TDL™
Triton Dynamic Language™

Communications Definition

PRELIMINARY

Version TDL0.54 - Part 1

February 16, 2022

Triton Systems of Delaware, LLC
21405 B. Street
Long Beach, MS 39560
www.triton.com

Copyright © 2022 by Triton Systems of Delaware, LLC. All rights reserved.

PRELIMINARY

TABLE OF CONTENTS

1 TDL SPECIFICATION ORGANIZATION 12

2 REVISION HISTORY 12

3 INTRODUCTION 15

4 DEFINITIONS 15

5 RELATED DOCUMENTS 16

6 SYSTEM DEFINITION 16

6.1 Layered Communication Model 16
6.1.1 TCP/IP Protocol 17
6.1.1.1 Optional TDL SDU Format 18
6.1.2 Visa II Protocol 18

6.2 Security 18
6.2.1 Message Encryption 18
6.2.2 Message Authentication 19

7 MESSAGE DEFINITIONS 20

7.1 TDL™ Packet Data Format 20
7.1.1 TDL Packet Structure 20

7.2 TDL Data Definitions 21
7.2.1 Basic ASCII Represented Values 21
7.2.2 Binary Data 21
7.2.3 Message Error 22
7.2.3.1 Error Code 22
7.2.3.2 Error Description 22

7.3 TDL Packet Process at Host 22
7.3.1 Transaction Options 24

PRELIMINARY

Highly Confidential: Property of Triton Systems – Do Not Disclose

7.3.2 OAR Data 25
7.3.3 Custom Screen Text 26
7.3.4 Host Withdrawal Surcharge Amount 26
7.3.5 Host Balance Surcharge Amount 26
7.3.6 FX-DCC 27
7.3.6.1 FX-DCC Eligibility Determination 27
7.3.6.2 FX-DCC Offer Determination 28
7.3.6.3 FX-DCC Working Key Acquisition 29
7.3.6.4 FX-DCC Transaction Processing 29
7.3.6.5 FX-DCC Reversal Processing 30
7.3.6.6 FX-DCC Close Transaction Processing 31
7.3.6.7 FX-DCC Supplemental TLVs 32
7.3.7 TDL Transaction Data 38
7.3.7.1 Terminal Identification 38
7.3.7.2 Transaction Type Identification 38
7.3.7.3 Account Type Definition 39
7.3.7.4 Transaction Sequence Number 39
7.3.7.5 Track Data Group 39
7.3.7.6 Encrypted PIN Block 40
7.3.7.7 Transaction Amount 40
7.3.7.8 Transaction Date 40
7.3.7.9 Authorization Code 40
7.3.7.10 Surcharge Amount 40
7.3.7.11 ATM Transaction Record 40
7.3.7.12 Account Number 41
7.3.7.13 Authorization Number 41
7.3.7.14 Ledger Balance 41
7.3.7.15 Available Balance 41
7.3.7.16 Reversal Status 41
7.3.7.17 Reversal Reason 42
7.3.7.18 Receipt Message 42
7.3.7.19 Deposit Request Message 43
7.3.7.20 Deposit Transaction Response 43
7.3.7.21 Deposit Totals Reporting 43
7.3.7.22 Extended Transaction Record Data 44
7.3.7.23 Withdrawal Transaction Record 44
7.3.7.24 Transfer Transaction Record 45
7.3.7.25 Dispensed Amount 45
7.3.7.26 Dispense Status 45
7.3.7.27 Dispensed Count by Cassette 45
7.3.7.28 Rejected Count by Cassette 45
7.3.7.29 Cassette A 46
7.3.7.30 Cassette B 46
7.3.7.31 Cassette C 46
7.3.7.32 Cassette D 46
7.3.7.33 Notification Info 46
7.3.7.34 Message Type 46
7.3.7.35 Recipient Address 47
7.3.7.36 Unique Identifier 47
7.3.7.37 Disable SMS Command 47

PRELIMINARY

7.3.7.38 Working Key 47
7.3.7.39 Language Selected 48
7.3.7.40 Enable SMS Message 48
7.3.7.41 Enable SMS Response 48
7.3.7.42 Enable SMS 48
7.3.7.43 Opt In Flag 49
7.3.7.44 Business Date 49
7.3.7.45 ATM Report Record 49
7.3.7.46 EMV Data 49
7.3.7.47 Triton Specific EMV Data 49
7.3.8 Prepaid/Gift Card Services 49
7.3.8.1 Credentials 50
7.3.8.2 Gift Card Operation Failure Information 50
7.3.8.3 Gift Card Activation 50
7.3.8.4 Gift Card Deactivation 52
7.3.8.5 Performance Monitoring Statistics 53
7.3.9 Cardless Transactions 53
7.3.9.1 Cardless Transaction Request 53
7.3.9.2 Cardless Transaction Request Example 54
7.3.9.3 Cardless Transaction Response 55
7.3.9.4 Cardless Transaction Response Example 56
7.3.9.5 Session Key Request 57
7.3.9.6 Session Key Request Example 57
7.3.9.7 Session Key Response 58
7.3.9.8 Session Key Response Example 59
7.3.10 Digital Currency Purchase Transactions 59
7.3.10.1 Digital Currency Purchase Request 60
7.3.10.2 Digital Currency Purchase Request Example 61
7.3.10.3 Digital Currency Purchase Response 61
7.3.10.4 Digital Currency Purchase Response Example 62
7.3.10.5 Digital Currency Commitment Request 63
7.3.10.6 Digital Currency Commitment Request Example 64
7.3.10.7 Digital Currency Commitment Response 65
7.3.10.8 Digital Currency Commitment Response Example 65

8 ALLOCATED TAG ENUMERATIONS 66

9 TDL EXAMPLES 72

9.1 Customer Specific Surcharge (TDL Legacy) 72

9.2 Overdraft Fee Disclosure (TDL Legacy) 72

9.3 Multiple User Accounts (TDL Legacy) 73

9.4 Terminal Management Examples 73

9.5 Gift Card Services Examples 73

PRELIMINARY

9.5.1 Gift Card Activation Request PDU 73
9.5.2 Gift Card Activation Response PDU Success Case 74
9.5.3 Gift Card Activation Response PDU Failure Case 75
9.5.4 Gift Card Deactivation Request PDU 77
9.5.5 Gift Card Deactivation Response PDU Success Case 77
9.5.6 Gift Card Deactivation Response PDU Failure Case 78

10 LEGACY 80

10.1 Legacy Message Extensions 80
10.1.1 Extended Transaction Codes 80
10.1.2 Extended FIDs 80
10.1.3 Tag and Length Conventions For TDL Legacy Payloads 80

10.2 MODEM COMMUNICATION PARAMETERS 81
10.2.1 Control Character Table 83
10.2.2 Transaction Request Message 84
10.2.3 Transaction Response Message 86
10.2.4 Configuration Table Download Request Message 88
10.2.4.1 TR-31 Key Block Support 89
10.2.5 Configuration Table Download Response Message 90
10.2.5.1 TR-31 Key Block Support 91
10.2.6 Host Totals Download Request Message 92
10.2.7 Host Totals Response Message 94
10.2.8 Reversal Request Message 95
10.2.9 Reversal Response Message 97
10.2.10 RKT Request Message 98
10.2.11 RKT Response message 99
10.2.12 Transaction Information Request 101
10.2.13 Transaction Information Response 102
10.2.14 Status Monitoring Field 103
10.2.15 Product Code Table for Program / Table Version Numbers 106
10.2.16 Product Code Table for EPROM Version Numbers 106
10.2.17 Transaction Code Table 107
10.2.18 Authorization Codes Table 109
10.2.19 Miscellaneous Field ID Descriptions 111
10.2.20 Valid Miscellaneous Field ID Table 122

10.3 PROTOCOL SPECIFICATION 125
10.3.1 Normal Message w/EOT 126
10.3.2 Normal Message w/o EOT 127
10.3.3 Correctable Request and Response Transmit Error 128
10.3.4 Uncorrectable Request Transmit Error 129
10.3.5 Uncorrectable Response Transmit Error 130
10.3.6 Terminal ACK Not Received By Host 131
10.3.7 Terminal Does Not Acknowledge Response From Host 132
10.3.8 Host Does Not Respond to ACK from Terminal (Default Mode) 133
10.3.9 Host Does Not Respond to Request Message 134
10.3.10 Terminal Does Not Respond to Host ENQ 135

PRELIMINARY

10.3.11 Host Does Not Answer Terminal Call 136

10.4 ANSI PIN Block Format 138
10.4.1 Clear-Text PIN Block Format 138
10.4.2 Cipher-Text PIN Format 139

10.5 MAC Protocol Addendum 141
10.5.1 Message Protocol with MAC Enabled 144

10.6 Extended Transaction Code Table 145

10.7 Triple-DES Encryption of the PIN Block 146

10.8 Encryption Key Sets 148

10.9 ANSI 16-bit CRC 150
10.9.1 ANSI 16-bit CRC 151

10.10 EMV 153
10.10.1 EMV-1 153
10.10.2 EMV-2 154
10.10.3 EMV-3 158
10.10.4 EMV-4 159
10.10.5 EMV-5 159
10.10.6 EMV-6 161

