since different specifications may use different bit-numbering conventions; that is, 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 sometimes refer to this data format as Binary Coded Decimal (“BCD”) or unsigned packed. **Example**: Amount, Authorised (Numeric) is defined as “n 12” with a length of six bytes. A value of 12345 is stored in Amount, Authorised (Numeric) as Hex '00 00 00 01 23 45'.

8_7_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. Example: '56 49 53 41' = "VISA"	Hexadecimal ASCII characters
T9F06	No	Application ID (AID). This is 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 7 bytes. ' 07 A0 00 00 00 03 10 10'	Hexadecimal (5 - 16 Bytes)

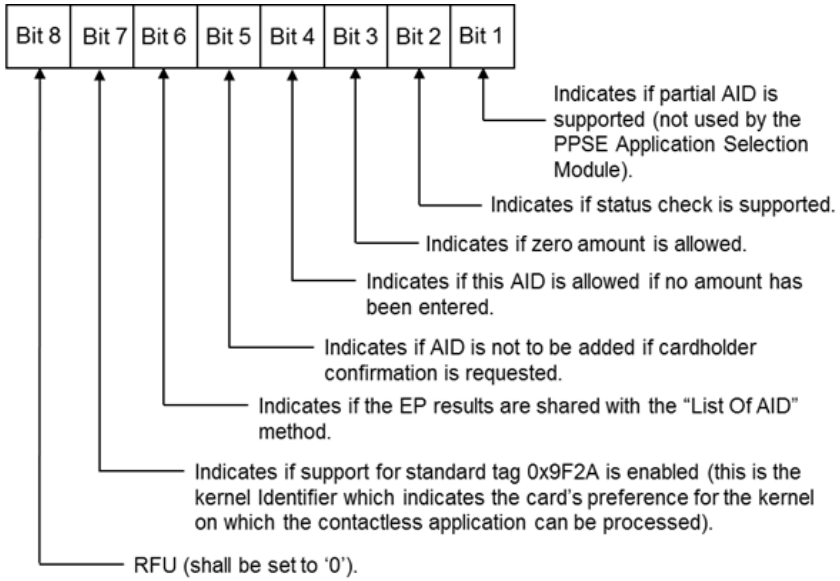
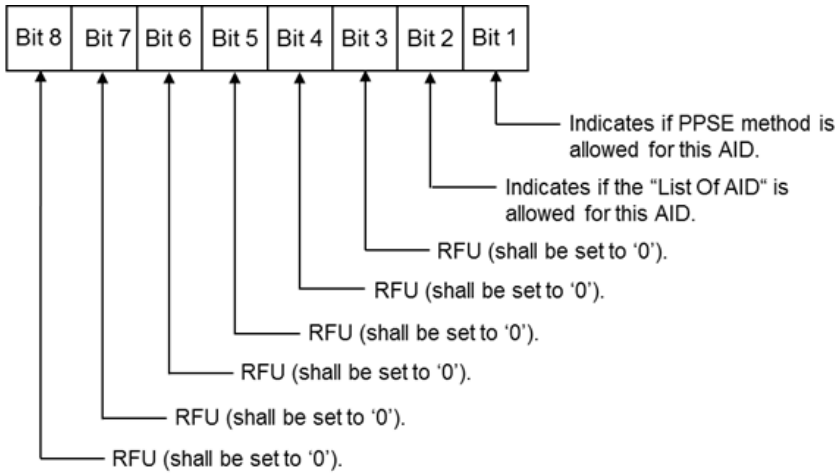
Parameter	ICS Default	Description	Format
T9F1A	Yes	Country code per ISO 3166 (International Standard for Country Codes). Country codes may be referenced at http://en.wikipedia.org/wiki/ISO_3166-1_numeric. For example: '01 24' = Canada '08 40' = United States	EMV Format "n 3" (2 Bytes)
T5F2A	Yes	Currency code per ISO 4217 (International Standard for currency). Currency codes may be referenced at http://en.wikipedia.org/wiki/ISO_4217. As examples: '08 26' = United Kingdom pound sterling. '08 40' = U.S. dollar. '01 24' = Canadian 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 in accordance with ISO 4217. For example: '02' indicates that here are two digits to the right of the decimal point.	EMV Format "n 1" (1 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 an explanation of this parameter. As an example: '00 00 03 E8' (hexadecimal) = '1000' (decimal) = $10.00 in U.S currency.	EMV Format b (Binary) (4 Bytes)
T9F8124	Yes	Default Dynamic Data Authentication Data Object List (DDOL). This is specified by the payment system and is to be used when a DDOL is not present on the EMV card. As an example: '9F 37 04'	EMV Format b (Binary) (Variable)
T9F8125	Yes	Default Transaction Certificate Data Object List (TDOL). This is specified by the payment system and is to be used when a TDOL is not present on the EMV card. As an example: '9F 02 06 95 05 5F 2A 02 9A 03 9C 01 9F 37 04'	EMV Format b (Binary) (Variable)
T9F8126	Yes	Maximum target percentage to be used for biased random selection. The value for this parameter must be within the range 0 to 99 and no less than the value of tag T9F8127. Refer to EMV Book 3, "Random Transaction Selection," for an explanation of this parameter. As an example: '32' (hexadecimal) = '50' (decimal) = 50%	EMV Format b (Binary) (1 Byte)

Parameter	ICS Default	Description	Format
T9F8127	Yes	Target Percentage to be Used for Random Selection. The value for this parameter must be within the range 0 to 99. Refer to EMV Book 3, "Random Transaction Selection," for an explanation of this parameter. As an example: '32' (hexadecimal) = '50' (decimal) = 50%	EMV Format b (Binary) (1 Byte)
T9F8128	Yes	Terminal Action Code (TAC) - Default. Specified by the acquirer. Refer to EMV Book 3, "Terminal Action Analysis," for an explanation of this parameter. As an example: 'DC 40 00 A8 00'	EMV Format b (Binary) (5 Bytes)
T9F8129	Yes	Terminal Action Code (TAC) - Denial. Specified by the acquirer. Refer to EMV Book 3, "Terminal Action Analysis," for an explanation of this parameter. As an example: '00 10 00 00 00'	EMV Format b (Binary) (5 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. As an example: 'DC 40 04 F8 00'	EMV Format b (Binary) (5 Bytes)
T9F09	Yes	Version Number assigned by the payment system for the application. As an example: '00 8C'	EMV Format b (Binary) (2 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 an explanation of this parameter. As an example: '00 00 27 10' (hexadecimal) = '10000' (decimal) = $100.00 in US currency.	EMV Format b (Binary) (4 Bytes)
T9F841D	Yes	Allow Partial Name Selection (Application Selection Indicator). For an application on the EMV card to be supported by an application in the terminal, this parameter indicates whether the associated Application ID in the terminal must exactly match the Application ID in the card, including the length of the Application ID, or only up to the length of the Application ID in the terminal. As an example: '01'	EMV Format b (Binary) (1 Byte)

8_7_3 Application ID (AID) Parameters in EMVCLESS.XML

Application ID Parameters in EMVCLESS.XML

Parameter	ICS Default	Description	Format
T9F06	No	Application ID used to match the AID configured on the EMV card. As an example: 'A0 00 00 00 03 10 10'	Hexadecimal (Variable)
T9F33	Yes	Terminal Capabilities. This indicates the card dat input, CVM, and security capabilities of the terminal. Refer to EMV Book 4, "Terminal Capabilities." 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, CDA.	EMV Format "b" (Binary) (3 Bytes)
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 Zip (magstripe only). • '01 03' = Interac. • '01 04' = Google Wallet.	(2 Bytes)

Parameter	ICS Default	Description	Format
T9F928100	No	Proximity Payment System Environment (PPSE) Application Selection, which indicates the Application ID options: **Byte 1**:  **Byte 2**:  **Byte 3 and Byte 4**: RFU (shall be set to '0'). As an example: '05 03 00 00' indicates Partial AID supported, zero amount allowed, PPSE and List Of AID methods allowed.	(4 Bytes)

Parameter	ICS Default	Description	Format
T9F92810D	No	Transaction Limit. As an example: '00 00 00 00 20 00' = $20.00	EMV Format "n 12" (6 Bytes)
T9F92810E	No	CVM Required Limit. As an example: '00 00 00 00 20 00' = $20.00	EMV Format "n 12" (6 Bytes)
T9F92810F	No	Contactless Floor Limit. As an example: '00 00 00 00 20 00' = $20.00	EMV Format "n 12" (6 Bytes)

Parameter	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). ⚠ 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.	EMV Format "b" (Binary) (4 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. As an example: '00 00 27 10' (hexadecimal) = '10000' (decimal) = $100.00 in U.S. currency.	EMV Format "b" (Binary) (4 Bytes)

Parameter	ICS Default	Description	Format
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)
T9F91841D	No	Terminal Supported Languages. As an example: '65 6e 65 73' = "enes" where '65 6e' = "en" (English) and '65 73' = "es" (Espanol, or Spanish).	Hex ASCII (2 - 8 Bytes)
T9FA15D	No	EMV Scheme Label. As an 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.	
T9F91850D	No	List of Application Version Numbers (AVNs) for PayPass Magstripe (PayPass only). As an 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). As an example: '01 05 02 00 02 05' indicates versions 0105, 0200, and 0205.	(Variable)
T9F918504	No	PayPass Terminal Capabilities with CVM Required (PayPass only). Refer to the preceding description for `T9F33`. As an 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`. As an example: 'E0 68 C8' indicates No CVM.	
T9F918502	No	PayPass Default UDOL (PayPass only). As an example: '9F 6A 04' - this fixed value is defined in the PayPass specification.	

Parameter	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. As an example, '01' indicates that Magstripe profile is allowed for this AID.	
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. As an 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.	
T9F918709	No	Terminal Action Code -Default. This 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. Refer to EMV Book 3, "Terminal Action Analysis," for an explanation of this parameter. As an example: 'FC 50 0C 88 00'	EMV Format "b" (Binary) (5 Bytes)
T9F91870A	No	Terminal Action Code - Denial. This specifies the acquirer's 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. As an example: '00 00 00 00 00'	EMV Format "b" (Binary) (5 Bytes)
T9F91870B	No	Terminal Action Code - Online. This specifies the acquirer's conditions which result in a transaction being transmitted online. Refer to EMV Book 3, "Terminal Action Analysis," for an explanation of this parameter. As an example: 'FC 50 0C 88 00'	EMV Format "b" (Binary) (5 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. As an example: '52' = "R"	EMV Format "an" (1 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. As an example: '9F 08 02'	EMV Format "b" (Binary) (Variable)

Parameter	ICS Default	Description	Format
T9F918A11	No	List of Application Version Numbers (AVNs) for Interac (Interac only). As an example: '00 02' indicates version 0002.	
T9F918A04	No	Interac Terminal Capabilities with CVM Required (Interac only). Refer to the preceding description for `T9F33`. As an example: 'E0 08 00' indicates No CVM only, no CDA.	
T9F918A05	No	Interac Terminal Capabilities with No CVM Required (Interac only). Refer to the preceding description for `T9F33`. As an example: 'E0 08 00' indicates No CVM only, no CDA.	
T9F58	No	Interac Merchant Type Indicator (Interac only). Must be in the range from '01' to '05'. As an example: '03'	
T9F59	No	Interac Terminal Transaction Information (Interac only). As an example: 'C0 80 00'	
T9F5A	No	Interac Terminal Transaction Type (Interac only). As examples: '00' indicates a purchase. '01' indicates a refund.	
T9F5D	No	Interac Receipt Limit (Interac only). As an example: '00 00 00 00 00 00'	EMV Format "n 12"
T9F5E	No	Interac Terminal Option Status (Interac only). As an example: '00 00'	
T9F918200	No	ExpressPay Unpredictable Number Range (ExpressPay only). As an example: '03 0C' = '60' (decimal).	EMV Format "b" (Binary) (2 Bytes)
T9F91820A	No	List of Application Version Numbers (AVNs) for ExpressPay (ExpressPay only). As an example: '00 01' indicates version 0001.	

Parameter	ICS Default	Description	Format
T9F91820F	No	ExpressPay Full Online EMV Removal Timeout (ExpressPay only). As an example: '00 00 27 10' = '2710' (hex) = '10000' (decimal) = 10 seconds.	EMV Format "b" (Binary) (4 Bytes)
T9F6D	No	ExpressPay Terminal Capabilities (ExpressPay only). • '00' = ExpressPay 1.0 • '40' = ExpressPay 2.0 Magstripe Only • '80' = ExpressPay 2.0 EMV and Magstripe If ExpressPay EMV functionality will not be used in a given installation,the setting of '80' should not be used in order to avoid any attempts to interact with a card or mobile terminal while in EMV mode.	
T9F918808	No	Couponing mode (Google Wallet only). • '00' = No couponing. • '01' = Couponing only. • '02' = Couponing prior to transaction. • '03' = Couponing after transaction.	(1 Byte)

8_7_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
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	

8_7_5 EMV.DAT Configuration File

emv.dat Parameters

Parameter Name	DFS Index	Default Value	Description
EMV Support	0019_0001	1	• 0 = Register does not support EMV. • 1 = Register does support EMV.
Language Auto-Select	0019_0002	0	• 0 = No auto-select language during an EMV transaction. • 1 = Auto-select language during an EMV transaction.
Application Auto-Select	0019_0003	0	• 0 = No application auto-select during an EMV transaction. • 1 = Application auto-select during an EMV transaction.
Asynchronous	0019_0004	0	• 0 = Do not asynchronously transmit the status message to the register on change of status. • 1 = Asynchronously transmit the status message to the register on change of status.
EMV Test Environment	0019_0005	0	• 0 = Load only the EMV production environment. • 1 = Load the EMV production environment and the test environment.
Wait after Confirmation	0019_0006	0	• 0 = No wait after confirmation message assumed in EMV transaction. This is as originally implemented in the CA implementation. • 1 = Wait after authorization confirmation message is sent to register. The terminal typically pauses for signature request if not already received after confirmation message.
Interac Application Selection	0019_0007	0	• 0 = Interac application selection is disabled. • 1 = Interac application selection is enabled.
Domestic VISA	0019_0008	0	• 0 = Domestic VISA Debit application selection is disabled. • 1 = Domestic VISA Debit application selection is enabled.
Allow CVM Modification by POS	0019_0009	0	• 0 = Do not allow CVM modification by POS. • 1 = Allow CVM modification by POS.

8_7_6 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
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)
T9F8441	Yes	Support Multiple Languages. • '00' = No. • '01' = Yes. As an example: '01' configures to support Multiple Languages.	(1 Byte)

Tag	ICS Default	Description	Format
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)
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)

Tag	ICS Default	Description	Format
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)
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)

Tag	ICS Default	Description	Format
T9F844E	Yes	Support POS Entry Mode. • '00' = No. • '01' = Yes. As an example: '01' configures to support POS Entry Mode.	(1 Byte)
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)
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)

Tag	ICS Default	Description	Format
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)
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)

Tag	ICS Default	Description	Format
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)
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.) • '00' = Yes. • '01' = No. As an example: '00' configures to Request CDA for ARQC.	(1 Byte)
T9F845E	Yes	Request CDA for TC in Second GENERATE AC. • '00' = Yes. • '01' = No. 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)

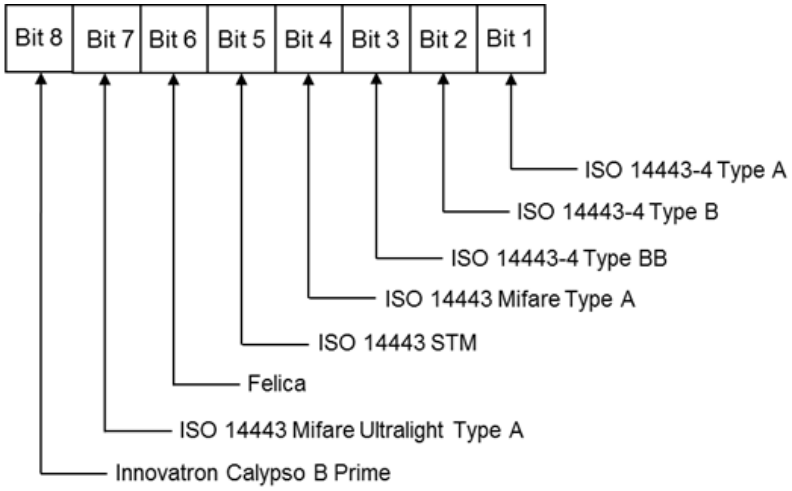
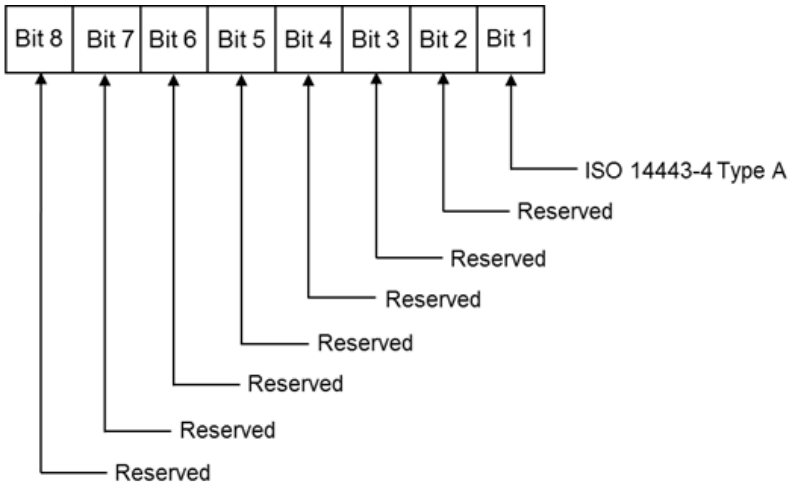
Tag	ICS Default	Description	Format
T9F8460	Yes	When Selecting to Bypass a PIN Method, all Other PIN Methods are Considered Bypassed. • '00' = No. • '01' = Yes. As an example: '01' configures for all Other PIN Methods are Considered Bypassed when Selecting to Bypass a PIN Method	(1 Byte)

8_7_7 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

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**:  **Byte 1**:  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)
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)

Tag	ICS Default	Description	Format
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_8 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.