10.11 Top Up 162
10.11.1 Top Up miscellaneous FIDs to use for certain version of terminal software 162

11 REMOTE KEY TRANSPORT EXTENSIONS 164

11.1 Introduction 164

11.2 Message Format Summary 164
11.2.1 New FIDs 164
11.2.2 New Timeout 164

11.3 RKT Message Flows 165
11.3.1 Initiate RKT Session from Host 165
11.3.2 Certificate Exchange Flow 166
11.3.3 Re-keying Flow 169
11.3.4 Error with RKT Data Packets 171

12 TRITON TERMINAL CURRENCY EXCHANGE SPECIFICATION 173

12.1 Introduction 173

12.2 Message Format Summary 174
12.2.1 Transaction Codes 174

PRELIMINARY

12.2.2 Authorization Codes 174
12.2.3 Miscellaneous FIDs 174

12.3 Currency Exchange FID Overview 175

13 TRITON TERMINAL PROSA FEE EXTENSIONS 178

13.1 Introduction 178

13.2 Message Format Summary 178
13.2.1 Miscellaneous FIDs 178

13.3 Prosa Fee FID Overview 179
13.3.1 Transaction Example 180

14 TRITON TERMINAL TCP/IP PROTOCOL SPECIFICATION 181

14.1 Introduction 181

14.2 Configuration 181
14.2.1 Software Configuration Parameters 181
14.2.2 Hardware Configuration 97XX 181
14.2.3 Hardware Configuration 96XX 181
14.2.4 Hardware Configuration 98XX 182

14.3 Message Formats 183

14.4 Protocol Definition 184
14.4.1 Response Message Timeout 184
14.4.2 Normal Request Message and Normal Response Message 184
14.4.3 Error Handling 184

14.5 CRC-16 (ANSI) Calculation 186
14.5.1 Including the CRC into a message 186
14.5.2 Validating the CRC on a message 186
14.5.3 C code example of a CRC-16 (ANSI) calculation. 187

15 MAC CALCULATION 188

15.1 Constructing a message with a MAC 188

15.2 Validating a message with a MAC 188

15.3 MAC field calculation 188

16 RADIO PAD PROTOCOL 189

PRELIMINARY

16.1 Control Mode 189

16.2 Data Mode 189

16.3 Application Layer Protocol 190

PRELIMINARY

List of Tables

TABLE 1 TRANSACTION OPTIONS PDU 24
TABLE 2 TRANSACTION OPTIONS LABEL VALUES 24
TABLE 3 OAR DATA PDU 25
TABLE 4 OAR RETURN CODES 25
TABLE 5 VALID OAR BUTTON LABELS 25
TABLE 6 CUSTOM SCREEN TEXT PDU 26
TABLE 7 – HOST WITHDRAWAL SURCHARGE AMOUNT 26
TABLE 8 – HOST BALANCE SURCHARGE AMOUNT 26
TABLE 9 – FX-DCC ELIGIBILITY REQUEST 27
TABLE 10 FX-DCC ELIGIBILITY RESPONSE 27
TABLE 11 - FX-DCC LOOKUP REQUEST 28
TABLE 12 - FX-DCC LOOKUP RESPONSE 28
TABLE 13 FX-DCC TRANSACTION REQUEST 29
TABLE 14 FX-DCC TRANSACTION RESPONSE 30
TABLE 15 FX-DCC REVERSAL REQUEST 30
TABLE 16 FX-DCC REVERSAL RESPONSE 31
TABLE 17 FX-DCC CLOSE REQUEST 31
TABLE 18 FX-DCC CLOSE RESPONSE 32
TABLE 19 – FX-DCC FAST CASH AMOUNTS LIST 32
TABLE 20 FX-DCC SURCHARGE TYPE 33
TABLE 21 FX-DCC SURCHARGE TYPE ENUMERATIONS 33
TABLE 22 – TERMINAL CASSETTE CONFIGURATION 33
TABLE 23 – VALID DISPENSER MECHANISM IDS 33
TABLE 24 CASSETTE CONFIGURATION INFORMATION 34
TABLE 25 – VALID CASSETTE STATE VALUES 34
TABLE 26 FX-DCC OFFER 35
TABLE 27 FX-DCC CONVERTED SURCHARGE AMOUNT 35
TABLE 28 FX-DCC CONVERTED AMOUNT 35
TABLE 29 FX-DCC DISCLAIMER SCHEME 36
TABLE 30 VAILD FX-DCC DISCLAIMER SCHEMES 36
TABLE 31 FX-DCC OFFER ACCEPTANCE FLAG 36
TABLE 32 FX-DCC PRESENTED AMOUNT 36
TABLE 33 FX-DCC CLOSE TYPE 36
TABLE 34 FX-DCC TRANSACTION COUNT 37
TABLE 35 FX-DCC SETTLEMENT AMOUNT 37
TABLE 36 FX-DCC CLOSE RESET FLAG 37
TABLE 37 FX-DCC REQUESTED AMOUNT 38
TABLE 38 FX-DCC DISPENSED AMOUNT 38
TABLE 39 FX-DCC MARKUP 38
TABLE 40 TDL TRANSACTION IDENTIFIERS 39
TABLE 41 TDL ACCOUNT IDENTIFIERS 39
TABLE 42 SURCHARGE AMOUNT 40
TABLE 43 REVERSAL STATUSES 42
TABLE 44 REVERSAL REASON ENUMERATIONS 42
TABLE 45 EXTENDED TRANSACTION RECORD TLVS 44
TABLE 46 E-RECEIPT MESSAGE TYPES 47
TABLE 47 WORKING KEY 47

PRELIMINARY

TABLE 48 WORKING KEY ID....48
TABLE 49 LANGUAGES....48
TABLE 50 GIFT CARD OPERATION FAILURE INFORMATION....50
TABLE 51 LIST OF GIFT CARD SERIAL NUMBERS....50
TABLE 52 GIFT CARD FAILURE CODES....50
TABLE 53 GIFT CARD ACTIVATION....51
TABLE 54 ACTIVATION STATUS....51
TABLE 55 GIFT CARD ACTIVATION LIST....51
TABLE 56 GIFT CARD ACTIVATION ENTRY....52
TABLE 57 GIFT CARD DEACTIVATION....52
TABLE 58 GIFT CARD DEACTIVATION LIST....52
TABLE 59 GIFT CARD DEACTIVATION ENTRY....52
TABLE 60 GIFT CARD PM STATISTICS....53
TABLE 61 SESSION PDU....53
TABLE 62 SESSION CREDENTIALS....53
TABLE 63 GET-DATA REQUEST....54
TABLE 64 CORRELATION INFORMATION....54
TABLE 65 CARDLESS TRANSACTION REQUEST....54
TABLE 66 SESSION PDU....55
TABLE 67 GET-DATA RESPONSE....55
TABLE 68 CARDLESS TRANSACTION RESPONSE....55
TABLE 69 TRANSACTION RESPONSE CODES....55
TABLE 70 MESSAGE ERROR....56
TABLE 71 SESSION PDU....57
TABLE 72 GET-DATA REQUEST....57
TABLE 73 CARDLESS SESSION KEY REQUEST....57
TABLE 74 TERMINAL MESSAGE ITEM....57
TABLE 75 TERMINAL MESSAGE TYPE....57
TABLE 76 SESSION PDU....58
TABLE 77 GET-DATA RESPONSE....59
TABLE 78 CARDLESS SESSION KEY RESPONSE....59
TABLE 79: SESSION PDU....60
TABLE 80: SESSION CREDENTIALS....60
TABLE 81: DIGITAL CURRENCY PURCHASE REQUEST ENCODED AS SET-DATA REQUEST....60
TABLE 82: CORRELATION INFORMATION....60
TABLE 83: SESSION PDU....60
TABLE 84: SESSION PDU....62
TABLE 85: DIGITAL CURRENCY PURCHASE RESPONSE ENCODED AS SET-DATA RESPONSE....62
TABLE 86: SET-DATA RESPONSE CODE ENUMERATION....62
TABLE 87: SESSION PDU....64
TABLE 88: DIGITAL CURRENCY COMMITMENT REQUEST ENCODED AS SET-DATA REQUEST....64
TABLE 89: AUTHORIZATION CODE....64
TABLE 90: SESSION PDU....65
TABLE 91: DIGITAL CURRENCY COMMITMENT RESPONSE ENCODED AS SET-DATA RESPONSE....65
TABLE 92 - ALLOCATED TAG ENUMERATIONS....71
TABLE 93 – CONTROL CHARACTER TABLE....83
TABLE 94 – TRANSACTION REQUEST MESSAGE....85
TABLE 95 - TRANSACTION REQUEST MESSAGE -2....87
TABLE 96 – CONFIGURATION TABLE DOWNLOAD REQUEST MESSAGE....89
TABLE 97 - CONFIGURATION TABLE DOWNLOAD REQUEST MESSAGE....90