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_8_1 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.
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	MAC data (data length less than 200 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	Indicates the MAC master key index.
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	MAC data (data length less than 200 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.

8_8_2 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.
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	MAC data (data length less than 200 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.
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	MAC data (data length less than 200 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.

8_9 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 EMV Configuration and Flow

8_10_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_10_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_10_3 Process Online PIN

Extract the PIN Block from the T99 tag included in the EMV '33.03.x' Authorization Request Message. This PIN Block is DUKPT encrypted and is padded with "F"s if required. Refer to the following example 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 per the regular EMV flow.

Refer to PIN Block Tag Format in Authorization Request Message for more information on the PIN Block tag format.

8_10_4 PIN Block Tag Format in Authorization Request Message

The T99 tag included in the EMV '33.03.x' Authorization Request Message provides the 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	DUKPT Encrypted PIN Block.
24	20	Alphanum	KSN.

8_11 EMV Full and Partial Refunds

8_11_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. As the refund transaction is processed, the Authorization Response Code (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, the Authorization Response Code will be set as "Offline Declined" and the cryptogram type will be AAC (Application Authentication Cryptogram).

8_11_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.
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). For EMV partial refund transactions, these are not host approved and tag D1003 will instead have a value of "C" indicated completion.

Refer to the following **transaction flow** sections for more details:

- EMV Full Refund Transaction Flow
- EMV Partial Refund Normal Transaction Flow
- EMV Partial Refund On-Demand Transaction Flow

8_11_3 Cancelling a Refund Transaction

The EMV '33.09.x' Set Tag Data Message can be used to cancel an EMV refund 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 '33.05.x' Authorization Confirmation Response Message. The following transaction flow illustrates a transaction cancelled using the 33.09.x 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		←
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 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_12 EMV Cashback

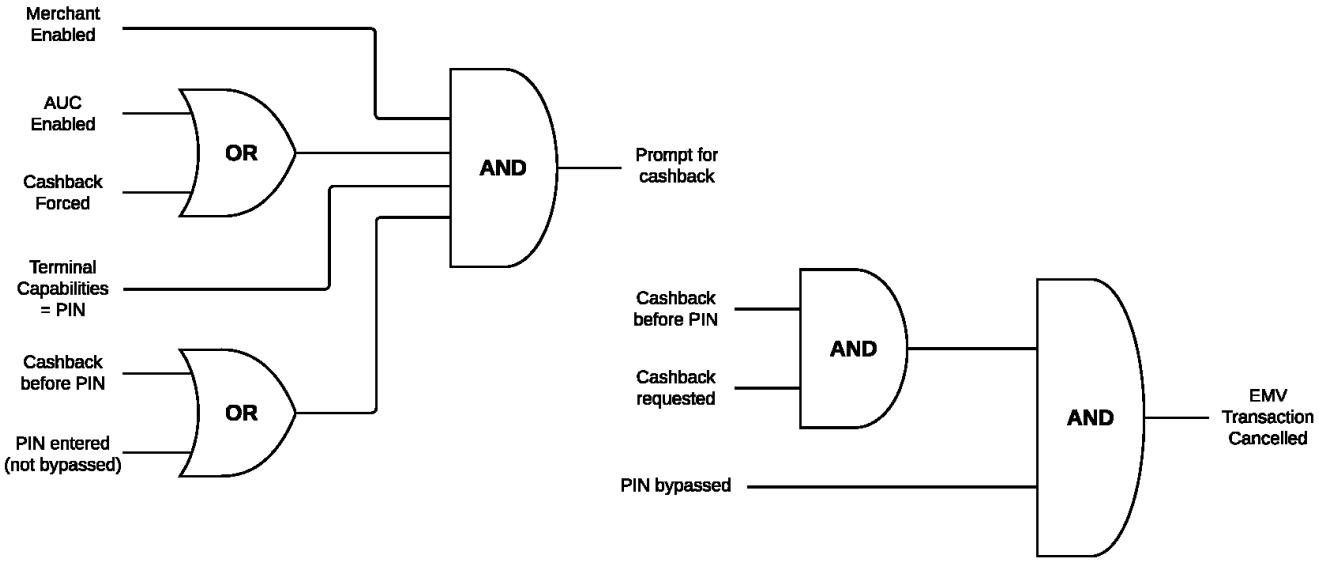
8_12_1 Overview

The following four conditions must be met in order for the terminal to prompt the cardholder for cashback:

- Cashback must be enabled by the merchant.
  - 04.x: Set Payment Type Request '0011_00xx'::'Cashback Limit' configured (set to 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.
- Cashback flow is configured for cashback before PIN entry ('0002_0010' = '0') OR the PIN is entered (not bypassed).

Refer to the following diagram which illustrates this process.



RBA EMV Cashback Prompt Logic

8_12_2 Cashback/Total Amount Confirmation

The cashback/total amount confirmation for EMV transactions will be displayed using one of the following forms:

- `$<Purchase amount> Please confirm`
  - This form should be displayed to confirm the total purchase amount only before PIN entry if and only if:
    - `NOT(CB>PIN AND CBReqd) AND AmtConf`
- `$<Purchase amount> + $<Cashback amount> = $<Total amount> Please confirm`
  - This form should be displayed to confirm the cashback/total amounts before PIN entry if and only if:
    - `NOT(CB>PIN AND CBReqd) AND (CBConf OR AmtConf)`
  - This form should be displayed to confirm the cashback/total amounts after PIN entry if and 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_12_3 Sample Messages

For example, a '13.3000[GS]2000' message signifies:

- 3000 = $30.00 total purchase amount including excluding $20.00 cashback
- 2000 = $20.00 cashback amount

When the above 13.x message is sent, the below 29.x messages will 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

> Th cashback amount, variable `305`'s value, must be less than the value of maximum cashback, variable `306`.

8_12_4 Terminal and ECR Notification

During on-demand EMV transactions, the ECR should inform the terminal of the selected cashback via a '13.<purchase amount>[GS]<cashback amount>' message.

The following table describes the process for notifying the ECR of the selected cashback.

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 cash back amount variable `305`).
Cashback before PIN Entry - On-Demand	• The ECR prompts for and handles the cashback amount and knows the cashback input.

Cashback after PIN entry - RBA Flow	• `'29.305'` (use 29.x: Get Variable Request message to retrieve cash back amount variable `305`).
	• Cashback amount = T9F02 (Authorized Amount) - final 13.x: Amount Message where tag T9F02 is included in on 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 ECR prompts for and handles the cashback amount and knows the cashback input.

9_Appendices

9_1 Appendix A. Differences Between U32 RBA and Telium RBA

This Appendix lists differences between the latest U32 and Telium versions of the RBA Integration Kit.

- A.1. At A Glance
- A.2. Support
- A.3. Signing Requirements
- A.4. Transaction Flows
- A.5. Host Interface Messages
- A.6. Variable Names
- A.7. Forms
- A.8. Prompts
- A.9. RBA Configuration (config.dfs)
- A.10. Format Specifiers

9_1_1 A.1. At A Glance

The following table lists key differences between the U32 RBA and the Telium RBA implementations:

Telium and U32 Key Differences

Feature	U32 RBA	Telium RBA
Form format	`*.icg`	`*.K3Z` with Javascript and JSON
Color displays	On color terminals, up to 256 colors	On color terminals, 24-bit color
Supported image formats	`*.bmp`, `*.gif`	`*.bmp`, `*.gif`, `*.jpg`, `*.png`
Supported video formats	No video support	`*.mp4`
Supported signature formats (for signature capture terminals)	3-byte ASCII format, 4-byte RAW format	3-byte ASCII format only
Button ID naming convention	Characters	Descriptive names
EFT file format	U32 EFT format	Telium EFT format

Since some `config.dfs` settings have been enhanced, deleted, or added for Telium RBA, use care when transferring settings from a U32 RBA implementation to a new Telium RBA implementation.

9_1_2 A.2. Support

See Introduction for detailed information on the types of features, terminals, and POS environments supported by the Telium implementation of RBA.

9_1_2_1 Features

The Telium implementation of RBA has added support for the following features:

- Internal/External BIN Range Checking
- Contactless Card Reader

The Telium implementation of RBA has removed support for the following features:

- Signia
- Active Card Management
- Smart Card
- Check Conversion

9_1_2_2 Terminals

The following Ingenico terminals support the Telium RBA implementation:

- iCMP (also referred to as iCM122)

- iSMP (also referred to as iMP350)
- iSMPc (also referred to as iMP352)
- iPP320
- iPP350
- iSC250 and iSC Touch 250
- iSC350
- iSC Touch 480
- iUN2xx (includes iUP250 and iUR250 and/or iUC150)
- iWL250

9_1_2_3 POS Environments

The Telium RBA implementation supports connectivity with the following types of systems:

- NCR Register / DOS environment
- IBM 4680/4690 Register environment
- IBM 4694 Register / Windows NT environment
- Windows XP and Windows 7
- Mac OS 10.6 (Snow Leopard)
- POS for .NET

9_1_3 A.3. Signing Requirements

All applications, data files, images, videos, form files, and prompt files EXCEPT the `PROMPT.xml` files must be signed by Ingenico before they can be loaded to a production terminal.

9_1_4 A.4. Transaction Flows

The Telium RBA implementation has incorporated the following changes to RBA's transaction flows:

- Updated RBA standard process flow (see Standard Process Flow for more information).
- Updated rules around execution of on-demand messages (see On-Demand for more information).
- Updated Cash Back transaction flow (see Cash Back for more information).
- Updated typical transaction flow for retrieving a signature using `GetVariable` (see Retrieval Using Get Variable for more information).
- Updated RBA data message format (see Data Message Format for more information).
- Updated RBA general message flow (see General Message Flow for more information).
- Added 31.x to on-demand message list (see Communication Messages for more information).

See Terminal Process Flow and Functions for detailed information on Telium RBA terminal transaction flows and functions.

9_1_5 A.5. Host Interface Messages

The following table lists differences between the U32 and Telium implementations of RBA host interface messages:

RBA Host Interface Message Handling

Message	Differences
01.x Online Message	Changed response for offset 4 to '0000.' • Changed response for offset 8 to '0000.' • Added at offset 12: ASCII control character – ETX • Added at offset 13: LRC Check Character See 01.x: Online Message for more information.
04.x Set Payment Type Request	Removed support for old format. • Referenced Payment Types are now defined in the `cards.dat` portion of `config.dfs`.
07.x Unit Data Request	Removed IBMEFT download support. • Response returns '0000' at U+1 and V+1 offsets in place of EFTL and EFTP version numbers. • Allowable Terminal ID values in response (at offset 11) updated to currently supported Telium terminals.
08.x Health Stat	• Added new message.
09.x Set Allowed Payments Message	• Payment types are now defined in `cards.dat` portion of `config.dfs`.
10.x Hard Reset Message	• Removed support for legacy format.
13.x Amount Message	Added limitations on number of amount fields allowed in message: minimum of 1 and maximum of 9 • Removed 13.x message single amount format • Added requirement: EBT Food transactions use index 2 of the amount field.

Message	Differences
19.x BIN Lookup	Removed allowable value from offset 4 in request format: N – No account data • Changed offset 6 in request format, is now set to 1 if account number is received from 12.x. • Changed offset M, M+1, N, and N+1 in request format to optional. • Removed support for old payment type format in message response.
20.x Signature Request Message	Message now aborts execution of the current function when received AND '0009_0006' = 0. • Message now goes to transaction end when received AND '0009_0006' = 0. • Changed offset 4 in request format to allow up to 200 characters. • Message only sent '0009_0002' = 1.

21.x Numeric Input Request Message, On-Demand	Updated - when 21.x is received with prompt text instead of an index, the reject response is sent.
	• Changed offset 9 in request format to prompt index number.
	• Changed offset M+1 in request format to format specifier ID number.
	• Added new allowable values to offset 4 in response format:
	  ○ 4 = Invalid Form
	  ○ 6 = Invalid prompt

23.x Card Read Request, On-Demand	Changed offset 4 in request format to prompt index number. • Response format no longer supports old format. • Changed offset 5 in response format to included only if '0013_0013' = 1. • Moved offset 6 in response format to offset M.

24.x Form Entry Request, On-Demand	• Changed offset 4 in request format to form number.
	• Changed offset P in request format to prompt index number.
	• Changed order of offsets R, R+1, R+2, S, and S+1 in request format.
	• Added new allowable values t offset 4 in response:
	  ○ 1 = Invalid form
	  ○ 6 = Invalid prompt
	• Message can no longer be nested.

25.x Terms and Conditions Request	Changed allowable values for offset 5 in response format to: • Y = Accept • N = Decline

Message	Differences
26.x Run Script Request	• Changed offset 4 in request format, script file must now follow Telium file name rules.
27.x Alpha Input Message	• Changed offset 9 in request format to prompt index number. • Added allowable values t offset 4 in response format RESPONSE format: • 4 = Invalid form • 6 = Invalid prompt
28.x Set Variable Request	• Updated – line display used on RBA forms is seven lines high, forty characters t a line. • Updated bullet describing variable 104. • Removed variable numbers 450-459, 800, and 801. • Changed offset 5 in request format to ASCII – 0.
29.x Get Variable Request	• Added “Variable IDs and Descriptions” table.
31.x PIN Entry Messages	• Changed offset 6 in request format to prompt index number. • Added new allowable values to offset 4 in response format: • 4 = PIN key pressed (key value is returned in the PIN data) • 6 = Invalid prompt • Added new offset 5 to response format, PIN data. • Moved old offset 5 in response format to offset N. • Moved old offset 6 in response format to N+1.
0x and 50.x Authorization Response Message Format	• Changed offset 29 in response format to prompt index number. • Changed offset 31 in response format, can now be either alpha display message from POS or RBA prompt index number.
62.x File Write	Support added for new form file and image formats. • Removed support for unsigned files and old form file format. • Removed allowable value from offset 4 in request format: • 4 = Abort current file • Added allowable value to offset 4 in response format: • 9 = Abort current file
70.x Update Screen	• Not supported.
96.x Convert Images	• Not supported.

See Host Interface Messages for detailed information on all Telium RBA host interface messages.

9_1_6 A.6. Variable Names

Since the Telium implementation of RBA makes use of XML files for various purposes, it has become necessary to modify Ingenico standards to ensure compliance with the XML format. In order to avoid compliance issues, the new Ingenico variable name format is as follows:

```
&lt;?ivPROMPT99?&gt;
```

Where:

- "`<?`" is the escape character for an open bracket.
- "`ivPROMPT99`" is the name of the variable (in this case, referencing prompt index number 99).
- "`?>`" is the escape character for a close bracket.

See Forms for a full list of all RBA variables.

9_1_7 A.7. Forms

The official Ingenico form file format now uses the `*.K3Z` file extension.

The Telium implementation of RBA includes form files for the following Ingenico terminals:

- iCMP (also referred to as iCM122)
- iSMP (also referred to as iMP350)
- iSMPc (also referred to as iMP352)
- iPP320
- iPP350
- iSC250 and iSC Touch 250
- iSC350
- iSC Touch 480
- iUN2xx (includes iUP250 and iUR250 and/or iUC150)
- iWL250

See Forms for detailed information on all form files included with the Telium implementation of RBA.

9_1_8 A.8. Prompts

To ensure compliance with the latest PCI DSS specification, Ingenico terminals now avoid the use of full dynamic text in form files and host interface messages. RBA now references all terminal prompts by calling the prompt ID assigned in the corresponding `*.XML` file.

- `SECURPROMPT.xml` – signed file containing pre-defined Ingenico terminal prompts and format specifiers.
- `CUSTPROMPTS.xml` – signed file containing customized Ingenico terminal prompts.
- `PROMPT.xml` – unsigned file containing custom RBA prompts.

The Telium implementation of RBA has also added support for special characters. This means that Spanish or French prompts containing accented or other special characters can be added to Ingenico prompt files without the need to use the equivalent escape characters.

See Prompts for detailed information on the new Telium RBA prompt implementation.

9_1_9 A.9. RBA Configuration (config.dfs)

See DAT Files DAT Files for detailed information on all Telium RBA configuration parameters.

9_1_9_1 General

General changes seen in the Telium implementation of the `config.dfs` file include the following:

- All timer values now specified in 1/1oths of a second.
- Removed sections `aux3.dat` to `aux6.dat`.

9_1_9_2 Parameters

The following table lists differences between the U32 and Telium implementations of RBA configuration parameters:

config.dfs Paramter Implementation Differences

Parameter	Differences
0002_0003	Removed parameter.
0002_0011	• Default value changed to 1. • Added option: ○ 2 = cashback screen
0002_0012	Added option: • 3 = Return to payment selection screen
0002_0013	Changed option 0, now shown in cents.
0002_0014	Removed parameter.
0003_0001	Changed allowable values to 1 – 65,000.
0003_0002	Changed allowable values to 1 – 5,000.
0003_0005	Changed default value to 0.
0003_0006	Changed, now “MSR Encryption Key Index.”

Parameter	Differences
0003_0012	Added parameter.
0003_0013	Added parameter.
0003_0014	Added parameter.
0003_0016	Added parameter.
0003_0017	Added parameter.
0004_0010	Removed default value (therefore terminal serial number is used).
0005_0002	Removed option 3.
0005_0005	Valid values are now 1 – 65,000.
0006_0001	Changed allowable values to 1 – 600.
0006_0002	Changed options: • 0 = 60 seconds • 100 – 600 = time in 1/10th of a second
0006_0003	Changed options: • 0 = 60 seconds • 20 – 600 = time in 1/10th of a second
0006_0004	Changed default value to 0.
0006_0005	Removed parameter.
0006_0006	• Changed default value to 30. • Changed allowable values to 0 – 65,000.
0006_0010	Removed parameter.
0006_0011	Added parameter.
0006_0012	Added parameter.
0007_0001	Changed allowable values to 2 – 5000.

Parameter	Differences
0007_0008	• Changed default value to 20. • Changed allowable values to 1 – 500.
0007_0009	Changed default value to 2. • Added option: ○ 2 = Send only if amount has been received
0007_0011	Changed parameter.
0007_0012	Changed parameter.
0007_0013	Changed parameter.
0007_0014	Changed parameter.
0007_0015	Removed parameter.
0007_0016	Removed parameter.
0007_0017	Removed parameter.
0007_0018	Removed parameter.
0007_0019	Updated all options.
0007_0020	Removed parameter.
0007_0022	Changed option: • 1 = Display until reset is received Added option: • 2-6500 = Time to display in 1/10th of a second
0007_0023	Changed, now references parameters '0010_0001' and '0007_0022'.
0007_0025	Added parameter.
0007_0026	Added parameter.
0007_0027	Added parameter.

Parameter	Differences
0007_0028	Added parameter.
0007_0029	Added parameter.
0007_0037	Added parameter, 20.x command display time: • 0 = Do not show results. • 1 = Show results with no timeout. • >1 = Results display duration specified in 1/10th second increments. Parameter '0007_0001' must be set to '2' to enable this feature.
0007_0038	Added parameter, 21.x command display time: • 0 = Do not show results. • 1 = Show results with no timeout. • >1 = Results display duration specified in 1/10th second increments. Parameter '0007_0001' must be set to '2' to enable this feature.
0007_0039	Added parameter, 22.x command display time: • 0 = Do not show results. • 1 = Show results with no timeout. • >1 = Results display duration specified in 1/10th second increments. Parameter '0007_0001' must be set to '2' to enable this feature.
0007_0040	Added parameter, 24.x command display time: • 0 = Do not show results. • 1 = Show results with no timeout. • >1 = Results display duration specified in 1/10th second increments. Parameter '0007_0001' must be set to '2' to enable this feature.
0007_0041	Added parameter, 25.x command display time: • 0 = Do not show results. • 1 = Show results with no timeout. • >1 = Results display duration specified in 1/10th second increments. Parameter '0007_0001' must be set to '2' to enable this feature.

Parameter	Differences
0007_0042	Added parameter, 27.x command display time: • 0 = Do not show results. • 1 = Show results with no timeout. • >1 = Results display duration specified in 1/10th second increments. Parameter '0007_0001' must be set to '2' to enable this feature.
0007_0043	Added parameter, 31.x command display time: • 0 = Do not show results. • 1 = Show results with no timeout. • >1 = Results display duration specified in 1/10th second increments. Parameter '0007_0001' must be set to '2' to enable this feature.
0008_0001	• Changed option, 1 = Internal reader. • Removed option 2.
0008_0002	Removed parameter.
0008_0003	Changed options to: • 0 = Do not sound a tone • 1 = Sound a tone
0008_0004	Removed parameter.
0008_0008	Removed parameter.
0008_0009	Removed parameter.
0008_0010	Added parameter.
0009_0001	• Changed default value to 0. • Changed allowable values to 1 – 65,000.
0009_0002	Removed option 2.
0009_0003	Removed parameter.
0009_0004	Changed allowable values to 1 – 65,000.
0009_0005	Removed parameter.

Parameter	Differences
0009_0006	Changed options: • 0 = Terminate before prompting for signature • 1 = Save current state, prompt for signature, then return to saved state
0009_0009	Removed parameter.
0009_0010	Removed parameter.
0009_0011	Removed parameter.
0009_0013	Changed allowable values to 1 – 65,000.
0009_0014	Added parameter.
0010_0002	Default value changed to 13.
0010_0003	Time now specified in 1/10ths of a second.
0010_0006	Obsolete.
0010_0009 – 0010_0028	Added restriction – maximum of 36 characters for each advertising image file name.
0013_0004	Changed option 1 to display time in 1/10ths of a second and added upper limit of 255.
0013_0009	Updated description.
0013_0010	Removed parameter.
0013_0013	Removed parameter.
0013_0014	Added parameter.
0013_0015	Added parameter.
0013_0016	Added parameter.
0030_0009	Added parameter.
0030_0010	Moved parameter (formerly '0030_0009' in U32 implementation).
0030_0011	Moved parameter (formerly '0030_0015' in U32 implementation).

Parameter	Differences
0030_0012	Added parameter.
0030_0013	Moved parameter (formerly '0030_0010' in U32 implementation).
0030_0014	Moved parameter (formerly '0030_0011' in U32 implementation).
0030_0015	Added parameter.
0030_0017	Moved parameter (formerly '0030_0016' in U32 implementation).
0030_0018	Moved parameter (formerly '0030_0017' in U32 implementation).
0030_0019	Moved parameter (formerly '0030_0012' in U32 implementation).
0030_0020	Moved parameter (formerly '0030_0013' in U32 implementation).
0030_0020 – 0030_0030	Updated custom form name parameter range.
0031_0136 – 0030_0145	Removed parameters.

9_1_10 A.10. Format Specifiers

In the latest version of the Telium RBA implementation, format specifiers are defined in the `SECURPROMPT.XML` files. Each format specifier has an assigned ID number, which is used to reference the specifier any time it is called by RBA.

See Format Specifiers for detailed information on Telium RBA format specifiers.

9_2 Appendix B. 3-Byte ASCII Signature Format

This specification describes the format of signature capture data as 3-byte ASCII output from the Retail Base Application. This output method is used to minimize the length of the captured data image while using only printable ASCII characters.

- B.1. Specifications
- B.2. Coordinate Data Reconstruction
- B.3. Format for Signature Data
- B.4. Pen-Up Control Character
- B.5. Segment Start Control Character
- B.6. Coordinate Character Data Set
- B.7. Unused Control Codes

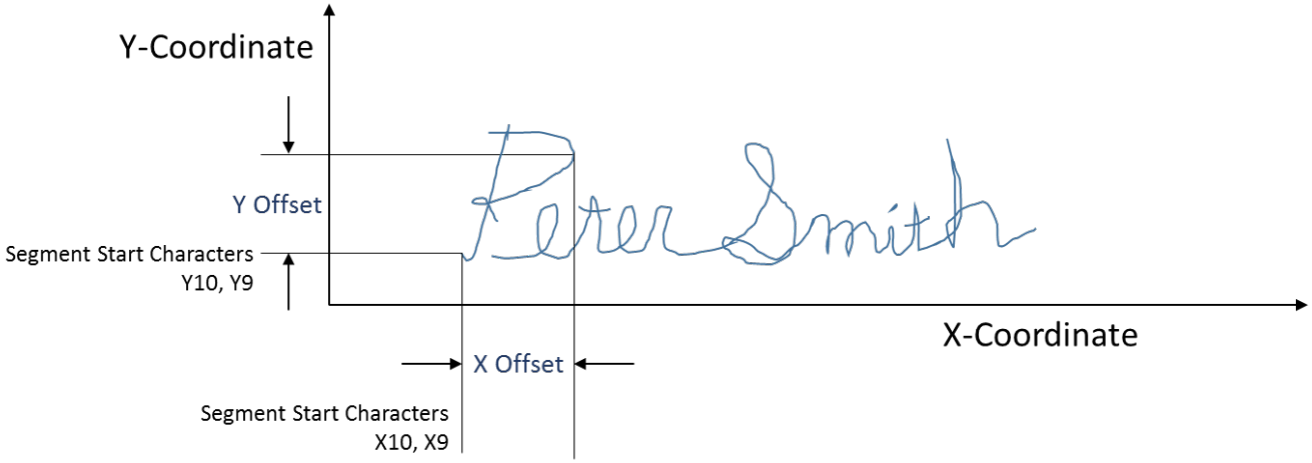
9_2_1 B.1. Specifications

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 3 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 3 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.

9_2_2 B.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 9 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 9 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	9 bytes (9 bit X position, 9 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'	Succeeding coordinates offsets are added again to this data set until the signature is completed or until 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 (e.g., X5 + x5 = X5')

9_2_3 B.3. Format for Signature Data

The signature data transmitted from the Retail Base Application shall have 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 will be followed by one of the following:

- A Pen-Up control character
- A new Segment Start control character
- Another coordinate data set

9_2_4 B.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`

9_2_5 B.5. Segment Start Control Character

- **Binary Format**: 0 1 1 0 X10 X9 Y10 Y9
- **Range (hex)**: 60 to 6F

9_2_6 B.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.

9_2_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

9_2_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

9_2_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

9_2_7 B.7. Unused Control Codes

Control codes 71 through 7E hex are reserved for future use.

9_3 Appendix C. RBA Script Language

9_3_1 What is a Script?

A **script** is a text file which contains a series of statements which define one or more tags with associated parameters. Each statement must begin and end on the same line. A script may also contain comments to explain the intent of the script and white space (space or tab characters) to enhance readability. Comments and white space are generally ignored by the script parser.

Each tag description in a script describes a screen to be displayed when that tag is active and the 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.

9_3_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.

9_3_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.

9_3_4 Tag Parameters

A tag definition must have one or more tag parameters. The following parameters can be specified:

- **Button** – Used to control transitions from one screen to the next, control a button's flags or terminate a script and return a key code.
- **Form** – Used to specify the form that is to be displayed for the tag when it is active.
- **Text** – Used to alter the text displayed on form labels.

9_3_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, alter the appearance of another button on the form via that button's 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 8 button parameters per tag are currently 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.

Currently, buttons, bitmap buttons and checkboxes 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 button's 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 currently 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 currently supported)

> **Note**
>
> `flags` is optional.

9_3_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.

9_3_7 Text Parameter

A text parameter is used to specify 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 effected 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. Currently, 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 and is not available for displaying text parameters.

For terms and conditions, the textID of the text parameter must be 'SELECTTEXT' to allow text parameter substitution.

9_3_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”.

Notes

1. 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.
2. If a form name is not specified, the Terms and Conditions form is used.
3. Any button that does not direct the script to another screen or run a command, terminates the script and sends a response message.
4. “`,`” 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.
5. Valid commands are:

- `-h` Show a button.
- `+h` Hide a button.
- `~h` Toggle a button’s hide state.

9_4 Appendix D. 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.

- D.1. Minimum Production Requirements
- D.2. PayPal Validation Flow
- D.3. Configuration
- D.4. Calculating GMT Offset (Variable 205)

9_4_1 D.1. Minimum Production Requirements

Minimum requirements for using PayPal as a part of an RBA production solution include the following:

- RBA application version 2.7.3 or newer.
- 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 individual terminal's Operation and Product Support Guide for this requirement.

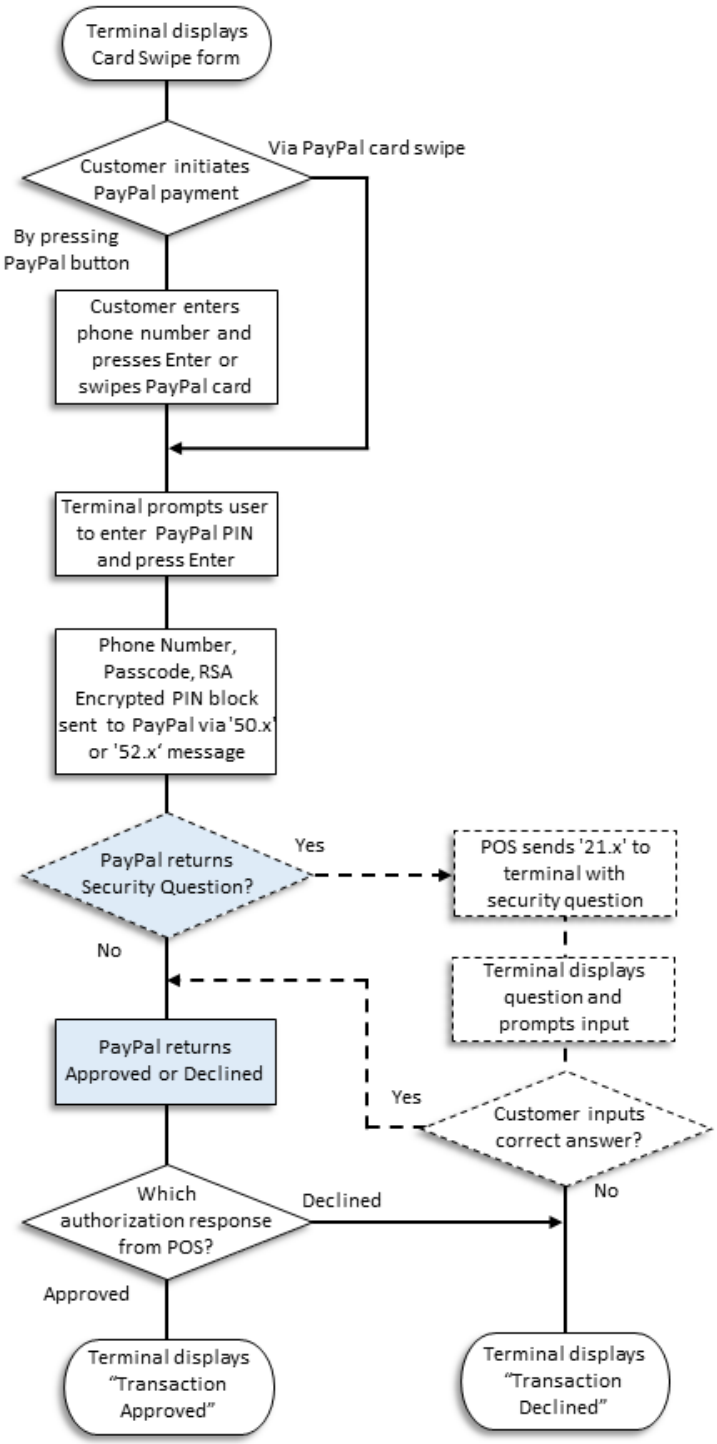
> For testing purposes, use `GENERIC_T.PEM`, the test key provided by PayPal. See section PayPal Configuration (paypal.dat) for additional information.

> 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.

> In order 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`.

9_4_2 D.2. PayPal Validation Flow

The PayPal validation flow is as follows. The optional flow (section with dotted lines) is unavailable in pre-release 2.7.0.0009.



PayPal Validation Flow

The PayPal enabled 50.x: Authorization Request Message includes new 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.

9_4_3 D.3. Configuration

9_4_3_1 PayPal.dat

See section PayPal Configuration (paypal.dat) for details.

9_4_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`.

Parameter Name	DFS Data Index	Default Value	Description
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`.
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.

9_4_4 D.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.

9_4_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`.

9_4_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.