PRELIMINARY

TABLE 98 – HOST TOTALS DOWNLOAD REQUEST MESSAGE....93
TABLE 99 - HOST TOTALS DOWNLOAD REQUEST MESSAGE - 2....94
TABLE 100 – REVERSAL REQUEST MESSAGE....96
TABLE 101 - REVERSAL RESPONSE MESSAGE....97
TABLE 102 - RKT REQUEST MESSAGE....98
TABLE 103 - RKT RESPONSE MESSAGE....99
TABLE 104 - TRANSACTION INFORMATION REQUEST MESSAGE....102
TABLE 105 - TRANSACTION INFORMATION RESPONSE....103
TABLE 106 – STATUS MONITORING FIELD....105
TABLE 107 – PRODUCT CODE TABLE FOR PROGRAM/TABLE VERSION NUMBERS....106
TABLE 108 - PRODUCT CODE TABLE FOR PROGRAM/TABLE VERSION NUMBERS....106
TABLE 109 – TRANSACTION CODE TABLE....108
TABLE 110 – AUTHORIZATION CODES TABLES....111
TABLE 111 – MISCELLANEOUS FIELD ID DESCRIPTIONS....121
TABLE 112 – VALID MISCELLANEOUS FIELD ID TABLES....124
TABLE 113 – EXTENDED TRANSACTIONS CODE TABLE....145
TABLE 114 E1 – SERVICE PARAMETERS.....148
TABLE 115 EMV-1.1.1....153
TABLE 116 EMV-1.1.2....154
TABLE 117 EMV-2.1.1....156
TABLE 118 EMV-2.1.2....156
TABLE 119 EMV-2.2.1....157
TABLE 120EMV-2.2.2....157
TABLE 121 EMV-3....158
TABLE 122 EMV-4....159
TABLE 123 EMV-5....159
TABLE 124 EMV-6....161
TABLE 125 – TOP UP MISCELLANEOUS FIDS TO USE FOR CERTAIN VERSION OF TERMINAL SOFTWARE....163
TABLE 126 – NEW FIDS....164
TABLE 127 – CURRENCY EXCHANGE FID OVERVIEW....176
TABLE 128 – PROSA FEE FID OVERVIEW....179
TABLE 129 – ERROR HANDLING....185

PRELIMINARY

1 TDL Specification Organization

The TDL specification is divided into the following parts:

Part	Description
1	TDL and TDL Legacy Specification (legacy transaction compatibility)
2	Session and Terminal Management
3	Linq3 Lotto Specification

2 Revision History

Version	Date	Description
0.1	March 8, 2010	Preliminary
0.2	April 19, 2010	Section 10.1– Modified language regarding amount 2 to state that it will be ignored in host responses. Sections 7.3.4– New section defining a TLV item for host provided withdrawal surcharge. Sections 7.3.5 – New section defining a TLV item for host provided balance surcharge. Section 0 – New section for listing all tag enumerations assigned by Triton Systems LLC in one location.
0.3	June 2, 2010	Added section 9 (TDL Examples)
0.4	August 5, 2010	Added tag IDs for FX-DCC in Section , fixed invalid TLV examples. Added sections describing FX-DCC TLVs.
0.5	August 17, 2010	Added section 10.1.3 to clarify usage of ASCII tag and length fields in TLV encodings for 7-bit ASCII communications.
0.6	August 20, 2010	Added section 7.3.7 for TDL Transaction Data
0.7	September 14, 2010	Added sections for GET-DATA, SET-DATA, ACTION requests and response PDUs and the EVENT notification PDU. Also added text to the table captions missing text.
0.8	November 3, 2010	Corrected format errors, made post review edits.
0.9	November 18, 2010	Modified Table 111 – Miscellaneous Field ID Descriptions to update the specification for the TS Legacy 'xq' FID.
0.10	December 3, 2010	Added section to define ASCII represented data Added section to define a generic message error Modified VersaSafe sections to use ASCII represented data instead of binary value types.
0.11	December 17, 2010	Added layered communications model diagram, section 6.1 Added TDL packet format section 7.1.1
0.12	December 27, 2010	Added Linq3 communication definition. See [3]
0.13	February 18, 2011	Added session treatment in [2], minor edits
0.14	April 5, 2011	Added section now described in [2] and moved several related sections around for better organization
0.15	June 8, 2011	Added more terminal management support (too much to describe in detail) and some document organization changes.
0.15	July 18, 2011	Added TDL examples for cassette document status change events, now described in [2]

PRELIMINARY

0.16	August 23, 2011	Updated VersaSafe Extensions
0.17	September 26, 2011	Added correlation information tag to GET-DATA, SET-DATA Request and Response PDU.
0.18	October 10, 2011	Removed tag 5337 "Status Information Group" as extraneous. It would only have been required if reports of the same type (add, remove, replace, report and reset) could include status more than 1 cassette in a single event.
0.19	October 27, 2011	Added optional 2-byte length at beginning of TDL message PDUs bound for the transport layer.
0.20	December 6, 2011	Added gift card services sections and examples. Also made changes to the session message fields to support an optional endpoint ID field.
0.21	December 15, 2011	Added information describing session credentials tag 5004, now found in [2]. This section makes CBC MAC a generic method of authentication. The gift card section 7.3.8.1 for gift card credentials will now refer back to this new section.
0.22	February 9, 2012	Modified TDL examples for cassette document management and gift card activation to reflect the correct format of encoded length octets.
0.23	June 18, 2012	Various cut and paste corrections.
0.24	June 21, 2012	Reorganized the FX-DCC section to make it more configurable per implementation needs.
0.25	June 26, 2012	Added FX-DCC disclaimer scheme TLV. Added FX-DCC transaction request/response tags and associated sub tags. Add FX-DCC PIN working key request/response data including modifications to the working key definition in section 7.3.7.38. Added DCC Close Request/Response framework.
0.26	July 18, 2012	Added FX-DCC Reversal section.
0.27	August 1, 2012	Added new tags under TDL Transaction Data. These new tags are associated with the new ATM Transaction Record tag.
0.28	August 13, 2012	Added new tags in gist card services section for reporting simple performance monitoring statistics (section 7.3.8.5)
0.29	August 27, 2012	Added new tags Disable SMS Command, Enable SMS Message, Enable SMS Response, Enable SMS, and Language Selected
0.30	September 27, 2012	Added new tags Business date and Authorization number to FX DCC transaction message. Updated FX DCC transaction, reversals, offer to latest needs.
0.31	October 11, 2012	Added TDL legacy FID 'zo' providing an extended conformation ID for Money Send/Receive VAS transactions
0.32	December 4, 2012	Re-organize the TDL specification into parts beginning with this part (part 1), Session and terminal management (part 2), and Lotto (part 3)
0.33	January 29, 2013	Added new tags EMV Data and Triton Specific EMV Data. Updated FX-DCC transaction and reversals to include EMV information.
0.34	February 1, 2013	Corrected reference to FX-DCC Requested Amount in reversals. It now uses tag ID 1003 instead of 1043, tag ID 1043 has been removed.
0.35	February 12, 2013	Added host surcharge to FX-DCC lookup response. Also added Host supplied surcharge type to FX-DCC Surcharge Type enumeration.
0.36	February 25, 2013	Added ATM Report Record
0.37	May 29, 2013	Added 'qc' FID definition to TDL Legacy extension section 10.1
0.38	June 25, 2013	Added section 16 describing RadioPAD protocol implementation
0.39	July 18, 2013	Clarified '085' authorization code information in 10.2.14 and 10.10.1.
0.40	November 1, 2013	Corrected DCC reversal amount information in 7.3.6.5
0.41	November 21, 2013	Added ACK to message completion for RadioPAD in protocol diagram
0.42	January 13, 2014	Added RKT request/response and TIR request/response message format sections

PRELIMINARY

0.43	August 19, 2014	Added tag ID 44 for simple error code option in SET-DATA Response
0.44	September 17, 2014	Added a new EI (Encoding Identifier) to support encryption of the TDL payload in section 7.1.1. Modified 'w' FID in section 10.2.19 to indicate 3 or 4 - character field.
0.45	February 18, 2014	Added 'converted' tags to FX-DCC transaction request section
0.46	November 17, 2015	Corrected table references for account identifier tags in section 7.3.7.3 and removed VersaSafe definition as it no resides in TDL Part 4.
0.47	March 17, 2016	Added constructed tags for transaction request/response for standard financial transactions (tag IDs 146 and 147) such as withdrawal and balance inquiry. Tags are referenced in section 7.3.7.
0.48	March 31, 2016	Added Cardless transaction tag enumerations to table
0.49	March 17, 2017	Added Balance Inquiry Surcharge, IVA Tax and Exchange Fee, and bank fees to the Allocated Tag Enumerations Table in Section 8. Added sections for deposit transaction tags
0.50	November 9, 2017	Corrected section 16.3 dealing with Radio PAD protocol
0.51	March 7, 2017	Repaired broken reference in section 10.4 Cleaned up related document references
0.52	July 9, 2018	Updated section 16.2 to define the maximum buffer size for Radio PAD protocol Repaired flawed TDL example in section 9.5.1 for gift card services. The session message ID indicated 8 (response) when it should indicate 7 (request) Added cardless transaction tags Added digital currency purchase tags Added value 3 to note status in status monitoring FID with the following note. "The note status may not be determined in some error cases, this will be indicated by value 3."
0.53	March 1, 2021	Modified tag enumeration table by moving tag ID 170 for TR-31 key block to 171. Added tag ID 170 as Card-less session key password to match implementation, added tag 172 to support a flag indicating TR-31 key block processing support. Also added sections 10.2.4.1 and 10.2.5.1 to describe the TR-31 key block options in the configuration table download request/response messages. Updated section 10.2.5 to indicate the '~' FID is not required for TR-31.
0.54	February 16, 2022	Section 10.6: Added Extended Transaction Code type Crypto Purchase

Table 1 - Revision History

PRELIMINARY

3 Introduction

Triton Dynamic Language™ (TDL™) is a communications language used to provide an interface between an unattended Terminal and the Host processing system.