9_5 Appendix E. RBA Best Practices

- E.1. Working With RBA
- E.2. Customizing RBA
- E.3. Packaging a New RBA Load

9_5_1 E.1. Working With RBA

9_5_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 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 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.

9_5_1_2 Retrieving EPS Encrypted Data

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.

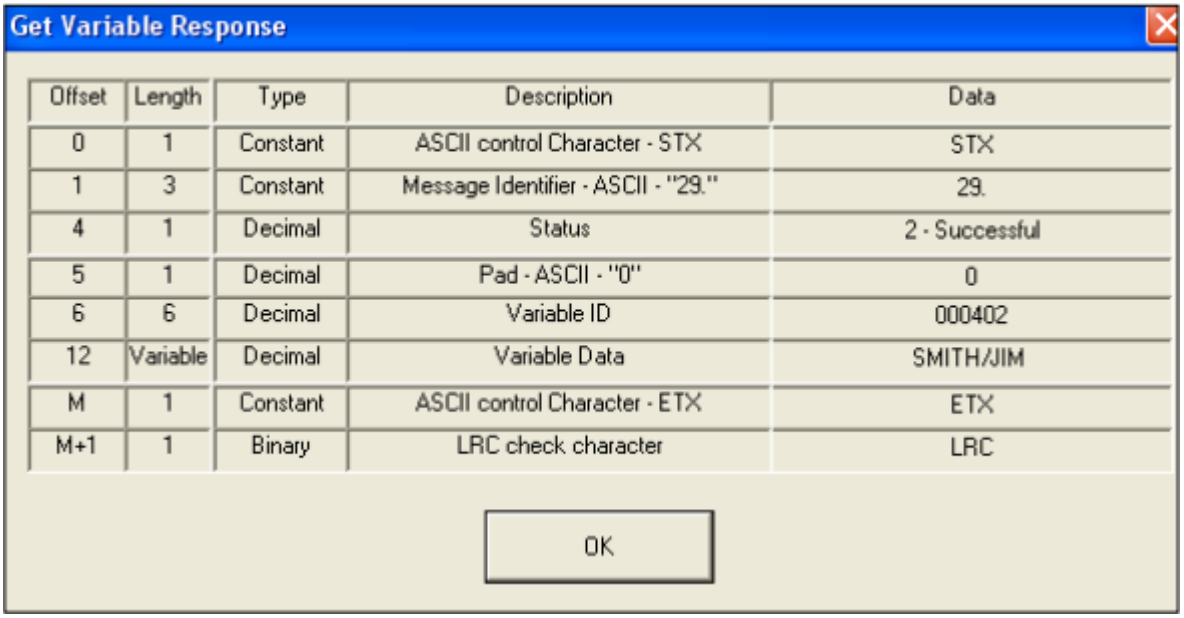
RBA’s tender processing flow continues to works normally.

The 19.x message is modified to support EPS functionality in the following ways:

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.



29.x Message Response with Customer Name

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.

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.

9_5_2 E.2. Customizing RBA

9_5_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.

9_5_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.

9_5_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.

9_5_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.

9_5_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.

9_5_3 E.3. Packaging a New RBA Load

To generate an RBA load package for an Ingenico terminal, follow these steps:

1. Click the `XXXXXXPackageGZ.BAT` file for the desired terminal (where "XXXXXX" is the desired terminal model).
2. The new package is placed in `RBA Data and Parameters\package\<terminal model>`.

Once the package is built, it can be loaded to the desired Ingenico terminal using LLT.

9_6 Appendix F. External Display for Telium Terminals

The external display feature is available as a factory option on Ingenico iSC480 terminals. Software has been updated for the iSC350 and iSC480 to support this feature. There are no user configuration options required to enable or disable this feature since the functionality is always enabled once the RBA is launched. While the iSC350 is also capable of outputting video (VGA) with the proper factory configuration, this option is not generally available. As of this time, the only available Telium terminal which supports external video is the iSC480 via an HDMI port.

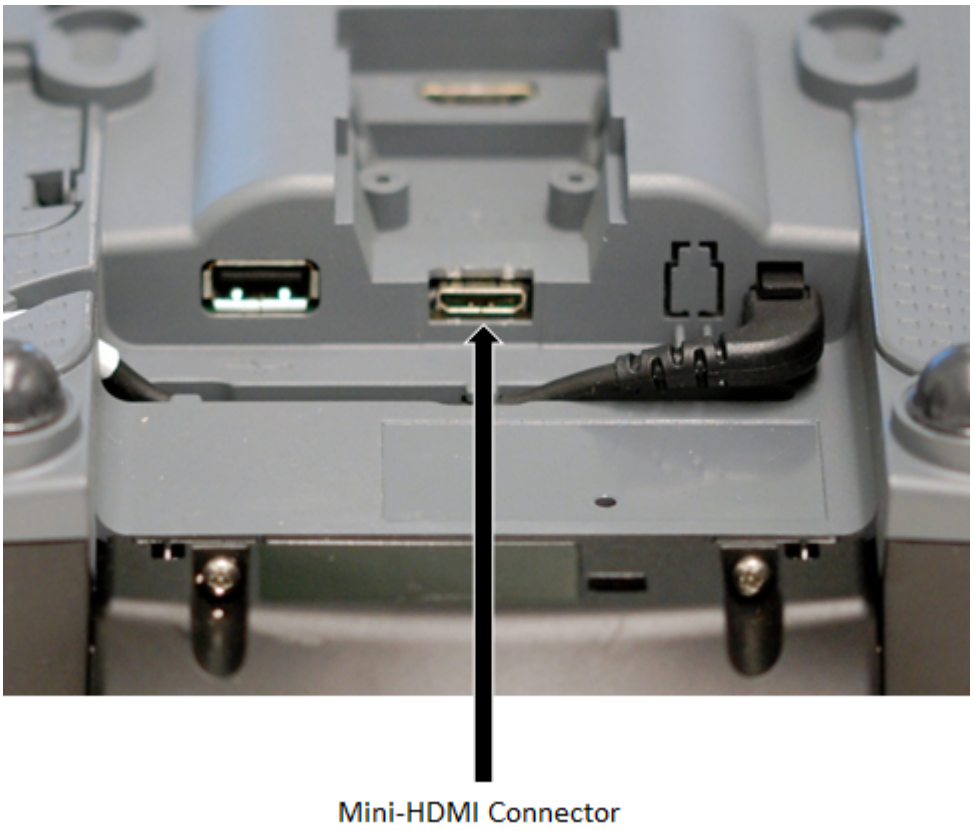
Refer to the below table which shows Telium terminals capable of supporting external video.

External Video Compatibility

Terminal	External Display	Display Interface
iPP320	No	
iPP350	No	
iSC250	No	
iSC350	See Note 1	Specific iSC350 hardware required for video output.
iSC480	Yes	Factory option external video via Mini-HDMI connector (Type C).
iSMP	No	
iUP250	No	
iWL250	No	

> **Note 1**
>
> The specific iSC350 hardware required for video output is not generally available.

Refer to the below image which shows 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

9_7 Appendix G: eWIC Implementation

9_7_1 G.1. eWIC Overview

9_7_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 now offline EBT smart cards. Ingenico terminals have supported online MSR WIC cards similar to all other MSR EBT cards for many years, but now also support offline WIC smart cards. Please refer to ANSI 2005 for specification details unless stated otherwise.

9_7_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.

9_7_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.

9_7_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's PAN to any of the following values, as described in the following table.

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 currently indicate the WIC smart card type in the 8th digit of the card's PAN whereas Texas's smart cards indicate the WIC smart card type in the 7th PAN digit.

As an example, the 7th digit of the following State of Texas smart card account is a ‘1’, indicating that the card is a production card.

```
5077171016802632028
```

As another example, the 8th digit of the following State of Wyoming smart card account is a ‘9’, indicating that the card is a training card.

```
5053495900027662
```

9_7_2 G.2. eWIC WMP Messages

9_7_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.

9_7_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.
_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.

9_7_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.

9_7_2_4 N on-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
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

9_7_2_5 WMP Message Format

The following tables describe the WIC Request and Response message formats. See Appendix G.3. eWIC 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 G.4. 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.
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 G.4. 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.
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.

9_7_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.

9_7_3 G.3. eWIC Transaction Flows

9_7_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.

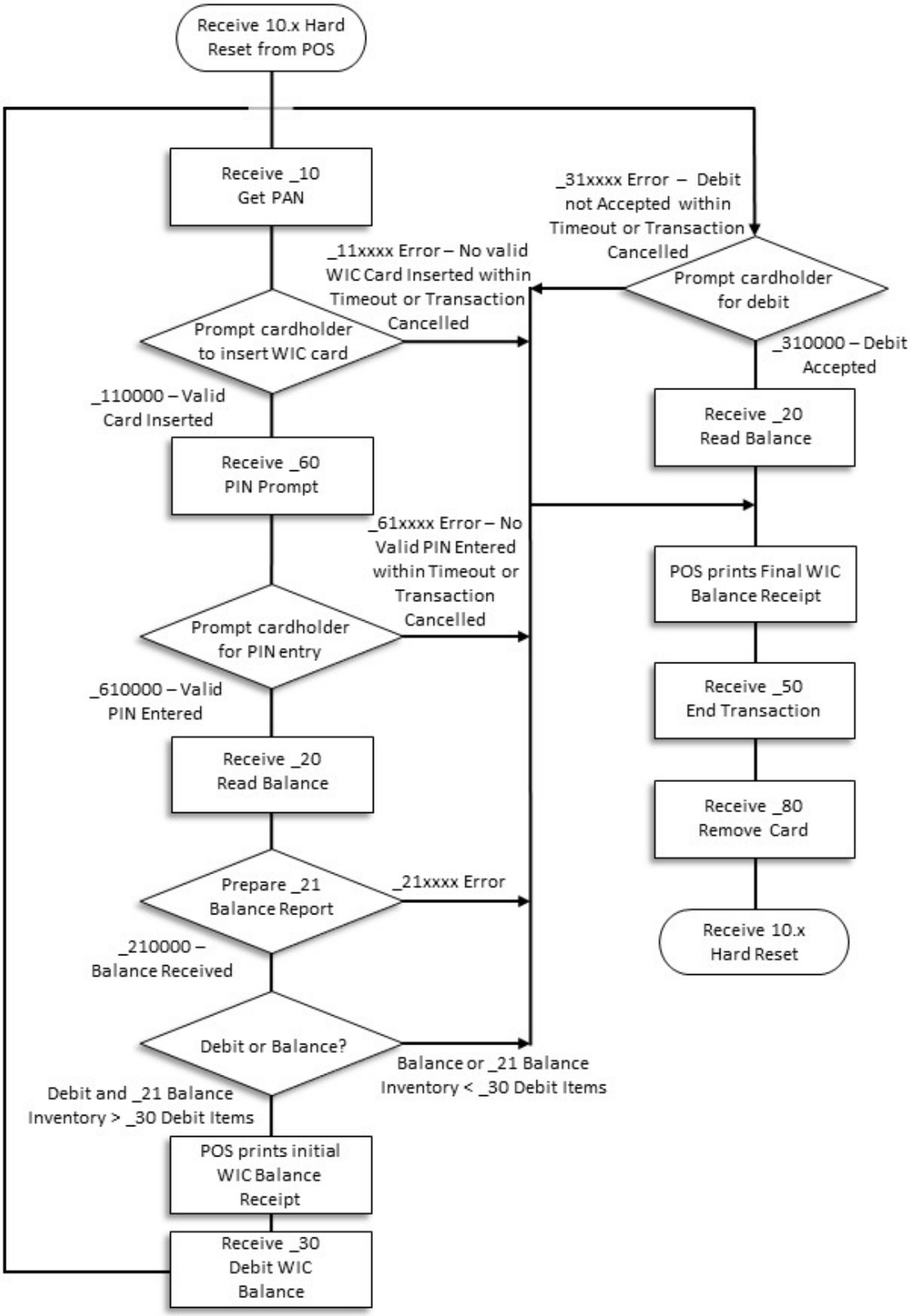
Half of the WIC transaction messages initiate and/or require cardholder-to-terminal interaction (see also G.2. eWIC WMP Messages, Section 'Variable/Interactive messages'). Most balance validation and debit calculations must be performed by the terminal's WIC library (WSPI layer) rather than the smart cards themselves.

9_7_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 Messages.

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 _10 Get PAN
Prompt cardholder to insert WIC card
_11xxxx Error – No valid WIC Card Inserted within Timeout or Transaction Cancelled
_110000 – Valid Card Inserted
Receive _60 PIN Prompt
Prompt cardholder for PIN entry
_61xxxx Error – No Valid PIN Entered within Timeout or Transaction Cancelled
_610000 – Valid PIN Entered
Receive _20 Read Balance
Prepare _21 Balance Report
_21xxxx Error
_210000 – Balance Received
Debit or Balance?
Balance or _21 Balance Inventory < _30 Debit Items
Debit and _21 Balance Inventory > _30 Debit Items
POS prints initial WIC Balance Receipt
Receive _30 Debit WIC Balance
_31xxxx Error – Debit not Accepted within Timeout or Transaction Cancelled
Prompt cardholder for debit
_310000 – Debit Accepted
Receive _20 Read Balance
POS prints Final WIC Balance Receipt
Receive _50 End Transaction
Receive _80 Remove Card
Receive 10.x Hard Reset

WIC Transaction Flow

9_7_3_3 eWIC Sample Message Flows

The table below illustrates the message flow for enabling WIC transactions.

Initial WIC Configuration

<table>
<tr><th colspan="2">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></td><td>00.</td><td>The POS commands the terminal to go offline.</td><td>11.00Lane Closed</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays the offline message "This Lane Closed".</td><td>11.00Lane Closed</td></tr>
<tr><td colspan="2"> </td><td></td><td>60.20[GS]1[GS]1</td><td>The POS enables WIC transactions.</td><td>11.00Lane Closed</td></tr>
<tr><td colspan="2"> </td><td></td><td>01.</td><td>The POS commands the terminal to go offline.</td><td>11.01Slide Card</td></tr>
<tr><td colspan="2"> </td><td></td><td>01.00000000</td><td>The terminal returns an online message.</td><td>11.01Slide Card</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays the online swipe/insert screen.</td><td>11.01Slide Card</td></tr>
<tr><td colspan="2"> </td><td></td><td>_00...</td><td>POS configures WIC authorities.</td><td>11.01Slide Card</td></tr>
<tr><td colspan="2"> </td><td></td><td>_010000NN??...
VERS..</td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_01</td><td>Message identifier: WSPM Reset /Activation response.</td></tr><tr><td>0000</td><td>No error code.</td></tr><tr><td>NN</td><td>Number of WIC authorities.</td></tr><tr><td>??...</td><td>List of WIC authorities.</td></tr><tr><td>VERS..</td><td>DLL versions for WIC authorities.</td></tr></table></td><td>11.01Slide Card</td></tr>
<tr><td colspan="2"> </td><td></td><td>10.</td><td>The POS resets the terminal's transaction data for the next transaction.</td><td>11.01Slide Card</td></tr>
</table>

9_7_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
			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

<table>
<tr><th colspan="2">POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="3"></td><td></td><td>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".</td><td>'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.</td></tr>
<tr><td colspan="3"> </td><td>_110000??LLPAN...</td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_11</td><td>Message identifier: Get Card PAN response message.</td></tr><tr><td>0000</td><td>No error code.</td></tr><tr><td>??...</td><td>WIC authority.</td></tr><tr><td>LL</td><td>PAN length.</td></tr><tr><td>PAN...</td><td>PAN.</td></tr></table></td><td>11.19WIC</td></tr>
<tr><td colspan="3"></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td>11.19WIC-?? / Please wait... Do not Remove Card</td></tr>
<tr><td colspan="3"> </td><td>_60</td><td>The POS sends a WIC Authenticate cardholder message.</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="3"></td><td></td><td>The terminal displays "Please enter your PIN".</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="3"></td><td></td><td>The cardholder enters a valid PIN.</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="3"></td><td></td><td>The terminal briefly displays "PIN OK".</td><td>11.19 WIC</td></tr>
</table>

<table>
<tr><th colspan="2">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>_61000000</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>11.19 WIC</td></tr>
<tr><td colspan="2"></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td>11.19WIC-?? / Please wait... Do not Remove Card</td></tr>
<tr><td colspan="2"> </td><td>_20</td><td>The POS sends a WIC Read Card Balance message.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </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 colspan="2"></td><td></td><td>The POS prints a balance receipt.</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>
<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>

9_7_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

<table>
<tr><th colspan="2">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></td><td></td><td>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".</td><td>'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.</td></tr>
<tr><td colspan="2"> </td><td></td><td>_110000??LLPAN...</td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_11</td><td>Message identifier: Get Card PAN response message.</td></tr><tr><td>0000</td><td>No error code.</td></tr><tr><td>??...</td><td>WIC authority.</td></tr><tr><td>LL</td><td>PAN length.</td></tr><tr><td>PAN...</td><td>PAN.</td></tr></table></td><td>11.19WIC</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </td><td></td><td>_60</td><td>The POS sends a WIC Authenticate cardholder message.</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays "Please enter your PIN".</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The cardholder enters a valid PIN.</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal briefly displays "PIN OK".</td><td>11.19 WIC</td></tr>
</table>