This specification is a draft specification and as such contains technical information currently in development. Sections of this document are subject to change during the development process.

4 Definitions

Address	32-bit number uniquely identifying a TCP/IP server, client, or gateway.
ANSI	American National Standards Institute.
TLV	Tag Length Value
CA	Certificate Authority: SSL certificates must be signed by a trusted source.
CRC	(Cyclic redundancy checking) a method of checking for errors in data that has been transmitted on a communications link
FID	Field Identification. This is a unique character or characters identifying each different type of data sent by the Terminal or Host.
Field	A portion of the data sent by the Terminal or Host that can be grouped by function.
FS	1C Hex (Field Separator).
Gateway	A central point through which connections and translations are performed to connect different and sometimes incompatible networks.
Host	Refers to the processing end of the communications.
IP	Internet Protocol: responsible for moving packet of data from node to node. IP forwards each packet based on a four byte (32 bit) destination address (the IP number).
MAC	Message Authentication Code: Data used to authenticate the message received has come from a known sender
OAR	Open Account Relationship
PDU	**P**rotocol **D**ata **U**nit. A unit of information at layer N that includes all data added to the beginning and/or ending of an SDU from layer N + 1.
Port	16-bit number identifying a specific application or service on a given TCP/IP server or client.
PPGC	**P**re**P**aid **G**ift **C**ard Services
PPP	Point-to-point protocol.
RKT	Remote Key Transport
SDU	**S**ervice **D**ata **U**nit. A unit of information at layer N transformed to a PDU by layer N -1.
Sockets	is a name given to the package of subroutines that provide access to TCP/IP.

PRELIMINARY

SSL	Secure Socket Layer: Method of encrypting data using a secure connection to the Host.
Subnet Mask	This is a 32-bit value that allows the recipients of IP packets to distinguish the network ID and the host ID of the IP address.
Subnetting	Subdividing a large network into smaller networks.
TCP	Transmission Control Protocol: responsible for verifying the correct delivery of data from client to server. Data can be lost in the intermediate network. TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received.
Terminal	Refers to the ATM or other unattended device.
Terminal Parameters	Selected configuration data from the ATM's non volatile memory storage.

5 Related Documents

[1] TSJR Triton Terminal Journal Record Format Specification
[2] TDL Part 2 – Session and Terminal Management
[3] TDL Part 3 – Linq3 Lotto Definition

6 System Definition

6.1 Layered Communication Model

Triton applications communicate with remote hosts over a variety of communications media. The most common among them being TCP/IP over Ethernet, modem protocol, and a version of GSM wireless Packet Assembler/Disassembler (PAD), also shortened to RadioPAD. Packetized information at the application layer is either TDL or TDL Legacy or data as specified in this document. Application processes using dialup modem communications passing through a Visa II Protocol layer for the purpose of application connection establishment and acknowledgement of exchanged packets. The application may optionally use the Visa II protocol over the other media as well. The following diagram illustrates the model:

PRELIMINARY

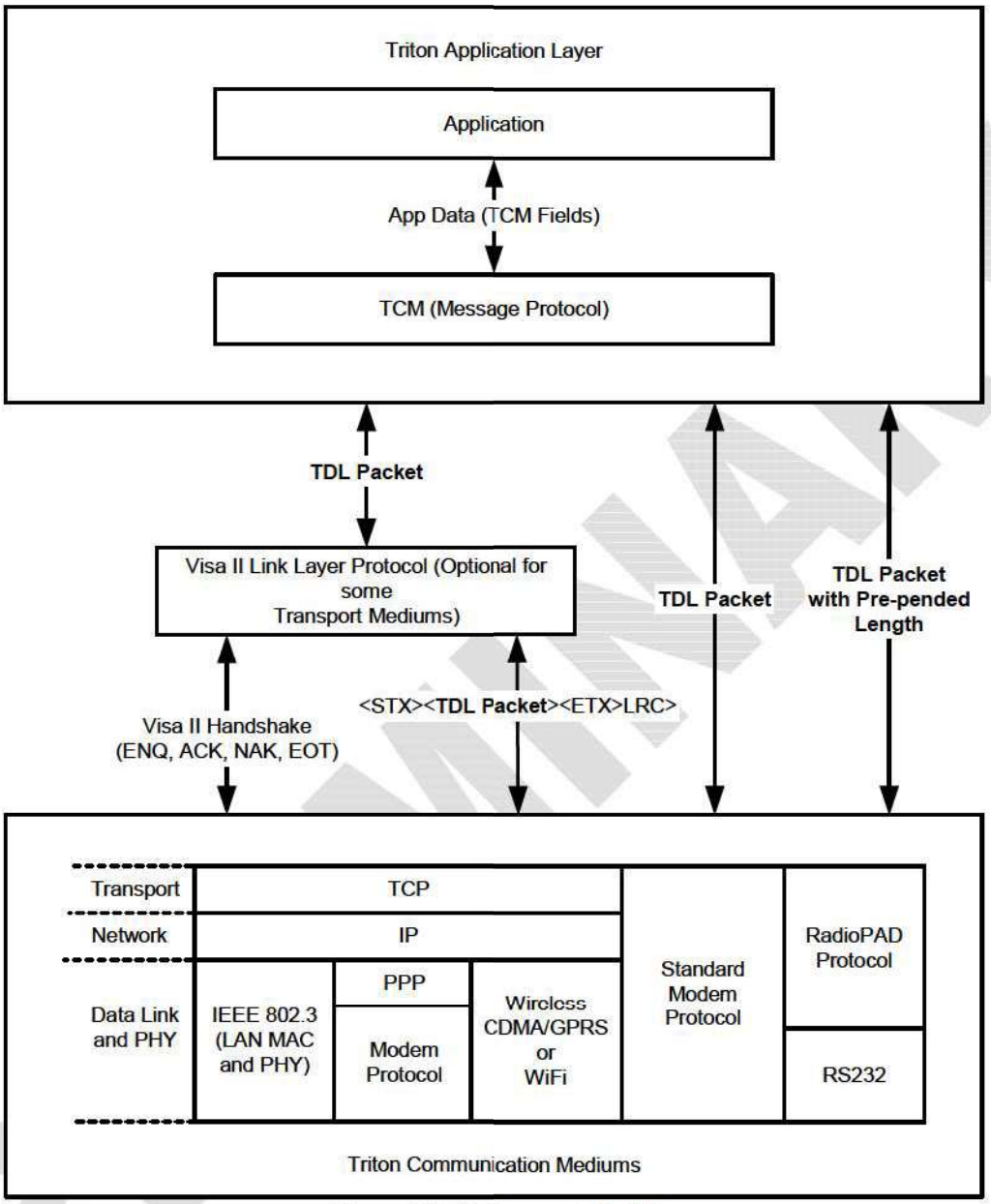


Figure 1 TDL Layered Model

6.1.1 TCP/IP Protocol

TDL™ is can be used across a TCP/IP network. A device using TDL™ can be connected to a network using a number of methods:

- Direct Ethernet connection
- Wireless modem to a CDMA/GPRS network
- Wifi modem to a wireless router

- Dialup modem to a PPP (Point-to-Point Protocol) host

TCP/IP communications enable the Terminal and the Host to transmit multiple request/response messages in a single transaction on the Terminal. When possible this is the preferred means of exchanging messages between the terminal and Host.

The Terminal can be configured to either connect to the Host at the time the Terminal goes in service to perform customer transactions (permanent connection), or perform a connection for each message request. Note that if using a dialup network with a PPP connection it is recommended to use permanent connection if multiple request/response messages are intended to be sent in a transaction.

6.1.1.1 Optional TDL SDU Format

For backward compatibility, the terminal will optionally support sending application data to the transport layer with a 2-byte length prepended to the TDL message.

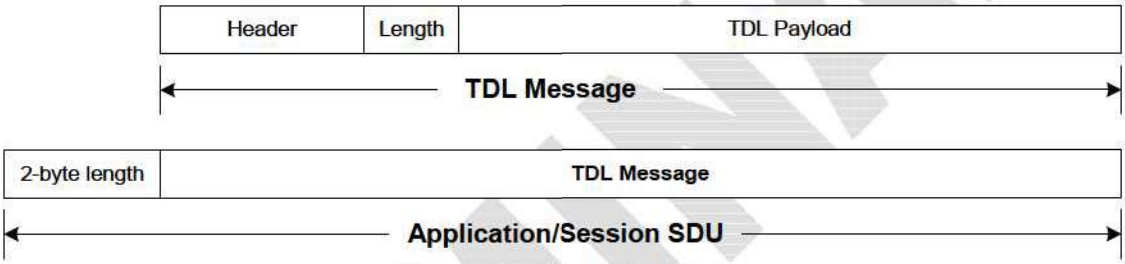


Figure 2 Optional TDL SDU

This essentially constitutes an optional application or session layer SDU at the transport (TCP) boundary. The TDL message is still structured as described in section 7.1.1, but prepends a 2-byte binary length in big ENDIAN format to the application data as it is sent through the TCP connection.