<table>
<tr><th colspan="2">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>_61000000</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>11.19 WIC</td></tr>
<tr><td colspan="2"></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td>11.19WIC-?? / Please wait... Do not Remove Card</td></tr>
<tr><td colspan="2"> </td><td>_20</td><td>The POS sends a WIC Read Card Balance message.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </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 colspan="2"></td><td></td><td>The POS prints an initial balance receipt.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </td><td>_30</td><td>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.</td><td>11.19WIC</td></tr>
</table>

<table>
<tr><th colspan="2">POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="3"></td><td></td><td>The terminal displays "Accept changes? / ENTER or CANCEL" with "YES / NO" buttons.</td><td>11.19WICAccept changes? / ENTER or CANCEL</td></tr>
<tr><td colspan="3"></td><td></td><td>The cardholder accepts via "ENTER" or "YES" buttons.</td><td>11.19WIC</td></tr>
<tr><td colspan="3"></td><td></td><td>The terminal briefly displays "Transaction accepted".</td><td>11.19Transaction accepted</td></tr>
<tr><td colspan="3"></td><td></td><td>The terminal displays "Updating card".</td><td>11.19Updating card</td></tr>
<tr><td colspan="3"></td><td>_310000??
LLSIG...</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>0000</td><td>No error code.</td></tr><tr><td>??...</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>11.19WIC</td></tr>
<tr><td colspan="3"> </td><td>_20</td><td>The POS sends a WIC Read Card Balance message.</td><td>11.19WIC</td></tr>
<tr><td colspan="3"> </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 balance information.</td></tr></table></td><td>11.19WIC</td></tr>
</table>

<table>
<tr><th colspan="2">POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="3"></td><td></td><td>The POS prints an updated balance receipt.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </td><td></td><td>_50</td><td>The POS sends a WIC End Transaction message.</td><td>11.19WIC</td></tr>
<tr><td colspan="3"></td><td></td><td>The terminal displays "Transaction complete".</td><td>11.19Transaction complete</td></tr>
<tr><td colspan="2"> </td><td></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></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></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></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

<table>
<tr><th colspan="2">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></td><td>_10</td><td>The POS sends a WIC Get PAN message.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays the "Please insert card" screen.</td><td>11.19Please insert card</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>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".</td><td>'11.19WIC' is also returned if a WIC smart card is inserted before _10 Get PAN Message is sent and processed.</td></tr>
<tr><td colspan="2"></td><td></td><td>_110000??
LLPAN...</td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_11</td><td>Message identifier: Get Card PAN response message.</td></tr><tr><td>0000</td><td>No error code.</td></tr><tr><td>??...</td><td>WIC authority.</td></tr><tr><td>LL</td><td>PAN length.</td></tr><tr><td>PAN...</td><td>PAN.</td></tr></table></td><td>11.19WIC</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td>11.19WIC-?? / Please wait... Do not Remove Card</td></tr>
<tr><td colspan="2"> </td><td></td><td>_60</td><td>The POS sends a WIC Authenticate cardholder message.</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays "Please enter your PIN".</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The cardholder enters a valid PIN.</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal briefly displays "PIN OK".</td><td>11.19 WIC</td></tr>
</table>

<table>
<tr><th colspan="2">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></td><td>_61000000</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>11.19WIC</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </td><td></td><td>_20</td><td>The POS sends a WIC Read Card Balance message.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </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 colspan="2"></td><td></td><td></td><td>The POS prints an initial balance receipt.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </td><td></td><td>_30</td><td>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.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays "Accept changes? / ENTER or CANCEL" with "YES / NO" buttons.</td><td>11.19Accept changes? / ENTER or CANCEL</td></tr>
</table>

<table>
<tr><th colspan="2">POS</th><th>Terminal</th><th>Message</th><th>Description</th><th>11.x Status Return Message</th></tr>
<tr><td colspan="3"></td><td></td><td>Cardholder cancels PIN entry via "CANCEL" button or by removing the WIC smart card.</td><td>11.19WIC</td></tr>
<tr><td colspan="3"></td><td></td><td>The terminal displays "WIC update cancelled".</td><td>11.19WIC</td></tr>
<tr><td colspan="3"> </td><td>_31008C??00</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>11.19WIC</td></tr>
<tr><td colspan="3"> </td><td>_50</td><td>The POS sends a WIC End Transaction message.</td><td>11.19WIC</td></tr>
<tr><td colspan="3"> </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="3"> </td><td>_80</td><td>The POS sends a WIC Remove Card message.</td><td>11.20Remove card</td></tr>
<tr><td colspan="3"> </td><td>_81</td><td>The terminals sends a WIC Remove Card response message.</td><td>11.20SMC</td></tr>
<tr><td colspan="3"> </td><td>10.</td><td>The POS resets the terminal's transaction data for the next transaction.</td><td>11.20SMC</td></tr>
</table>

9_7_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

<table>
<tr><th colspan="2">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></td><td>10.</td><td>The POS resets the terminal's transaction data for the next transaction.</td><td>11.01Slide Card</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays the online swipe/insert screen.</td><td>11.01Slide Card</td></tr>
<tr><td colspan="2"> </td><td></td><td>_10</td><td>The POS sends a WIC Get PAN message.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays the "Please insert card" screen.</td><td>11.19Please insert card</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>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".</td><td>'11.19WIC' is also returned if a WIC smart card is inserted before _10 Get PAN Message is sent and processed.</td></tr>
<tr><td colspan="2"> </td><td></td><td>_110000??LLPAN...</td><td><table><tr><th>Message Fragment</th><th>Breakdown</th></tr><tr><td>_11</td><td>Message identifier: Get Card PAN response message.</td></tr><tr><td>0000</td><td>No error code.</td></tr><tr><td>??...</td><td>WIC authority.</td></tr><tr><td>LL</td><td>PAN length.</td></tr><tr><td>PAN...</td><td>PAN.</td></tr></table></td><td>11.19WIC</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td>11.19WIC-?? / Please wait... Do not Remove Card</td></tr>
</table>

POS	Terminal	Message	Description	11.x Status Return Message
	→	_60	The POS sends a WIC Authenticate cardholder message.	11.03Please enter your 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 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 colspan="2">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>_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<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>11.06Transaction cancelled</td></tr>
<tr><td colspan="2"> </td><td>_50</td><td>The POS sends a WIC End Transaction message.</td><td>11.06Transaction cancelled</td></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>
</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

9_7_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
			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.
←		_110000??LLPAN...	**Message Fragment** / **Breakdown** _11 — Message identifier: Get Card PAN response message. 0000 — No error code. ??... — WIC authority. LL — PAN length. PAN... — PAN.	11.19WIC

<table>
<tr><th colspan="2">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></td><td></td><td>The terminal displays "WIC-?? / Please wait... Do not Remove Card".</td><td>11.19WIC-?? / Please wait... Do not Remove Card</td></tr>
<tr><td colspan="2"> </td><td></td><td>_60</td><td>The POS sends a WIC Authenticate cardholder message.</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>Cardholder cancels PIN entry via "CANCEL" button or by removing the WIC smart card.</td><td>11.03Please enter your PIN</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td>The terminal displays "Transaction cancelled."</td><td>11.19Transaction cancelled</td></tr>
<tr><td colspan="2"> </td><td></td><td>_61008EFF</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>008E</td><td>WIC PIN entry cancelled.</td></tr><tr><td>FF</td><td>No valid PIN entered.</td></tr></table></td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </td><td></td><td>_50</td><td>The POS sends a WIC End Transaction message.</td><td>11.19WIC</td></tr>
<tr><td colspan="2"> </td><td></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></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

9_7_4 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".
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".

Error Code	Error Code Name	Error Conditions	Terminal Displays
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".

Error Code	Error Code Name	Error Conditions	Terminal Displays
0x0093	WSPI_CARDTYPE_ERROR	WIC smart card type is not enabled for current mode (configured in _00 message, see G.2. eWIC WMP Messages).	"Authentication failed".
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

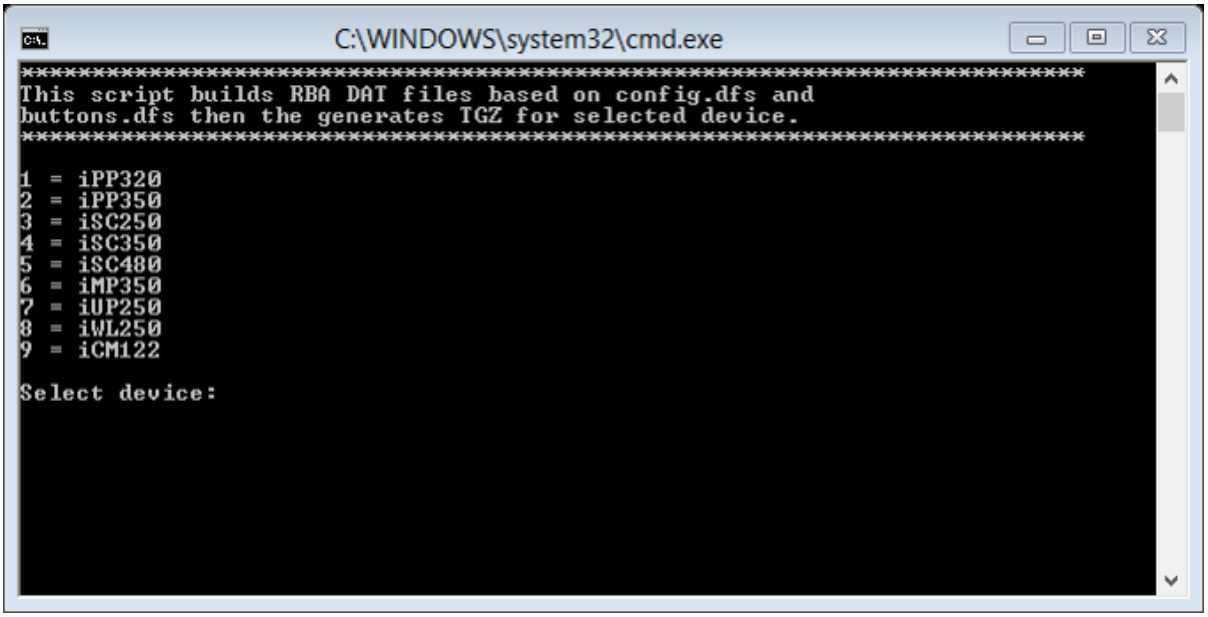
Error Code	Error Code Name	Error Conditions	Terminal Displays
0x9993	WSPI_TENDER_MISMATCH	Cardholder selected another tender on terminal (not WIC).	No change in display.
0x9995	WSPI_NO_ACTIVE_STATE	WIC disabled (i.e. RBA configuration parameter '0020_0001' = '0').	No change in display.
0x9997	WSPI_STATE_MODULE_MISSING	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".

9_8 Appendix H: 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.



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_Revision History

Manual Revision	Application Revision	Changes
Rev 9	RBA 15.0.1 (Non-KP4) RBA 15.0.2 (KP4)	• Added flag 31 (EMV Cashback) to the EMV '33.01.x' Status Message. • Added a list of RBA Form Variables for customizing forms. Moved IP Address and Port Display Variable for TCP-IP to immediately follow. • Added a page for the Bluetooth Status Icon. • Added a new Form Customization page, parent to Bluetooth Status Icon and RBA Form Variables. • Corrected incorrect values on the ICS Tags in EMVCONTACT.XML page. • Added an info box to the Form Contents and Descriptions page regarding the new line display length /number of columns on iSC480 forms. • Added a note in Main Flow (mainFlow.dat) for the 0007_0034 and 0007_0035 parameters to show that active contactless or smart card readers cause the device to remain active and ignore inactivity timers. • Reorganized content; created a new section titled Drivers and Tools. Moved the RBA Testing Tool page to this section. Separated Ingenico iConnectEFT Constants from Ingenico iConnectREST and included these pages in this new section as well. • Added a new section describing the procedure for Disabling Electronic Signature on iSC Series Terminals. • Added a paragraph to the Card Configuration (cards.dat) section describing use of the Verify Amount flag to suppress the "Amount OK?" screen when the merchant opts to use their own amount verification screen during EMV transactions. Added a notation regarding this to the EMV Purchase Transaction Flow. • Added the section "Data to be Encrypted if Track 3 is Available" to the Monetra CardShield Encryption page and renamed the original "Data to be Encrypted" section "Data to be Encrypted if Track 3 is Unavailable". Also updated the Introduction section. • Added payment type '-' to the 19.x: BIN Lookup and 86.x On-Guard and KME BIN Lookup (PIN Encouragement) Message response messages. • Both 28.x: Set Variable Request and EMV Cashback now show that using variable 303 in a 29.x message will return the current transaction's purchase amount. • Added variable 306 to the 28.x: Set Variable Request's RBA Variables table, and marked it for use with only the 29.x message. • EMV Cashback shows that variable 307 returns the current transaction total (purchase amount + cashback amount). • EMV '33.09.x' Set Tag Data Message now states that only tags with a tag ID of 2 digits or 4 digits may be set with this message. • Terms and Conditions (TC1.xml) now lists the maximum number of characters (7500) it will allow. • EMV Cashback now shows 13.x cashback notification, and sample 13.x and 29.x messages.

Manual Revision	Application Revision	Changes

		• Added offset "O + 1" to the 50.x: Authorization Request's message format table for the optional inclusion of an ETB.
		• Expanded upon 0091_0008 in the table "Relevant Parameters for Voltage TEP1 and TEP2 in config.dfs" on the Voltage TEP1 and TEP2 Encryptions page, giving both an example format for and breakdown of a sample 50.x message including an ETB. Also corrected the Sample Message table which contained an incorrect sample Account Message.
		• Form Files (forms.dat) has been updated per the latest config.dfs release.
		• Added parameter '0005_0010' "Append Service Code" to the table of parameters in BIN Lookup (stb.dat). Updated the 19.x: BIN Lookup message with the optional service code field.
		• Added a new section on Using the iSC480 Terminal Screen to Display Contactless Status.
		• Added a new section describing EMV Full and Partial Refunds. Added the following transaction flows:
		  ◦ EMV Partial Refund Normal Transaction Flow
		  ◦ EMV Partial Refund On-Demand Transaction Flow
		• Created a section for iCMP and iSMPc Bluetooth Pairing and Unpairing.
		• Updated the variable table for the 28.x: Set Variable Request message.
		• Added a new section describing Using the Function Keys to Select Menu Options.
		• Created a page for Communications Supported per Terminal Model.
		• Reordered the Transaction Step List and removed the obsolete "set proprietary tags" step.
		• Updated the Application Selection section by adding the four following flow diagrams based on the setting of configuration parameter '0019_0003':
		  ◦ Application Selection for '0019_0003' = '0'
		  ◦ Application Selection for '0019_0003' = '1'
		  ◦ Application Selection for '0019_0003' = '2'
		  ◦ Application Selection for '0019_0003' = '3'
		• Updated Editing Config.dfs with the new GEN_TGZ.BAT method for generating DAT files and TGZ packages.

Manual Revision	Application Revision	Changes

Rev 8.1	RBA 14.0.5 (Non-KP4)
	PK-RGENxx-1405x
	RBA 14.0.4 (KP6)
	PK-RGENxx-1406x
		• Updated the arrangement of information on the Examples of Format Specifiers page.
		• Restored the iConnectEFT constants to the 09.x Card Status Message.
		• The spacing and symbol usage in host interface messages has been standardized.
		• Standardized the usage of the Note macro.
		• Standardized monospace and all-caps usage.
		• Added mention of automatic file name capitalization in both the 62.x: File Write and the 63.x: Find File messages.
		• Updated the body text of Mercury Encryption to remove redundancies and obsolete information.
		• Updated the EMV and Non-EMV Tags Transmitted in Host Interface Messages section with additional tags used in the EMV '33.02.x' and '33.07.x' messages. This includes:
		  ◦ Tags T42, T84, T5F55, T9FE1, T9F33 and D1016 for the EMV '33.07.x' message.
		  ◦ Tag T9F07 for the EMV '33.02.x' message.
		• Updated the following EMV response messages to return only '0' for both packet number and packet type fields (as only one packet is returned by an EMV response message):
		  ◦ EMV '33.00.x' Transaction Initiation Message
		  ◦ EMV '33.04.x' Authorization Response Error Reply Message
		  ◦ EMV '33.07.x' Terminal Capabilities Message
		  ◦ EMV '33.08.x' Set Variables Message
		  ◦ EMV '33.09.x' Set Tag Data Message
		  ◦ EMV '33.10.x' Get Tag Data Message
		• The Non-EMV Refund Transaction Flow has been corrected to exclude the deprecated 0.x and 50.x messages.

Rev 8	RBA 14.0.3 (Non-KP4)
	PK-RGENxx-1403x
	RBA 14.0.4 (KP4)
	PK-RGENxx-1404x
		• Updated the following sections with new requirements for selecting from TLS v1.1 and v1.2:
		  ◦ SSL/TLS for Ethernet
		  ◦ Security Parameters (security.dat)
		• Updated inclusions and exceptions to parameter '0091_0002' in Security Parameters (security.dat).
		• Created new illustration for S1 Encryption Process based on original illustration.
		• Updated list of environments which integrate with RBA in About This Manual.
		• Clarified function of 30.x: Advertising Request Message (On-Demand) and updated page format.
		• Updated 07.x: Unit Data Request and 08.x: Health Stat to return "ALERT" during Alert Irruption condition.
		• Updated 87.x On-Guard and KME Card Read Data to be available while the device is offline.
		• Updated 61.x: Configuration Read to allow for reading of multiple parameters.
		• Removed references to variables 121, 122, 123, and xxx from 28.x: Set Variable Request.

Manual Revision	Application Revision	Changes
		• Updated format and content references of Standard ISO Cards, On-Demand, and the MSR Encryption section. • Updated the Transaction Codes section of the BIN Processing (allBins.dat, bin0.dat - bin20.dat) page. Created a separate Card Type Default Values table which lists the default values for each card type. Originally there was a common reference to a default value in the cards.dat extract, but the default for some of these is different. • Improved format of B.2. Coordinate Data Reconstruction, adding an image for signature capture as a visual aid. • Added information on the power-saving limitation to the 09.x Card Status Message and EMV '33.01.x' Status Message. • Updated the Data Message Format page for format, consistency, and content. • Added a page for Authorization Response Codes. • Added exit type 8 for the 24.x: Form Entry Request (On-Demand). • Updated Transaction Prompts (PROMPT.xml) to include a complete list of prompts in three languages. • Added and edited the description on the Non-EMV Tag Definitions page and alphabetized the error codes for tag D1010. • Table column entries now have uniform standardized centering, and their column headers are all centered. • Table and image titles are now all bold, not italicized. Tables now all have titles above them, and images have titles captioned below. • I.e. and e.g. usage has all been checked. • Updated the Communication Settings page and migrated Bluetooth-related features to the Telium Download and User's Guide. • Added new section on Support for Voice Referral for EMV. • Updated Main Flow (mainFlow.dat) and 09.x Card Status Message to include the '0007_0047' variable. • Added the page EMV Cashback. • Updated EMV and Non-EMV Tags Transmitted in Host Interface Messages for a number of tags newly used in EMV messages • Updated 12.x: Account Message's criteria for '12.1' response message "failure" status. • Updated EMV Tag Data Format Examples. • Added '0003_0018' flag to Non-Standard Cards and MSR Card Swipe Options (msr.dat) to enable reading (some) non-standard MSR cards.

Manual Revision	Application Revision	Changes

		• Updated .DAT file tables per the latest config.dfs release:
		  ◦ Barcode Parameters (barcode.dat)
		  ◦ BIN Lookup (stb.dat)
		  ◦ BIN Processing (allBins.dat, bin0.dat - bin20.dat)
		  ◦ Compatibility Flags (compat.dat)
		  ◦ EMV Flags (emv.dat)
		  ◦ Form Files (forms.dat)
		  ◦ Main Flow (mainFlow.dat)
		  ◦ Security Parameters (security.dat)
		  ◦ Status Messages (status.dat)
		  ◦ Store/Lane Information (store.dat)
		• Added new section describing MSR Encryption Example for On-Guard encryption.
		• Updated Security Parameters (security.dat) to update option '2' for On-Guard encryption so that it no longer shows "unsupported". Also added a note on changing this setting.
		• Updated EMV with P2PE Enabled and added a page for EMV Tag Encryption to correct discrepancies on which/when tags are encrypted, masked, or sent in the clear.
		• Updated 04.x: Set Payment Type Request and Card Configuration (cards.dat) pages for the latest config.dfs release.
		• Added source type "S" to the 23.x: Card Read Response Message.
		• Added support for cashback, describing parameters in emvaid.dat and emvbrand.dat in Available EMV Application IDs.
		• Added tags for partial and complete authorization to EMV '33.03.x' Authorization Request Message and updated tag usage on the EMV and Non-EMV Tags Transmitted in Host Interface Messages page.
		• Corrected the packet length and behavior of the EMV '33.04.x' Authorization Response Error Reply Message.
		• Host Interface Messages are now in a standard format: number in quotes if the decimal is followed with a character other than "x", followed by the message name for its first appearance on a page, thereafter referred to as its number followed by "message", "request", or "response".
		• Added a notation in the Dynamic Update of RSA-OAEP Public Keys section stating that this feature is only supported in v12.2.1/2 of RBA.
		• Updated the EMV and Non-EMV Tags Transmitted in Host Interface Messages table to show that tag T9F11 is included in the EMV '33.02.x' Track 2 Equivalent Data Message.
		• Standardized configuration parameter references in-text. All should now appear as 'xxxx_xxxx'.
		• Remaining instances of "ECR" were changed to "POS"; likewise for "device" to "terminal" when referring to an Ingenico pinpad terminal.
		• I.e. and e.g. mis-usage has been corrected.

Manual Revision	Application Revision	Changes

		• Updated links for 82.x, 83.x, 84.x, 85.x, 86.x, 87.x, 88.x, and 89.x throughout the document to state "On-Guard and KME".
		• Added Manual Entry specifications to Voltage TEP1 and TEP2 Encryption Examples, RSA-OAEP and TransArmor Encryption Examples, and Monetra CardShield Encryption Examples.
		• Updated the 86.x On-Guard and KME BIN Lookup (PIN Encouragement) Message to supplanted by 09.x, except in Canadian applications with On Guard Encryption enabled.
		• Added the D1011 tag to Non-EMV Tag Definitions and special authorization functionality to the EMV '33.04.x' Authorization Response Message.
		• Reformatted the BIN Processing Parameters and Descriptions table in the BIN Processing (allBins.dat, bin0.dat - bin20.dat) section into separate smaller tables.
		• Updated the 00.x: Offline Message with Message Responses in the Offline State.
		• Updated EMV and Non-EMV Tags Transmitted in Host Interface Messages to include T9F07.
		• Standardized use of "Track #" vs "track data".
		• Updated the EMV and Non-EMV Tags Transmitted in Host Interface Messages section with tags used in the '33.07.x' Terminal Capabilities message.
		• Updated criteria for the '12.1' response message failure status.
		• Re-uploaded images so that more users could view them in Confluence.
		• Updated the EMV and Non-EMV Tags Transmitted in Host Interface Messages section with additional tags included in the '33.02.x' message.
		• Included complete message type (command '33.' and subcommand XX.x) in all references to EMV messages (e.g., EMV '33.01.x' Status Message).
		• Moved the "Configuring GMT Variables for PayPal Authorization" subsection to follow the 28.x: Set Variable Request "Overview."
		• Added a new section describing EMV Authorization Response Codes.
		• Updated iConnectEFT constants where necessary to bring current.
		• Reorganized the MSR Encryption section so that encryption types are organized alphabetically, also organized other content in this section.
		Updated Voltage TEP1 and TEP2 Encryption Examples with message format examples of the 23.x and 50.x messages for manual entry.
		• Updated the following EMV messages to indicate that the last [FS] character of a tag list is optional:
		  ◦ EMV '33.03.x' Authorization Request Message
		  ◦ EMV '33.04.x' Authorization Response Message
		  ◦ EMV '33.07.x' Terminal Capabilities Message
		  ◦ EMV '33.09.x' Set Tag Data Message
		  ◦ EMV '33.10.x' Get Tag Data Message

Rev 7.1		• Updated 11.x: Status Message with new '99' Idle status.

Manual Revision	Application Revision	Changes

	RBA 13.0.5 (Non-KP4)
	PK-RGENxx-1305x
	RBA 13.0.6 (KP4)
	PK-RGENxx-1306x
		• Removed software requirements from the following sections:
		  ◦ Contactless Key Card Support
		  ◦ Google Wallet Implementation
		  ◦ Softcard SmartTap Implementation
		  ◦ S1 Encryption
		• Created new transaction flow diagrams for G.3. eWIC Transaction Flows and D.2. PayPal Validation Flow.
		• Updated Application Selection adding a transaction flow diagram and a new process for selecting the application based on priority.
		• Page added for the IP Address and Port Display Variable for TCP-IP.
		• Updated the 63.x: Find File message to accurately reflect usage of the checksum flag to request the CRC32 checksum in the response message.
		• Added notation to the Pen Status field of the 08.x: Health Stat message stating that this value is not valid for Telium terminals.
		• Updated the BIN Lookup (stb.dat) table with the new '0005_0010' parameter used to determine if the service code is to be appended to the 19.x: BIN Lookup message.
		• Added notations to the Cash Back Configuration (cashback.dat) section stating that the iUN (unattended terminal) does not provide the cash back button when the Amount Verification screen is displayed.
		• Updated the Prompts page to address proper line break format.
		• Updated 07.x: Unit Data Request message to include the P07_RES_CLESS_INTERFACE_IS_SUPPORTED variable.
		• Added a new section describing the 58.x Terminal Discovery Message.
		• Updated the A.2. Support section with a complete list of Ingenico terminals which support the RBA Telium implementation.
		• Updated the A.7. Forms section with the list of terminals for which the Telium implementation of RBA provides form files.
		• Updated Compatibility Flags (compat.dat) and 12.x: Account Message pages for '0013_0021' parameter.
		• Included EMV Tag T9F6E in the EMV and Non-EMV Tags Transmitted in Host Interface Messages table.
		• Updated iConnectEFT constants for the following messages:
		  ◦ 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
		  ◦ 87.x On-Guard and KME Card Read Data
		• Added mention of varying display size and font size's affect on number of prompt lines displayed to the Signature (On-Demand) page.

Manual Revision	Application Revision	Changes
		• Added a Restrictions column to the 28.x: Set Variable Request table to specify any individual restrictions that apply. • Updated terminology from "contactless device" to "contactless reader" and "barcode reader [device]" to "barcode scanner". • Added Offline On-Demand Transaction Recommendations to the On-Demand page. • Updated message flow table in Retrieval Using Get Variable section. • Updated illustration in EFT Overview section. • Reformatted the Specific Attributes section, adding a table. • Reformatted the General Attributes section, adding a table. • Reorganized content in the Card Configuration Table section using one table. • Updated all EMV Host Interface Messages with more detailed failure codes in the Status field. • Added descriptions to the Error Code table on the Non-EMV Tag Definitions page. • Updated Appendix G: eWIC Implementation with new messages, ANSI spec information, and extensive reformatting. • Added more Encryption Requirements to the table. • Added data for On Demand and 50.x response messages with Mercury Encryption enabled.

Rev 7	RBA 13.0.3 (Non-KP4)
	PK-RGENxx-1301x
	RBA 13.0.4 (KP4)
	PK-RGENxx-1302x
		• Updated the Offline Remote Key Injection (RKI) Support section as follows:
		  ◦ All references to ".out" for file names were changed to ".rki".
		  ◦ Remote Key Injection can be enabled or disabled using the RKIenable.PGZ file.
		• Removed all references to CODM and CCODM forms and notated new methodology and forms for manual entry.
		• Renamed the parent section to 91.x: Print Message and included the 91.x: Barcode Printing section as a child page. Reorganized content in the parent page so that the request and response message tables are in the first section of that page.
		• Changed all references from "ECR" to "POS" in the 40.x: Survey Messages section.
		• Added new section Appendix H: Creating a .TGZ File.
		• Updated 11.x Status Request and Response messages per "Request Form Name" field and '20' smart card State Indicator.
		• Updated new information pertaining to invalid card swipe and cancelled manual card entry using the 23.x: Card Read Request (On-Demand) message. Included the "?" character in the table of card sources as "unknown/invalid" card type in the BIN Processing (allBins.dat, bin0.dat - bin20.dat) section as well.

Manual Revision	Application Revision	Changes

		• Updated Notes on EMV Configuration Parameters section with new information pertaining to ICS default values. Also updated the following section tables regarding ICS Default values:
		  ◦ Application ID (AID) Parameters in EMVCLESS.XML
		  ◦ Application ID (AID) Parameters in EMVCONTACT.XML
		  ◦ ICS Tags in EMVCLESS.XML
		  ◦ ICS Tags in EMVCONTACT.XML
		• Updated 28.x: Set Variable Request section with new variables 121, 122, 123, 203, 204, 205, 310 and 396.
		• Updated notes in BIN Processing (allBins.dat, bin0.dat - bin20.dat) table for bin1.dat and bin11.dat to clarify the BIN processing when PayPal is disabled. Stated in notes that bin1 and bin11 should not be reassigned to non PayPal cards.
		• Streamlined EMV tags and non-EMV tags into one combined table in the new EMV and Non-EMV Tags Transmitted in Host Interface Messages section.
		• Created new section on Mod-10 Checking and referenced this section from the Magtek Encryption and S1 Encryption sections.
		• Updated 09.x Card Status Message with transaction type "00" for unknown/invalid cards.
		  ◦ Updated Example Card Status Message table with '09.990200P' message for damaged/invalid cards.
		  ◦ Removed currently unsupported statuses S (MSR card swiped) and T (contactless card tapped).
		• Updated G.3. eWIC Transaction Flows with new values for example 09.x messages.
		• Added new section for EMV '33.00.x' Transaction Initiation Message.
		• Updated the Possible iWL250 Terminal and Base Connections illustration in the Installing the Bluetooth Pass-Through Service Utility - Relocated section.
		• Added new section on Dynamic Update of RSA-OAEP Public Keys.
		  ◦ Updated the 90.x: P2PE Data Message section with new '90.5', '90.6', and '90.7' messages to support this feature.
		  ◦ Updated the Security Parameters (security.dat) table with parameters '0091_0032' and '0091_0033'.
		• Updated the procedure for using the 60.x: Configuration Write message.
		• Updated Non-EMV Tag Definitions with new error response codes.
		• Provided a new section describing the EMV '33.07.x' Terminal Capabilities Message Flow.
		• Updated the EMV '33.00.x' Transaction Initiation Message section with new message format.
		• Updated the 07.x: Unit Data Request message with EMV/contactless kernel versions.

Manual Revision	Application Revision	Changes

		• Added the following new EMV messages:
		  ◦ EMV '33.01.x' Status Message
		  ◦ EMV '33.08.x' Set Variables Message
		  ◦ EMV '33.09.x' Set Tag Data Message
		  ◦ EMV '33.10.x' Get Tag Data Message
		• Updated default and valid encryption key slots for the following:
		  ◦ Mercury Encryption
		  ◦ Magtek Encryption
		  ◦ S1 Encryption
		• Updated the 20.x: Signature Message (On-Demand) section with the '20.02' signature accepted message.
		• Updated the default configuration parameter values for Monetra CardShield Encryption.
		• Updated the following EMV host interface message tables with iConnectEFT constants:
		  ◦ 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 '33.07.x' Terminal Capabilities Message
		• Updated options for the '0091_0031' EMV Configuration XML File Type parameter.

Manual Revision	Application Revision	Changes

Rev 6.1	RBA 12.1.1 (Non-KP4)
	PK-RGENxx-1211x
	RBA 12.1.2 (KP4)
	PK-RGENxx-1212x
		• Improved documentation of the Main Flow (mainFlow.dat) parameter '0007_0023'.
		• Added a new section for Google Wallet Implementation.
		• Modified the 16.x: Contactless Mode Request message and 17.x: Merchant Data Write message to include usage and format for Google Wallet.
		• Added a 17.x Merchant Data Write Message Usage Examples section.
		• Reorganized pages referenced from the RBA Low-Level Contactless Key Card Support section. All references and examples for the 16.x, 17.x, and 36.x messages have been moved directly into those sections.
		• Created separate sections for the 09.x Card Status Message and 09.x: Set Allowed Payments Message.
		• Added new section describing the EMV '33.07.x' Terminal Capabilities Message.
		• Updated transaction flow tables in the following sections:
		  ◦ EMV Purchase Transaction Flow
		  ◦ EMV Contactless Transaction Flow
		  ◦ Non-EMV Refund Transaction Flow
		  ◦ EMV Refund Transaction Flow
		  ◦ MSD Contactless Transaction Flow
		• Updated scenario for EMV Fallback.
		• Added new configuration parameter '0091_0031' to Security Parameters (security.dat) section which allows the user to load EMVCONTACT.XML and EMVCLESS.XML files from a locked or unlocked directory.
		• Updated the Available EMV Application IDs section with the new "Allow PIN Bypass" parameter which determines whether or not the customer is allowed to bypass PIN entry by pressing the "Cancel" key.
		• Updated the 09.x Card Status Message section to document the repurposed second "." (dot) character in the 09.x message.
		• Updated the Payment Type field and Transaction Amount field in the 04.x: Set Payment Type Request message.
		• Added a new section for Softcard SmartTap Implementation and updated the 16.x: Contactless Mode Request and 17.x: Merchant Data Write message sections accordingly.