6.1.2 Visa II Protocol

TDL™ can also be used within Visa II protocol as described in section 10.3. This protocol includes wrapping the messages with STX/ETX/LRC wrappers for the purpose of transmitting data for legacy systems (for example, dialup or serial Radiopad).

It is not recommended to use this protocol if multiple request/response messages are intended to be sent in a transaction.

6.2 Security

6.2.1 Message Encryption

Secure encrypted communications using TDL™ can be implemented using SSL over the TCP/IP communication. SSL requires the device to authenticate the host's certificate by verifying the certificate against known Certificate Authorities stored on the device.

The following elements are authenticated by the Terminal at the time an SSL connection is established:

- The CA (Certificate Authority)
- The expiration date of the certificate
- The common name of the certificate contains the domain name for the SSL server

PRELIMINARY

6.2.2 Message Authentication

Message authentication can be implemented using a MAC field as defined in section 10.5. MACing is performed according to ANSI Standard X9.19 and ISO/IEC 9797-1.

PRELIMINARY

7 Message Definitions

This section defines all messages supported by this version of TDL™.

7.1 TDL™ Packet Data Format

The basic unit of the TDL™ payload is a Triton Proprietary Tag/Length/Value (TLV). It is loosely based on the transfer syntax one might expect from an ASN.1 DER-capable engine in terms of how the tag and length octets are generated with the following exceptions:

1. There is no conformance to ASN.1 DER implied by this document.
2. Triton transfer syntax defines all TLV tags as PRIVATE class.
3. Triton private class tags are not necessarily encoded as single bytes for identifiers of value 30 or less.
4. Only the definite short and long form of length octets is used.

7.1.1 TDL Packet Structure

The structure of the payload is as follows:

8 Bytes	**4 Bytes**	**N Bytes**
Header	Length	Packed and Encoded Payload Data

1. Header – The header is 8ASCII printable characters with the form: "TVVVPPEE". The first character is always "T" and identifies the message protocol as TDL, the remainder of the bytes are defined as follows:
    a. VVV – This is a 3 byte version number consisting of 3 ASCII numeric characters and an implied decimal point between the 1nd and 2nd character. The first character specifies the major version number and the remainder form the minor version number.
    b. PP – is a data Packing Identifier (PI) consisting of 2 ASCII alphanumeric characters with the following possible values.
        i. "00" – indicates bytes are encoded and packed in raw binary
        ii. "01" – indicates bytes are encoded and packed using Base64
    c. EE – is an Encoding Identifier (EI) consisting of 2 ASCII alphanumeric characters with the following possible values.
        i. "00" – indicates Triton Encoding Rules level 1
        ii. "01" - indicates encrypted Triton Encoding Rules level 1. The encryption mechanism is outside the scope of this document and left to the sending/receiving contexts but implies that once "unpacked according to the PP bits, the payload data must be decrypted by a mutually agreed upon encryption scheme prior to implementing the Triton encoding rules.
2. Length – 4 ASCII HEX character string indicating the length of the packed and encoded TDL payload data with zero "0" padding on the left as needed. The string represents a 2-byte HEX unsigned integer in BIG endian order. For example "010A" would translate to $10A_{16}$ (266_{10}).
3. Packed and Encoded Payload Data – This is the payload data encoded by the means indicated in the header before transmission by the sender. The receiver first unpacks the data in the manner specified by the Packing Identifier, and then decodes the data in the manner specified by the Encoding Identifier.

PRELIMINARY

7.2 TDL Data Definitions

This section defines many basic and specialized features in terms of their TDL TLV data items.

7.2.1 Basic ASCII Represented Values

Most primitive types may be represented in ASCII for readability and platform independence. TDL elements which use this representation should specify so in the definition. The following table outlines the types and ASCII representation of those types.

Data Type	Description
String	Basic ASCII string data
Boolean	A boolean expressed as a single ASCII decimal digit. The digit may be a '0' for false or a '1' for true.
Integer	A 32 bit integer value expressed as a group of ASCII values consisting of the decimal digits '0' (0x30) to '9' (0x39) inclusive and a minus sign '-' (0x2D) if negative. Commas are not allowed. Valid range is between '-2147483648' and '2147483647'.
Unsigned Integer	A 32 bit unsigned integer value expressed as a group of ASCII values consisting of the decimal digits '0' (0x30) to '9' (0x39) inclusive. Commas are not allowed. Valid range is between '0' and '4294967295'.
Hexadecimal Unsigned Integer	A 32 bit unsigned integer value in hexadecimal form expressed as a group of ASCII values consisting of the decimal digits '0' (0x30) to '9' (0x39) and characters 'A' (0x41) to 'F' (0x46) inclusive. The value can contain up to 8 characters (4 ASCII represented bytes). Valid range is between '00' and 'FFFFFFFF'.
Double[1]	A floating point value expressed as a group of ASCII values consisting of the decimal digits '0' (0x30) to '9' (0x39) inclusive, one '.' (0x2E) character as the decimal point, and a minus sign '-' (0x2D) if negative. There must be at least 1 ASCII decimal digit on either side of the decimal point.
Date-Time	A date-time value will be expressed as a group of ASCII values representing the date and time. The format of the date-time string will be 'YYYY-MM-DD HH:MM:SS', where YYYY is the year, MM is the month, DD is the day, HH is the hour (24 hour clock), MM is the minute, and SS is the second. The full 4 digit year must be used. The month, day, hour, minute, and second must be left padded with a '0' if single digit.
Date	A date value will be expressed as a group of ASCII values representing the date only. The format of the date string will be 'YYYY-MM-DD', where YYYY is the year, MM is the month, and DD is the day. The full 4 digit year must be used. The month and day must be left padded with a '0' if single digit.
Time	A time value will be expressed as a group of ASCII values representing the time only. The format of the time string will be 'HH:MM:SS', where HH is the hour (24 hour clock), MM is the minute, and SS is the second. The hour, minute, and second must be left padded with a '0' if single digit.

7.2.2 Binary Data

Some data types cannot be represented as ASCII values, for instance file blocks and journal record data. The Binary data type is defined for this purpose. Binary data values are sent as a variable length block of bytes, each byte assuming the value between 0x00 and 0xFF. Communication mediums sensitive to binary data, such as 7-bit ASCII modem protocols should use base64 packing for the TDL payload in this case.

[1] Floating point values are subject to the rounding conventions of standard 'C' language text to floating point conversion library routines. Customers can expect values sent by the host to be rounded to 6 decimal places

Highly Confidential: Property of Triton Systems – Do Not Disclose

7.2.3 Message Error

Tag: FF1C
Length: Variable
Value: The Message Error TLV is used to inform the client end that there was some type of error with the request message. Errors can include but not limited to: checksum errors, decode errors, unknown request types, missing mandatory elements, etc.

Tag	Value	
FF1C	Message Error (tag ID 28)	
	Tag	**Value**
	DF1D	Error Code (tag ID 29)
	DF1E	Error Description (tag ID 30)

7.2.3.1 Error Code

The Error Code TLV contains the error code. The value type is an ASCII represented integer.

Tag	Value
DF1D	Error Code (tag ID 29)

7.2.3.2 Error Description

The Error Description TLV contains a description of the associated error code. The data type is a variable length ASCII string.

Tag	Value
DF1E	Error Description (tag ID 30)

7.3 TDL Packet Process at Host

This diagram below illustrates in brief the TDL packet process for a host processing a terminal TDL message request:

PRELIMINARY

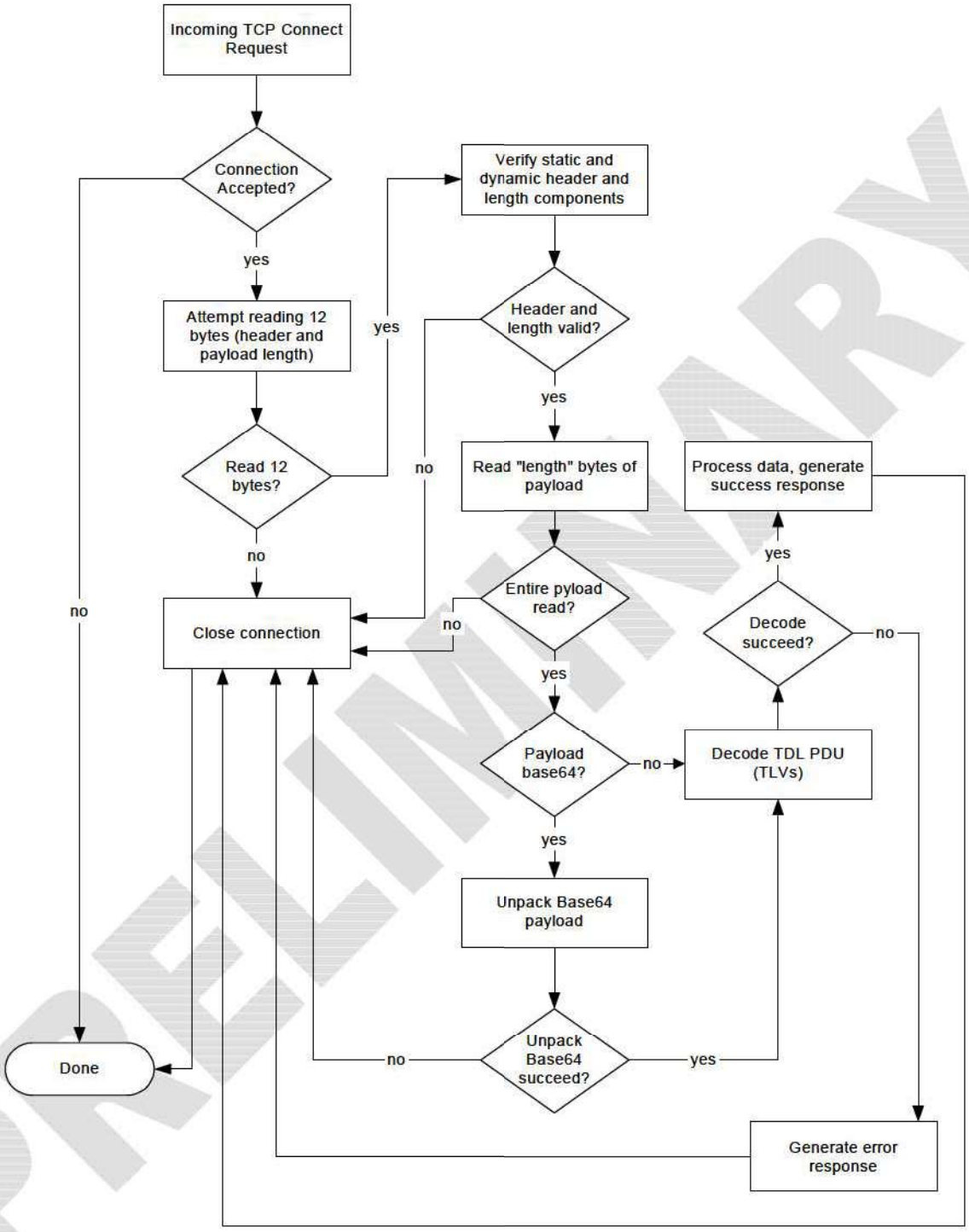


Figure 3 Host TDL Message Process

Implementers should note that when a terminal is setup to do so, there is a message length field prepended in front of the TDL message header. See section 6.1.1.1 for details the optional format.

Highly Confidential: Property of Triton Systems – Do Not Disclose

7.3.1 Transaction Options

Tag: FF01
Length: Variable
Value: See the following table for format of the data in this item:

Tag	Value	
FF01	Transaction Option Data (tag ID 1)	
	Tag	**Value**
	DF02	Label string (tag ID 2)
	DF03	Boolean State Flag (0 = Disable, 1 = Enable), (tag ID 3)

Table 1 Transaction Options PDU

Example: "FF01**19**DF02**0C**Option LabelDF03**01**0" (Disables specified option)

Notes: Terminal will send this TLV item for each option that can be enabled or disabled by the host. Terminal will indicate the current enable/disable status. Host should send desired setting or not sending this TLV item indicates to use default value.

Label values must be synchronized between the Terminal and the Host and can be any ASCII string. In the example above, the label is "Option Label" and the state flag is "Disable".

Valid Label Values:

Label	Description
Bansi	If enabled, this flag indicates the customer is a member of Bansi bank.
Prepaid Phone	Indicates if current transaction should allow Prepaid Phone option.
Overdraft Fee	Indicates for the terminal to display an overdraft fee message to the user (Amount1 in TIR will indicate requested withdrawal amount).

Table 2 Transaction Options Label Values

PRELIMINARY

7.3.2 OAR Data

Tag: FF04
Length: Variable
Value: See the following table for format of the data in this item:

Tag	Value	
FF04	OAR Data (tag ID 4)	
	Tag	**Value**
	DF05	Return Code (Required), (tag ID 5)
	DF06	OAR Screen Text (Host), (tag ID 6)
	DF07	OAR Button Label (Host), (tag ID 7)
	DF08	Customer Selected Label (Terminal), (tag ID 8)

Table 3 OAR Data PDU

Notes: Terminal will send this TLV item indicating the host can send an OAR screen or response code 2 indicating end of OAR processing. Terminal will continue sending OAR data until response code 2 received.

Valid Return Codes:

Code	Description
00	Sent from terminal for initial OAR request.
01	Sent from host to indicate to display specified OAR screen (will include DF06 and DF07 tags).
02	Sent from host to indicate no OAR screen.
03	Sent from terminal to indicate customer selection (will include DF08 tag).

Table 4 OAR Return Codes

Valid Button Labels:

Label	Description
0-9	Numeric keypad input
OK	OK key
Cancel	Cancel key
Clear	Clear/Change key
Ctrl	Ctrl key
#xxxxxx	Custom label for screen keys (xxxxxx can be any text string)

Table 5 Valid OAR Button Labels

PRELIMINARY

7.3.3 Custom Screen Text

Tag: FF09
Length: Variable
Value: Label and string for a text string that can be customized by the host for a transaction. The label will be a unique string for the ATM. See the following table for format of the data in this item:

Tag	Value	
FF09	Custom Screen Text Data (tag ID 9)	
	Tag	**Value**
	DF02	Label string (tag ID 2)
	DF0A	Text string (tag ID 10)
	DF03	Boolean State Flag (0 = optional, 1 = required), (tag ID 3)

Table 6 Custom Screen Text PDU

Example: "FF09**2D**DF02**05**Label DF0A**15**Text string for label DF03**01**1"

Notes: ATM will send this TLV item in a TIR for each text field that can be customized by host. ATM can populate text with default string defined at the ATM or a null string. Host should send custom text (or null string to indicate use default). If the Terminal sends a "1" in the required tag, the host must send a value back to the Terminal or the transaction will be cancelled.

7.3.4 Host Withdrawal Surcharge Amount

Tag: DF0B
Length: Variable
Value: See the following table for format of the data in this item:

Tag	Value
DF0B	This value represents a withdrawal transaction surcharge amount sent by the host to the terminal (tag ID 11). The value type is an ASCII represented double.

Table 7 – Host Withdrawal Surcharge Amount

Example: "DF0B**04**1.50"

The amount 2 field of the TIR response will be ignored. All surcharge information associated with a withdrawal transaction that issues TIR requests must send this host withdrawal surcharge amount TLV in the TDL payload.

7.3.5 Host Balance Surcharge Amount

Tag: DF0C
Length: Variable
Value: See the following table for format of the data in this item:

Tag	Value
DF0C	This value represents a balance transaction surcharge amount sent by the host to the terminal (tag ID 12). The value type is an ASCII represented double.

Table 8 – Host Balance Surcharge Amount

Example: "DF0C**04**1.50"

The amount 2 field of the TIR response will be ignored. All surcharge information associated with a balance transaction that issues TIR requests must send this host balance surcharge amount TLV in the TDL payload.

7.3.6 FX-DCC

Since FX-DCC can be implemented in a variety of ways based on customer needs, this section is designed with flexibility in mind. Implementers use combinations of a static component such as the FX-DCC Eligibility Request or Response tag from section 7.3.6.1 and selected FX-DCC supplemental tags from section 7.3.6.7 to customize eligibility discovery for their needs and so on. It is recommended that implementers choose session conventions described in [2] to wrap FX-DCC messages used in exchanges for the purpose of routing PDUs to the correct application in multi-process computing environments.

7.3.6.1 FX-DCC Eligibility Determination

There are a variety of ways to determine FX-DCC eligibility through TDL. This section describes the various ways to determine a customer's eligibility for an FX or DCC transaction through the exchange of TDL PDUs. Eligibility may be determined through a separate request as shown in 7.3.6.1.1 or may be combined with the offer lookup, as in 7.3.6.2.1.

7.3.6.1.1 FX-DCC Eligibility Request

Tag: FF8802
Length: Variable
Value: FX-DCC Eligibility Request

Tag	Value
FF8802	FX-DCC Eligibility Request (tag ID 1026)

Table 9 – FX-DCC Eligibility Request

One or more of the following sub-TLVs must be included in the FX-DCC Eligibility Request based on the customer's needs.

1. Cassette Configuration – Tag ID 1000 (see 7.3.6.7.6)
2. PAN – Tag ID 1027 (see 7.3.6.7.7)

7.3.6.1.2 FX-DCC Eligibility Response

Tag: FF8804
Length: Variable
Value: FX-DCC Eligibility Response

Tag	Value	
FF8804	FX-DCC Eligibility Response (tag ID 1028)	
	Tag	**Value**
	DF876E	FX-DCC Eligibility Flag (tag ID 1006) – Boolean '1' for eligible, '0' for ineligible.

Table 10 FX-DCC Eligibility Response

The eligibility flag (tag ID 1006) is always present. Additional the sub-TLVs are included at the discretion of the implementer.

One or more of the following additional sub-TLVs may be included in the FX-DCC Eligibility Response based on the customer's needs.

1. FX-DCC Fast Cash Amounts – Tag ID 1016 (see 7.3.6.7.1). Unused fast cash amount sub-tags may be zeroed in the response or omitted altogether. For example if only 5 fast cash amount values are supported by the terminal, tags for fast cash amount values 6, 7, and 8 may be zeroed or left out of the PDU.

PRELIMINARY

7.3.6.2 FX-DCC Offer Determination

FX-DCC offer determination involves exchanging TDL messages to negotiate the terms of the offer to be presented to the customer and as required present disclaimers. This section describes those message elements required to do so.

7.3.6.2.1 FX-DCC Lookup Request

Tag: FF876F
Length: Variable
Value: FX-DCC Lookup Request

Tag	Value
FF876F	FX-DCC Lookup Request (tag ID 1007)

Table 11 - FX-DCC Lookup Request

This TLV is sent after the currency type and amount is selected at a terminal that offers FX-DCC transactions. The host needs this information to determine which type of FX/DCC offer, if any, to make to the customer.

One or more of the following tags must be included:

1. Requested Amount – Tag ID 1008 (see 7.3.6.7.8)
2. Surcharge Amount – Tag ID 114 (see 7.3.7.10)
3. Currency Code – Tag ID 5212, the ISO numeric currency code as defined for the generic fast cash item described in [2]
4. Selected Language - Tag ID 1024 the current language selected by the user (see 7.3.6.7.3)
5. Disclaimer Region Capacity - Tag ID 1025 – the maximum characters that can fit into the disclaimer region (see 7.3.6.7.4)
6. PAN – Tag ID 1027 (see 7.3.6.7.7)

7.3.6.2.2 FX-DCC Lookup Response

Tag: FF8771
Length: Variable
Value: FX-DCC Lookup Response

Tag	Value
FF8771	FX-DCC lookup response PDU (tag TD 1009)

Table 12 - FX-DCC Lookup Response

The FX-DCC Lookup Response must contain one or more of the following tags:

1. FX-DCC Offer – Tag ID 1010 (see 7.3.6.7.9)
2. Custom Screen Text – Tag ID 9– this is the disclaimer packaged as a custom screen text TLV. See section 7.3.3 for details
3. FX-DCC Eligibility - Flag Tag ID 1006 – Boolean '1' for eligible, '0' for ineligible.
4. FX-DCC Surcharge -Type Tag ID 1029 – (see 7.3.6.7.5)
5. FX-DCC Markup - Tag ID 1045 the markup, if any, associated with the transaction see section 7.3.6.7.21
6. FX-DCC Disclaimer Scheme - Tag ID 1030 (see 7.3.6.7.12)
7. Surcharge Amount – Tag ID 114 – Host supplied surcharge amount, if any, associated with the transaction. See section 7.3.7.10 for details. This may not be supported for all DCC applications.

PRELIMINARY

7.3.6.3 FX-DCC Working Key Acquisition

This section describes tag for acquiring a PIN or MAC working key for FX-DCC transactions

7.3.6.3.1 FX-DCC Working Key Request

Tag: FF880B
Length: Variable
Value: FX-DCC Working Key Request

Tag	Value
FF880B	FX-DCC Working Key Request (tag ID 1035)

Figure 4 FX-DCC PIN Working Key Request

One or more of the following tags must be included based on the customer's needs:

1. Target Tag ID – Tag ID 138 identifying the specified working key see Table 48 in section 7.3.7.38 for details.

7.3.6.3.2 FX-DCC PIN Working Key Response

Tag: FF880C
Length: Variable
Value: FX-DCC Working Key Response

Tag	Value
FF880C	FX-DCC Working Key Response (tag ID 1036)

Figure 5 FX-DCC PIN Working Key Response

One or more of the following tags must be included based on the customer's needs:

1. PIN Working Key – Tag ID 137 containing the requested working key value, see section 7.3.7.38 for details.

7.3.6.4 FX-DCC Transaction Processing

This section defines a tag for initiating an FX-DCC transaction with a host.

7.3.6.4.1 FX-DCC Transaction Request

Tag: FF8807
Length: Variable
Value: FX-DCC Transaction Request:

Tag	Value
FF8807	FX-DCC Transaction Request PDU (tag TD 1031)

Table 13 FX-DCC Transaction Request

The FX-DCC Transaction Request must contain one or more of the following tags:

1. Offer Acceptance Flag – Tag 1033, see section 7.3.6.7.13
2. Requested Amount – Tag 1008, see section 7.3.6.7.8
3. Requested Currency Code – Tag ID 5212, the ISO numeric currency code as defined for the generic fast cash item described in [2]

PRELIMINARY

4. Surcharge Amount – Tag 114, see section 7.3.7.10
5. Presented Amount – Tag 1034, see section 7.3.6.7.14
6. Track Data – Tag 105, see section 7.3.7.5
7. ANSI standard PIN Block – Tag 109, see section 7.3.7.6
8. Source Account Identifier – Tag 102, see section 7.3.7.3

Depending on the implementation, converted amounts may be provided by the host or calculated at the terminal. If the implementation uses host calculated amounts, the following tags should be included:

1. Converted Amount – Tag 1014, see section 7.3.6.7.11
2. Converted Currency Code – Tag 1012, the ISO numeric billing/settlement currency code
3. Converted Surcharge Amount – Tag 1013, see section 7.3.6.7.10

For EMV transactions, the following tags should also be included:

1. EMV Data – Tag 148, see section 7.3.7.46
2. Triton Specific EMV Data (if applicable) – Tag 149, see section 7.3.7.47

7.3.6.4.2 FX-DCC Transaction Response

Tag: FF8808
Length: Variable
Value: FX-DCC Transaction Response:

Tag	Value
FF8808	FX-DCC Transaction Response PDU (tag TD 1032)

Table 14 FX-DCC Transaction Response

The FX-DCC Transaction Response must contain one or more of the following tags:

1. Authorization Code – Tag 112 (Mandatory) see section 7.3.7.9
2. Authorization Number – Tag 117 (Mandatory) see section 7.3.7.13
3. Business Date – Tag 144 (Mandatory) see section 7.3.7.44

For EMV transactions, the following tags should also be included:

1. EMV Data – Tag 148, see section 7.3.7.46
2. Triton Specific EMV Data (if applicable) – Tag 149, see section 7.3.7.47

7.3.6.5 FX-DCC Reversal Processing

This section defines a tag for initiating an FX-DCC reversal with a host.

7.3.6.5.1 FX-DCC Reversal Request

Tag: FF8816
Length: Variable
Value: FX-DCC Reversal Request:

Tag	Value
FF8816	FX-DCC Reversal Request PDU (tag TD 1046)

Table 15 FX-DCC Reversal Request

PRELIMINARY

The FX-DCC Reversal Request must contain one or more of the following tags:

1. Reversal Reason – Tag 121 is the reason for the reversal, see section 7.3.7.17. The enumerations for this integer value may be extended for FX-DCC usage.
2. Offer Acceptance Flag – Tag 1033, see section 7.3.6.7.13
3. Requested Amount – Tag 1008 is the amount requested, see section 7.3.6.7.19
4. Dispensed Amount – Tag 1044 is the amount dispensed, see section 7.3.6.7.20
5. Numeric Currency Code – Tag ID 5212, the ISO numeric currency code as defined for the generic fast cash item described in [2].
6. Surcharge Amount – Tag 114 the surcharge amount, if any, associated with the transaction see section 7.3.7.10
7. Presented Amount – Tag 1034, see section 7.3.6.7.14
8. Track Data – Tag 105, see section 7.3.7.5

For EMV transactions, the following tags should also be included:

1. EMV Data – Tag 148, see section 7.3.7.46
2. Triton Specific EMV Data (if applicable) – Tag 149, see section 7.3.7.47

7.3.6.5.2 FX-DCC Reversal Response

Tag: FF8817
Length: Variable
Value: FX-DCC Reversal Response:

Tag	Value
FF8817	FX-DCC Reversal Response PDU (tag TD 1047)

Table 16 FX-DCC Reversal Response

The FX-DCC Reversal Response must contain one or more of the following tags:

1. Authorization Code – Tag 112 (Mandatory) see section 7.3.7.9

7.3.6.6 FX-DCC Close Transaction Processing

This section defines tags for initiating FX-DCC close transaction with a host.

7.3.6.6.1 FX-DCC Close request

Tag: FF880D
Length: Variable
Value: FX-DCC Close Request:

Tag	Value
FF880D	FX-DCC Close Request (tag ID 1037)

Table 17 FX-DCC Close Request

The FX-DCC Close Request must contain one or more of the following tags per customer requirements:

1. Close Type – Tag 1039, see section 7.3.6.7.15
2. Transaction Count – Tag 1040, see section 7.3.6.7.16

PRELIMINARY

3. Settlement Total – Tag 1041, see section 7.3.6.7.17

7.3.6.6.2 FX-DCC Close Response

Tag: FF880D
Length: Variable
Value: FX-DCC Close Request:

Tag	Value
FF880E	FX-DCC Close Response (tag ID 1038)

Table 18 FX-DCC Close Response

The FX-DCC Close Request must contain one or more of the following tags per customer requirements:

1. Close Reset Flag – Tag 1042, see section 7.3.6.7.18

7.3.6.7 FX-DCC Supplemental TLVs

7.3.6.7.1 FX-DCC Fast Cash Amounts List

Tag: FF8778
Length: Variable
Value: FX-DCC Fast Cash Amounts List:

Tag	Value	
FF8778	FX-DCC Fast Cash Amounts List (tag ID 1016)	
	Tag	**Value**
	FFA85B	FX-DCC Fast Cash Amount Entry (tag ID 5211) see [2]

Table 19 – FX-DCC Fast Cash Amounts List

7.3.6.7.2 FX-DCC Offer Disclaimer

The FX-DCC disclaimer is a custom screen text TLV as described below is a custom screen text item as defined in section 7.3.3.