Rev 6	RBA 12.0.1 (Non-KP4) PK-RGENxx-1201x RBA 12.0.2 (KP4)	• Updated 12.x: Account Message with correct parameters for discretionary data so that only valid numeric data is included in order to prevent errors and the device going offline. • Updated RBA documentation and provided an explanation for the '31.7' PIN Entry Response message which customers are now experiencing. • Updated documentation to provide clarification on the '00.' Offline Request message and '00.' Offline Response message. Improved the description for these messages. • Provided manual entry example for the 23.x: Card Read Request (On-Demand) message.

Manual Revision	Application Revision	Changes

	PK-RGENxx-1202x	• Specified configuration requirements for manual entry when using TDES encryption. Added descriptions to the following sections:
		  ◦ Data Format Prior to Encryption
		  ◦ Main Flow (mainFlow.dat)
		  ◦ TDES DUKPT Configuration
		• Replaced "Information Card" with "Non-Payment Card" throughout Rev 6 RBA Developer's Guide. Provided definition and examples for non payment card type. The following sections were updated:
		  ◦ 18.x: Non-Payment Card Message
		  ◦ 85.x: E2EE Information Card Message
		  ◦ Card Configuration (cards.dat)
		  ◦ Card Configuration Table
		  ◦ Non-Payment Cards
		• Updated RBA documentation with correct nomenclature for Telium devices covered in this manual. The following sections have been updated accordingly:
		  ◦ Definitions
		  ◦ Introduction
		  ◦ Minimum Requirements for RBA Customization
		• Updated Contactless Reader Configuration (cless.dat) parameters with new floor limit parameter.
		• Updated the syntax of the 24.x: Form Entry Request (On-Demand) and 70.x: Update Form Element Message messages for setting prompt text, button text, button visibility and form timeout.
		• Updated documentation with explanation of how to use the 62.x: File Write message to abort a previous file download.
		• Updated BIN range information in the BIN Processing (allBins.dat, bin0.dat - bin20.dat) section and streamlined Bin range tables into a single combined table.
		• Updated documentation with new "Enter PIN or Press Green for Credit" prompt. The following sections were updated:
		  ◦ 31.x: PIN Entry Messages (On-Demand)
		  ◦ Enter PIN or Press Green for Credit
		  ◦ Form Files (forms.dat)
		• S1 Encryption is now supported during standard flow card swipe screens. Documentation has been updated accordingly.
		• Updated RSA-OAEP and TransArmor Encryption public key exponent.
		• Updated the Security Prompts (SECURPROMPT.xml) section with the correct security prompts.
		• Updated the EPS P2PE Encryption Processing Examples section with examples for manual entry showing that both Track 1 and Track 2 are encrypted when using EPS P2PE encryption.

Manual Revision	Application Revision	Changes

		• Documented how the new '11.01' Status Request message will result in a status response message with the form name appended to it. Updated message table to include new optional field separator and field.
		• Cash Back Selection and Cash Back Other have been added to the No or Cancel Process flow diagram.
		• Updated the 61.x: Configuration Read section with description of how message works and provided examples.
		• Updated Magtek Encryption section noting that Track 3 data will not be returned.
		• Provided description of the 23.x: Card Read Request (On-Demand) message flow change depending on inclusion or exclusion of source field from the '23.' message for iConnectEFT.
		• Provided clarification in Security BIN (secbin.dat) section of RBA documentation on when to encrypt card data.
		• Updated documentation to show that track 1 is not handled during manual entry when using EPS encryption. The following sections were updated:
		  ◦ EPS (Element Payment Systems) P2PE Encryption
		  ◦ EPS P2PE Encryption Processing Examples
		  ◦ MSR Encryption Processing
		• Updated the Monetra CardShield Encryption section to convey that empty tracks are not encrypted; they are returned to the POS as blank. Only present tracks will be encrypted.
		• Updated the signature prompt ID field for the 25.x: Terms and Conditions Request (On-Demand) message.
		• Updated the Key Press ID options for the 25.x: Terms and Conditions Response message.
		• Updated Advertising Parameters (ads.dat).
		• Updated signature capture process and function of the "Cancel" and "Clear" keypad keys during signature capture process. Updated sections include:
		  ◦ 20.x: Signature Message (On-Demand)
		  ◦ No or Cancel Process
		• Updated Card Swipe and Card Swipe with Language Selection sections with new form images for the iUP250 showing the prompt for card "insert."
		• Changed all "RBA_SDK" references in document to "iConnectEFT" and added new Ingenico iConnectEFT Constants section which replaces the previous section titled RBA SDK Constants.
		• Updated Main Flow (mainFlow.dat) section describing use of configuration parameters to individually control select command display durations.
		• Added new form images for the iCM122 to the Form Contents and Descriptions section.
		• Updated the Voltage TEP1 and TEP2 Encryptions section with new TEP1 and TEP2 encryption examples using the 12.x: Account Message. The PAN is not encrypted when it is the only information included in this message.
		• Provided additional TEP1 encryption examples in the Voltage TEP1 and TEP2 Encryptions section using using 19.x, 23.x, 29.x and 50.x messages.

Manual Revision	Application Revision	Changes

		• Updated the 09.x: Set Allowed Payments Message section with the new message format. Provided description of usage and examples for smart card insertion and removal detection.
		• Updated Cash Back Flow parameter '0002_0010' in the Cash Back Configuration (cashback.dat) section with notation describing device function when option '2' is selected.
		• Merged EMV Implementation into this document as a new section. This section provides detailed information on how Ingenico payment devices process EMV transactions.
		• Updated MSD Contactless Transaction Flow section, specifying that "If EMV contactless mode is enabled (Contactless Device Mode parameter '0008_0001' = '9'), the device will automatically proceed with a contactless transaction when the MSD card is tapped."
		• Added a new page to the EMV Implementation section of the RBA Developer's Guide describing configuration and initiation of the EMV On-Demand Flow transaction flow. Updated the EMV '33.00.x' Transaction Initiation Message format.
		• Moved Appendix H (iSMP Integration with iPOD Touch) to Telium Troubleshooting Guide.
		• Moved the Telium Download section from the Appendices to the Telium Download User's and Developer's Guide.
		• Added the following new sections to the EMV Configuration Parameters section:
		  ◦ Application ID (AID) Parameters in EMVCONTACT.XML
		  ◦ Application ID (AID) Parameters in EMVCLESS.XML
		  ◦ Certificate Authority Public Keys in EMVCONTACT.XML
		  ◦ ICS Tags in EMVCONTACT.XML
		  ◦ ICS Tags in EMVCLESS.XML
		• Updated device names (iSMP, iSMPc, and iCMP) in Form Contents and Descriptions section.
		• Updated the following sections to convey that Contactless Device Mode parameter ('0008_0001') must be set to '9' which will enable contactless mode for EMV.
		  ◦ Contactless Smart Card (EMV) and Swipe
		  ◦ Contactless Smart Card (EMV) and Swipe with Language Selection
		  ◦ EMV Contactless Transaction Flow
		• Updated all encryption sections with the notation stating the RBA does not support encryption /masking of PANs containing less than 9 digits.
		• Updated the PayPal Forms section, adding the iUN250 to the list of devices which do not support PayPal.
		• Changed verbiage pertaining to Ingenico device throughout document. The term "device" replaces "PIN pad" and "terminal" except where used in parameter naming conventions.
		• Add new EMV Tag Data Format section which describes the format for EMV tags and provides usage examples.

Manual Revision	Application Revision	Changes

		• Updated the following EMV messages:
		  ◦ EMV Transaction Initiation Message
		  ◦ EMV Track 2 Equivalent Data Message
		  ◦ EMV Authorization Request Message
		  ◦ EMV Authorization Response Message
		  ◦ EMV Authorization Confirmation Response Message
		• Added new section to the RBA documentation describing EMV Configuration and Flow. Included a page describing PIN Block Tag Format in Authorization Request Message.
		• Updated the device gallery images and device names in the Introduction section.
		• Added new section for iCMP Button IDs and Images.

Rev 5	RBA 11.0.1 (Non-KP4)
	PK-RGENxx-1101x
	RBA 11.0.2 (KP4)
	PK-RGENxx-1102x
		• Added new '0007_0046' parameter to Main Flow (mainFlow.dat) table to support daily reboot time in 24-hour format.
		• Included a description the '0007_0029' configuration parameter setting for manual entry for the following forms:
		  ◦ Smart Card (EMV) and Swipe
		  ◦ Contactless Smart Card (EMV) and Swipe with Language Selection
		  ◦ Contactless Smart Card (EMV) and Swipe
		  ◦ Contactless Smart Card (EMV) and Swipe with Language Selection
		• Added content to the Cash Back Selection section stating that on the cashback screen, the displayed "No" button selects no cash back while the physical "Cancel" key cancels the current payment.
		• Specified that when using the 60.x: Configuration Write message to change configuration, the '60.' message must be followed by a '00.x' Offline message or '01.x' Online message to retain the configuration change following reboot.
		• Specified that when using the 62.x: File Write message to download large files, the process may take up to several minutes.
		• Added additional ASCII File Number field to the 62.x: File Write message.
		• Added Vendor ID and Product ID values to the VID and PID Settings for HID and CDC Communications table for the iUP250 device.
		• Added error code '8' (error unzipping file) to the 62.x: File Write response message.
		• Documented new text field maximum of 500 characters in the 70.x: Update Form Element Message.
		• Reformatted content and added example to 70.x: Update Form Element Message pertaining to the use of the ID and new field data.
		• Updated No or Cancel Process section with new flow diagram which includes updated form images.
		• Added new section describing Ingenico iConnectEFT Constants.
		• Improved documentation 26.x: Run Script Request (On-Demand) message and provided description of tag script parameters.

Manual Revision	Application Revision	Changes

		• Added structure and usage descriptions for the 93.x: Terminal Authentication Messages.
		• Added configuration items to Security Parameters (security.dat) section to enable/disable encryption for driver's licenses and non-standard cards.
		• Added description for usage of 21.x: Numeric Input Request Message (On-Demand) and 27.x: Alpha Input Message (On-Demand) to send encrypted clear entry data. Updated parameters '0091_0019' through '0091_0022', '0091_0026' and '0091_0027' in the Security Parameters (security.dat) to support this feature.
		• Added description/example of 27.x: Alpha Input Message (On-Demand) message structure.
		• Updated Mercury Encryption section, stating the requirement for all 3 tracks, including start and end sentinels.
		• Updated form description tables for the following forms:
		  ◦ Contactless Smart Card (EMV) and Swipe with Language Selection
		  ◦ Contactless Smart Card (EMV) and Swipe
		  ◦ Smart Card (EMV) and Swipe
		  ◦ Smart Card (EMV) and Swipe with Language Selection
		  ◦ Contactless Card Swipe with PayPal Selection
		  ◦ Contactless Card Swipe with PayPal and Language Selection
		  ◦ Card Swipe with PayPal
		  ◦ Card Swipe with PayPal and Language Selection
		  ◦ Approved/Disapproved
		  ◦ Pre-Sign
		• Added new section for Offline Remote Key Injection (RKI) Support.
		• Added the account data origin option of "A" (to enable the application to distinguish between manual entry and an account message) to the 19.x: BIN Lookup section.
		• Added excerpt from config.dfs file and updated 3-didit card codes for Debit and Credit cards.
		• Updated the forms section on Signature (On-Demand) describing the change in functionality of the "Cancel" button and "Clear" button once signature is initiated.
		• Added new '0091_0030' configuration parameter to the table in the Security Parameters (security.dat) section. This parameter is used to block account messages when encrypting MSR data.
		• Added new item "EFTERROR" to the sample TDA.XML file in C.5. Editing the TDA.XML File. Added new section C.7. TDA Error Reporting During EFT Loads which lists error codes and provides examples.
		• Updated 24.x: Form Entry Request (On-Demand) message section with new maximum radio button character length and check box length.
		• Added content to the 28.x: Set Variable Request section and Contactless Reader Configuration (cless.dat) section regarding the use of the '28.' message to temporarily change contactless settings, and the '60.' Configuration Write message to permanently change contactless settings.

Manual Revision	Application Revision	Changes
		• Updated iMP350/iMP352 form images in Form Contents and Descriptions section to show battery icon. Added notation to PayPal forms and other forms where not supported by iMP350/iMP352. • Updated 28.x: Set Variable Request and Retrieving MSR information using the 29.x (Get Variable) Message sections with new variables '395' and '414' which are used to determine whether or not card data has been encrypted. • Updated configuration parameter '0091_0016' in the Security Parameters (security.dat) section to include new options for TransArmor encryption target. • Added excerpt from config.dfs file to Card Configuration (cards.dat) section and documented card source identification. • Updated encryption examples in the Voltage TEP1 and TEP2 Encryptions section.

Rev 4	RBA 10.0.1 (Non-KP4)
	PK-RGENxx-1001x
	RBA 10.0.2 (KP4)
	PK-RGENxx-1002x
		• Added 0011_0008 card type H: EMV (key ID = 72) to Card Configuration (cards.dat) section.
		• Added iSC480 form images to Form Contents and Descriptions section.
		• Replaced iSC250 and iSC350 form images with new format images (red and grey) in the Form Contents and Descriptions section.
		• Replaced iPP320 and iPP350 form images in the Form Contents and Descriptions section. New images have red dashed lines removed and eliminated instances of text out of frame.
		• Added the following new forms:
		  ◦ Contactless Card Swipe with PayPal Selection
		  ◦ Contactless Card Swipe with PayPal and Language Selection
		  ◦ 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
		• Updated table in Supported Encryption Methods section to show devices which rely solely on non-KP4 keys will no longer be supported by Telium.
		• Added expected behavior for Key Sequence Number when using EPS to EPS (Element Payment Systems) P2PE Encryption page and EPS P2PE Card Swipe Examples page.
		• In Supported Encryption Methods table, changed "Ingecrypt" to "On-Guard." Removed incorrect comment pertaining to support for this encryption method.
		• Retitled On-Guard Encryption page and removed references to "Ingecrypt."
		• Added new images to the Post-Sign forms section to illustrate pre-signature and post-signature forms.
		• Add new button images for the iSC250 Button IDs and Images, iSC350 Button IDs and Images, and iSC480 Button IDs and Images.
		• Added new battery life health stat response message to 08.x: Health Stat section.
		• Generated and added iMP350 forms to Form Contents and Descriptions section.

Manual Revision	Application Revision	Changes

		• Added Button IDs for cashback options and reformatted Button ID and Image tables for the following:
		  ◦ 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
		• Updated device image gallery in Introduction with new images for the iSMP and iCMP.
		• Added content to the 25.x: Terms and Conditions Request (On-Demand) message section describing acknowledgement of the Cancel, Clear, and Enter keys when the generic Terms and Conditions form is displayed.
		• Removed RBA Test Application from Appendices, which was formerly located in Appendix F. Renamed appendices G through I to F through H, respectively.
		• Removed content from the Call Scheduling section and referenced documentation in the Telium Download User's and Developer's Guide.
		• Added reference to the RBA Testing Tool in the Introduction section.
		• Added new section and table for EMV Flags (emv.dat).
		• Updated Voltage TEP1 and TEP2 Encryptions section with additional TransArmor parameters.
		• Updated the Main Flow (mainFlow.dat) section with additional parameter '0007_0032' for Bluetooth pairing.
		• Updated Form Files (forms.dat) section with added form files.
		• Updated Security Parameters (security.dat) section with TransArmor parameters.
		• Added new section with MAC Entry (mac.dat) parameters.
		• Updated Bin Range table in BIN Processing (allBins.dat, bin0.dat - bin20.dat) section with important notes on PayPal Discover cards and PayPal physical cards.
		• Edited form files in the Form Contents and Descriptions section so that file names matched those listed in the Form Files (forms.dat) table. Linked table elements to form files.
		• Added new Approved/Disapproved section with images to Form Contents and Descriptions section. Updated Form Files (forms.dat) table with new form information.
		• Added Card Swipe with PayPal section with images to Form Contents and Descriptions section. Updated Form Files (forms.dat) table with new form information.
		• Added Card Swipe with PayPal and Language Selection section with images to Form Contents and Descriptions section. Updated Form Files (forms.dat) table with new form information.
		• Added new Pre-Sign section with images to Form Contents and Descriptions which describes the function of the new generic pre-sign form. Updated Form Files (forms.dat) table with new form information.

Manual Revision	Application Revision	Changes

		• Added new Post-Sign section with images to Form Contents and Descriptions which describes the function of the new generic pre-sign form. Updated Form Files (forms.dat) table with new form information.
		• Updated parameters in Security Parameters (security.dat) table for generic message encryption using the barcode encryption key.
		• Documented the new generic message encryption feature using the 21.x: Numeric Input Request Message (On-Demand) and 27.x: Alpha Input Message (On-Demand) to return encrypted data using the same key used for the 95. Barcode Data message.
		• Updated Status Messages (status.dat) table with new parameters for Rewards, Button Input, Generic Display Message, Get Key Press, Internal WIC PIN Entry, and WIC Insert Card.
		• Updated 62.x: File Write Request message with record type "9" which indicates that the current File Write request process needs to be aborted.
		• Corrected device keys pressed in C.1. Accessing the Telium Download Application section for accessing FUNCTIONS menu.
		• Corrected step 2 menu option by creating new image which includes "3 - TMS Settings" in Setting Communication Type in section C.2. Configuring device Communication Settings.
		• Included iSC250 and iSC480 for MagicBox Serial in section C.2. Configuring device Communication Settings, and removed reference to iSC250 not supporting USB-HID.
		• Updated default values for the following parameters in the Main Flow (mainFlow.dat) table:
		  ◦ 0007_0032
		  ◦ 0007_0034
		  ◦ 0007_0035
		• Updated default values for parameters '0007_0032', '0007_0034', and '0007_0035' in the Main Flow (mainFlow.dat) table. Also reformatted entries in table where needed.
		• Corrected example data values in the On-Guard Encryption Configuration section, and reformatted the section as needed.
		• Appended manufacturer serial number to 07.x: Unit Data Request and 08.x: Health Stat messages.
		• Appended MSC field to 23.x: Card Read Request (On-Demand) message.
		• Added new section on Contactless Key Card Support.
		• Added updated DFS Data Index values to the Form Files (forms.dat) table. Updated the following form file names in the Form Contents and Descriptions section, and updated the associated DFS Data Index in the form file table:
		  ◦ 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
		• Corrected default values for '0099_0003' and '0102_0001' parameters in BIN Processing (allBins.dat, bin0.dat - bin20.dat) section. Also reformatted Transaction Codes table by adding sub-table for string characters. removed duplicate '0099_0002' entry in allBins.dat table.

Manual Revision	Application Revision	Changes

		• Updated the 62.x: File Write message to indicate that the first two bytes in the "reserved" data field now contain an ASCII file segment number.
		• Updated 63.x: Find File message section with implementation of CRC value retrieval using this message with optional appended [FS] character and return field.
		• Updated forms pages in Form Contents and Descriptions section to include the iMP352 with the iMP350.
		• Removed and referenced duplicate forms in Form Contents and Descriptions section.
		• Created the following new subsections in the Form Contents and Descriptions section for organizing related forms:
		  ◦ Card Swipe Forms
		  ◦ Cash Back Forms
		  ◦ Contactless Enabled Forms
		  ◦ Payment Forms
		  ◦ PayPal Forms
		  ◦ Signature Forms
		  ◦ Smart Card (SMC) and EMV Forms
		  ◦ Terms and Conditions Forms
		• Restored Encryption Data Returned to the POS section to document, under RSA-OAEP and TransArmor Encryption section.
		• Updated 83.x: E2EE Enable Message key number options in request message, and reformatted text on this page.
		• Updated "Beep Volume" parameter in Main Flow (mainFlow.dat) table to include iSMP. Also reformatted table content.
		• Incorporated 41.x Card Read Message into document.
		• Added the Form Name field and PayPal Key Type option to the 31.x: PIN Entry Messages (On-Demand) request message. Added text and notations regarding the use of the '31.' message to support forms for capturing PayPal passcodes.
		• Updated the Generic TDES DUKPT Encryption section in RBA Developer's Guide to explain auto-incrementation of the KSN after every 10 transactions.
		• Updated '0007_0030' parameter in the Main Flow (mainFlow.dat) table to indicate that with EPS encryption the expiration date and CVV will be left blank if not entered manually.
		• Updated 23.x: Card Read Request (On-Demand) message "Source of card read" field to include option "H" for manual entry.
		• Included option "R" for manual entry Track 1 and Track 2 in 50.x: Authorization Request message "Account Data Source" field, and reformatted page text.
		• Added [FS] character between Prompt Index Number and ASCII character "B" in the 24.x: Form Entry Request (On-Demand) message. Also reviewed message content and updated "Offset" values in table.

Manual Revision	Application Revision	Changes

		• Added a notation to the "Id of field to update" field in the 70.x: Update Form Element Message: 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.
		• Added reference to 34.x: Save and Restore State Messages in section on 23.x: Card Read Request (On-Demand) regarding nested On-Demand messages.
		• In the Examples of Format Specifiers section;
		  ○ Aligned error beep with first "5" in table, not "clear".
		  ○ Changed format from %M to %m in comment pertaining to minimum numbers.
		• Replaced Standard Process Flow diagram with new diagram which includes updated forms, and restructured diagram layout.
		• Added notations to table in BIN Processing (allBins.dat, bin0.dat - bin20.dat) section. There are up to 30 BIN ranges. All BIN ranges can be configured by the customer.
		• Removed duplicate '0099_0002' entry in table in BIN Processing (allBins.dat, bin0.dat - bin20.dat) table.
		• Updated Magtek Encryption section to show that all valid '0007_0029' parameter values are acceptable for Magtek encryption. Also reformatted text int his section.
		• Added new section Reserved Form Buttons which lists reserved BmpButton IDs for all forms.
		• Updated button ID for iSC250 in iSC250 Button IDs and Images section.
		• Added more description, links, and message table for the 36.x Notification of Command Execution.
		• Added notations stating that the 24-hour reboot function only applies to production units. Added notations to 28.x: Set Variable Request and Main Flow (mainFlow.dat) sections.
		• Updated key press value "L" for Language in the following sections:
		  ○ iSC250 Button IDs and Images
		  ○ iSC350 Button IDs and Images
		  ○ iSC480 Button IDs and Images
		• Corrected parameter values in the 25.x: Terms and Conditions Request (On-Demand) response message table:
		  ○ Removed "3" Cancelled from Signature on Accept field.
		  ○ 'Y' is returned to indicate "Accept."
		  ○ 'N' is returned to indicate "Decline."
		• Created new form flow illustration for No or Cancel Process which includes updated form images and device.
		• Created a new process flow illustration in the Signature Forms section for signature capture.

Rev 3	RBA 9.0.1 (Non-KP4)	• Corrected 25.x table message per DOC-133. See changes on page 25.x: Terms and Conditions Request (On-Demand). • In the 19.x: BIN Lookup section, added manual entry response of "19.T000000699999998000009901" to Sample Requests table.

Manual Revision	Application Revision	Changes
	PK-RGENxx-0901x RBA 9.0.2 (KP4) PK-RGENxx-0902x	• Added '11.' status values/messages 11.16 - 11.19 to Status Response Format table on 11.x: Status Message page, and updated Status Messages (status.dat) table with corresponding DFS Data Index values for Menu, Textbox, EMV, and WIC. • Included iSMP350 device in Definitions table for devices covered in this manual. • On the Generic TDES DUKPT Encryption page, edited text to reference RBA instead of UPOS, OPOS, and UIA. Also added some content and defined acronyms for TDES DUKPT. • Added DFS Data Indexes 0007_0032 through 0007_0044 to mainflow.dat table on the Main Flow (mainFlow.dat) page. • Removed 'infinite timeout' option from 0006_0001 overall timeout for PIN entry in pin.dat table on PIN Entry (pin.dat) page. • Added new Call Scheduling features section which describes the process for incorporating call scheduling into device using U.S. Retail applications. • Added to documentation an alternate on demand method which enables a new set of messages that allow you to save and restore a state. Added new 34.x: Save and Restore State Messages and updated the Communication Messages page. Updated mainflow.dat table on Main Flow (mainFlow.dat) page with new parameters for this feature under DFS Data Index 0007_0015. • Added reference in Appendix C: Configuring the Application on upgrade using the TDA 2.6.0.0002. Created new page C.6. Upgrading Using TDA 2.6.0.0002 and incorporated the procedure for upgrade using EFT-download with TDA 2.6.0.0002. • Added descriptions to 28.x: Set Variable Request variable table for default language and current language for clarification. • Updated Security Parameters (security.dat) table with new '0091_0018' parameter to enable/disable device authentication. • Updated message format table for 50.x: Authorization Request to indicate that if a transaction amount in cents is entered with less than 3 digits, '0's will be prepended to the amount to ensure that 3 digits are sent. • Created new section on EPS (Element Payment Systems) P2PE Encryption which includes EPS P2PE card swipe and encryption examples. • Created new page which provides an overview of 93.x: Terminal Authentication Messages. • Renamed 90.x: P2PE Data Message from previous name of "90.x: MSR Encryption Support Message". Edited text where needed to indicate that this applies to all cardholder data, and is not specific to MSR encryption. • Added a new page under MSR Encryption which describes the process for Requesting the PIN Block Using the Masked PAN. • Add new 51.x: Beep On-Demand Message page which describes the message format for the beep on demand message. • Corrected offset values in 62.x: File Write message table. • Updated A.9. RBA Configuration (config.dfs) table to incorporate result display control features for 20.x, 21.x, 23.x, 24.x, 25.x, 27.x, and 31.x commands. • Updated contactless mode parameters in Contactless Reader Configuration (cless.dat) table.

Manual Revision	Application Revision	Changes

		• Added parameter to PIN Entry (pin.dat) table to enable 0-length PINs in 31. message. Added reference to this feature on 31.x: PIN Entry Messages (On-Demand) page.
		• Added new section for On-Guard Encryption. Modified 31.x: PIN Entry Messages (On-Demand) table in accordance with addendum for On-Guard encryption.
		• Added new section for LCD Display Preservation for Telium Terminals.
		• Added new section for iSMP Integration with iPOD Touch.
		• Edited 62.x: File Write section and specified that the 62.x message is intended to update single files whereas larger amounts of data should be updated via IBMEFT download or by using TMS. Added a notation on the 97.x: Reboot page which references the 62.x message.
		• Added new section on external display interface for Telium devices. refer to Appendix F. External Display for Telium Terminals.
		• Updated 50.x: Authorization Request message with response "A" which is received when account information is being sent by the 12.x message.
		• Added iSC480 documentation to the following sections:
		  ◦ 07.x: Unit Data Request
		  ◦ 15.x: Soft Reset Message
		  ◦ C.1. Accessing the Telium Download Application
		  ◦ Configuring Signature Retrieval
		  ◦ Introduction
		  ◦ Retrieval Using Get Variable
		  ◦ Signature Format
		  ◦ Signature Items (sig.dat)
		• Added "Cancelled" option to 25.x: Terms and Conditions Request (On-Demand) response message.
		• Added "Bad read" parameter '0008_0008' to table in Contactless Reader Configuration (cless.dat) section.
		• Added content to Button IDs and Images page to the effect that, during the normal application flow, any button not handled by the application will result in a 24. message to be sent to the POS.
		• Modified the 20.x: Signature RESPONSE Message (On-Demand) message format table to include new Status and Data input field parameters. The RBA now returns the key pressed in the event the on-demand signature request is cancelled so that the POS can take the appropriate action.

Manual Revision	Application Revision	Changes
Rev 2	RBA 8.0.1 (Non-KP4) PK-RGENxx-0801x RBA 8.0.2 (KP4) PK-RGENxx-0802x	• In the table on 19.x: BIN Lookup page, updated the code value of the field "Indicates from where account was derived" to be the same as the 50.x response code field "Account Data Source." • In the table on the MSR Encryption Processing page, for manually entered data, add the characters "% M" to the "Original" data to correctly indicate manual entry. • On the Card Configuration (cards.dat) page, corrected the values for the "Required Track" and reformatted track component by organizing values using bullets. • On the 70.x: Update Form Element Message page, updated the table to change visibility of buttons on the form on run time. • Created new section on Generic TDES DUKPT encryption. See page Generic TDES DUKPT Encryption. • On the 13.x: Amount Message page, changed reference to Amount Message Format table, and changed field length from 9 maximum to 16 maximum. • In the table of examples on the Examples of Format Specifiers page, removed the second column which was a textual representation of the first column. All information needed is provided in the first column. This was done to eliminate the duplication and also to format the table for PDF export. • Reduced image size on Standard Process Flow page in order to fit image on one page in PDF generated document. • Reformatted Sample Response table on 23.x: Card Read Request (On-Demand) page for PDF generated document. • Reorganized forms on all child pages of the Form Contents and Descriptions page to prevent images from running off-page. • Reduced table width on MSR Encryption Processing page to format for PDF generated document. • Reformatted table width on Voltage TEP1 and TEP2 Encryptions page for PDF generated document. • Reduced table width on Data Returned to the POS Application page for PDF generated document. • Resized image on E.2. PayPal Validation Flow for PDF generated document. • Resized security.dat table on Security Parameters (security.dat) page for PDF generated document. • On Signing Requirements for .DAT File Changes page, created new flow diagram for MSR Encryption Configuration Process. • Reformatted tables on BIN Processing (allBins.dat, bin0.dat - bin20.dat) page for PDF generated document. Original tables had issues with PDF.
Rev 1	RBA 7.0.1 (Non-KP4) PK-RGENxx-0701x RBA 7.0.2 (KP4)	Added Appendices page. Added iSC480 Button IDs and Images page. Added S1 Encryption page. Added Communication Settings page. Added Telium Terminal Information page. Added VID and PID Settings for HID and CDC Communications page. Added C.4. How to Change App Entry Code (iUN250) page.

Manual Revision	Application Revision	Changes
	PK-RGENxx-0702x	Added C.5. Editing the TDA.XML File page. Added 94.x and 95.x: Barcode Configuration Messages page. **10_0_58_6** Revisions to Document: • Added RBA_SDK info for each message format. • Corrected USB HID devices specification on the Communication Settings page. • Incorporated and reformatted the new section on the new S1 Encryption page. • Updated table in Supported Encryption Methods section to include TDES DUKPT Generic encryption. • Updated button images for the iSC250, iSC350, and iSC480. • Moved all appendices to the new Appendices section. • Moved the Communication Settings page to Telium Terminal Information page, as a child page. • Renamed Appendices E through H as Appendices D through G , respectively, since the contents of the now deleted Appendix D were moved to a new page. • Combined the HID table and CDC table into one table on the VID and PID Settings for HID and CDC Communications page, which is now a child page of the Telium Terminal Information page. • Incorporated iSC480 parameters into the combined table on the VID and PID Settings for HID and CDC Communications page. • Added ECC Encryption parameters to table on Security Parameters (security.dat) page. • Added documentation to the Barcode Parameters (barcode.dat) page. • Updated signature encoding format parameters to the table on the Signature Items (sig.dat) page. • Added a new section to the 24.x: Form Entry Request (On-Demand) page, which is titled "Displaying Text Data." • On the 30.x: Advertising Request Message (On-Demand) page, changed from "11.x request is received and the response contains numerical value 06 and text Advertising.” to "11.x request is received and the response contains numerical value 15 and text Advertising.” under 30.x execution. • Added a new section to the Monetra CardShield Encryption page, which is titled “Initiating an Encrypted Swipe/Tap/Manual Entry.” This section addresses the newly added capability of processing card swipes from the standard flow "swipe" screen without error. • Updated the table on the Security Parameters (security.dat) page by adding a new function key sequence in DFS Data Index 0091_0023 for accessing the Telium Manager on iUN devices which have not been commissioned. • Added a new child page C.4. How to Change App Entry Code (iUN250) to Appendix C, which makes reference to the new function key sequence used to access the Telium Manager on iUN devices which are not commissioned. • On the 12.x: Account Message page, changed the description for the M offset character in the table from ASCII character "=" to ASCII character FS.

Manual Revision	Application Revision	Changes
		• On the 23.x: Card Read Request (On-Demand) page, changed "a response with cancel state is sent out (23.1)." to "a response with cancel state is sent out (23.2)." • On the Card Configuration (cards.dat) page, changed the description for Hex Representation '002' from 'Contactless' to 'MSR' in the example table. • On the new C.5. Editing the TDA.XML File page, added an example TDA.XML file and instructions on how to edit the file to change communications parameters. Edited the example TDA.XML file and incorporated the new SSL parameters for Ethernet communications. • On the new 94.x and 95.x: Barcode Configuration Messages page, created 10 child pages which contain all of the message tables for the 94.x and 95.x messages. • On the new 94.x and 95.x: Barcode Configuration Messages page, added tables for barcode configurations and barcode actions. • On the Security Parameters (security.dat) page, changed the default value for DFS Data Index 0091_0002 to '4'. • Modified message tables so that all tables use the starting character 'M' after the first variable length for consistency.

Refer to the Telium Retail Base Application (RBA) Developer's Guide Rev 3 for the revision history prior to Rev 1.