7.3.6.7.3 Current Language Selected

Tag: DF8800
Length: Variable
Value: FX-DCC Current Language Selected:

Tag	Value
DF8800	Current language selected at the terminal (tag ID 1024) Type is TBD.

The value to be used for indicating the current language selected at the terminal is still TBD at this time.

7.3.6.7.4 Disclaimer Region Capacity

Tag: DF8801
Length: Variable
Value: FX-DCC Disclaimer Region Capacity:

PRELIMINARY

Tag	Value
DF8801	Maximum Characters in Region to Place Disclaimer Text (tag ID 1025) Integer.

7.3.6.7.5 FX-DCC Surcharge Type

Tag: FF8805
Length: Variable
Value: FX-DCC Surcharge Type:

Tag	Value
DF8805	FX-DCC Surcharge Type (tag ID 1029) – Integer (see Table 21)

Table 20 FX-DCC Surcharge Type

FX-DCC Surcharge Types:

ID	Description
1	No surcharge is to be applied
2	Standard withdrawal surcharge
3	Host supplied surcharge (if supported)

Table 21 FX-DCC Surcharge Type Enumerations

7.3.6.7.6 Cassette Configuration

Tag: FF8768
Length: Variable
Value: Cassette Configuration:

Tag	Value	
FF8768	Cassette Configuration (tag ID 1000)	
	Tag	**Value**
	DF8769	Dispenser mechanism ID (tag ID 1001) Integer
	FF876A	Cassette configuration information (tag ID 1002)

Table 22 – Terminal Cassette Configuration

Each terminal cassette configuration TLV identifies the dispenser type and one or more cassette configurations for the dispenser.

Valid Dispenser Mechanism IDs (to date):

ID	Description
0	Not Present
1	NMD-20
2	NMD-50
3	NMD-100
4	SDD-1700
5	Mini-Mech
6	TDM Single-cassette
7	TDM Multi-cassette

Table 23 – Valid Dispenser Mechanism IDs

Tag: FF876A
Length: Variable
Value: Cassette Configuration Information:

Tag	Value	
FF876A	Cassette configuration information (tag ID 1002)	
	Tag	**Value**
	DF876B	Cassette ID (tag ID 1003) String
	DFA85C	Numeric currency code (tag ID 5212) Unsigned Integer
	DF876D	State of the cassette (tag ID 1005) Integer

Table 24 Cassette Configuration Information

Valid Cassette State Values:

State Value	Description
1	Enabled
2	Disabled

Table 25 – Valid Cassette State Values

Cassette ID sub-TLVs will be a printable string with one of the alpha characters in the set [A | B | C | D] expressed as "Cassette A" for example.

7.3.6.7.7 FX-DCC Customer PAN

The customer PAN can be used as eligibility criteria.

Tag: DF8803
Length: Variable
Value: FX-DCC Primary Account Number:

Tag	Value
DF8803	Primary Account Number (PAN) (tag ID 1027) String

For some applications the PAN may be enough information to make the determination for offering an FX-DCC offer.

7.3.6.7.8 FX-DCC Requested Amount

Tag: DF8770
Length: Variable
Value: FX-DCC Requested Amount

Tag	Value
DF8770	Requested amount (tag ID 1008) Double

7.3.6.7.9 FX-DCC Offer

Tag: DF8772
Length: Variable
Value: FX-DCC Offer

PRELIMINARY

Tag	Value	
FF8772	FX-DCC offer (tag ID 1010)	
	Tag	**Value**
	DF8773	FX-DCC offer type (tag ID 1011) Integer (see table below)
	DFA900	FX-DCC Conversion Rate (tag ID 5248) Double (see [2] for description)
	DF8776	FX-DCC Converted amount (tag ID 1014) Double
	DFA85C	FX-DCC Billing/Settlement Currency Code (tag ID 5212) Unsigned Integer
	DFA901	Quotation Type (tag ID 5249) Unsigned Integer

Table 26 FX-DCC Offer

Valid Offer Types:

ID	Description
1	DCC – Direct Currency Conversion
2	FX – Foreign Exchange with complete conversion details
3	FX – "Scheme" offer, the amount returned is expressed in terms of the requested currency along with the commission. The other fields are meaningless and will be ignored

The conversion rate, converted amount and commission are all floating point values encoded as ASCII strings with an explicit decimal point and at least one ASCII decimal character on either side of the decimal point. Some implementations may contain more than one offer of differing offer types. Multiple offers of the same type in the same PDU are discarded.

7.3.6.7.10 FX-DCC Converted Surcharge Amount

Tag: DF8775
Length: Variable
Value: FX-DCC Converted Surcharge Amount

Tag	Value
DF8775	FX-DCC Converted Surcharge Amount (tag TD 1013) Double

Table 27 FX-DCC Converted Surcharge Amount

This value is the converted surcharge amount in the billing currency for the FX-DCC transaction. This should be based on the FX-DCC conversion rate and local surcharge amount.

7.3.6.7.11 FX-DCC Converted Amount

Tag: DF8776
Length: Variable
Value: FX-DCC Converted Amount

Tag	Value
DF8776	FX-DCC Converted Amount (tag TD 1014) Double

Table 28 FX-DCC Converted Amount

This value is the converted requested amount in the billing currency for the FX-DCC transaction. This should be based on the FX-DCC conversion rate and local requested amount.

7.3.6.7.12 FX-DCC Disclaimer Scheme

Tag: DF8806
Length: Variable
Value: FX-DCC Disclaimer Scheme

Tag	Value
DF8806	FX-DCC Disclaimer Scheme (tag ID 1030) Integer (see Table 30)

Table 29 FX-DCC Disclaimer Scheme

The disclaimer scheme identifies specific text to be displayed (if any) when making the FX or DCC offer.

ID	Description
0	None
1	VISA
2	MasterCard

Table 30 Vaild FX-DCC Disclaimer Schemes

7.3.6.7.13 FX-DCC Offer Acceptance Flag

Tag: DF8809
Length: Variable
Value: FX-DCC Offer Acceptance Flag:

Tag	Value
DF8809	FX-DCC Offer Acceptance Flag (tag TD 1033) Boolean

Table 31 FX-DCC Offer Acceptance Flag

This value indicates that the customer has or has not accepted the FX-DCC offer.

7.3.6.7.14 FX-DCC Presented Amount

Tag: DF880A
Length: Variable
Value: FX-DCC Presented Amount:

Tag	Value
DF880A	FX-DCC Presented Amount (tag TD 1034) Double

Table 32 FX-DCC Presented Amount

This value indicates that the converted amount in the billing currency that was presented to the customer in the FX-DCC offer.

7.3.6.7.15 FX-DCC Close Type

Tag: DF880F
Length: Variable
Value: FX-DCC Close Type:

Tag	Value
DF880F	FX-DCC Close Type (tag ID 1039) – Integer

Table 33 FX-DCC Close Type

PRELIMINARY

The close type indicates a value from the table below:

Close Types	Description
1	Trial Day Close
2	Day Close

7.3.6.7.16 FX-DCC Transaction Count

Tag: DF8810
Length: Variable
Value: FX-DCC Transaction Count:

Tag	Value
DF8810	FX-DCC Transaction Count (tag ID 1040) – Unsigned Integer

Table 34 FX-DCC Transaction Count

The cumulative transaction count since last totals reset.

7.3.6.7.17 FX-DCC Settlement Amount

Tag: DF8811
Length: Variable
Value: FX-DCC Settlement Amount:

Tag	Value
DF8811	FX-DCC Settlement Total (tag ID 1041) – Double

Table 35 FX-DCC Settlement Amount

The cumulative settlement amount since last totals reset.

7.3.6.7.18 FX-DCC Close Reset Flag

Tag: DF8812
Length: Variable
Value: FX-DCC Close Reset Flag:

Tag	Value
DF8812	FX-DCC Close Reset Flag (tag ID 1042) – Boolean

Table 36 FX-DCC Close Reset Flag

This flag indicates TRUE ('1') if the terminal is to reset FX-DCC totals at conclusion of the close operation, FALSE ('0') otherwise.

7.3.6.7.19 FX-DCC Requested Amount

Tag: DF8770
Length: Variable
Value: FX-DCC Requested Amount

PRELIMINARY

Tag	Value
DF8770	FX-DCC Requested Amount (tag ID 1008) Double

Table 37 FX-DCC Requested Amount

7.3.6.7.20 FX-DCC Dispensed Amount

Tag: DF88814
Length: Variable
Value: FX-DCC Dispensed Amount

Tag	Value
DF8814	FX-DCC Dispensed Amount (tag ID 1044) Double

Table 38 FX-DCC Dispensed Amount

7.3.6.7.21 FX-DCC Markup

Tag: DF88815
Length: Variable
Value: FX-DCC Markup

Tag	Value
DF8815	FX-DCC Markup (tag ID 1045) Double

Table 39 FX-DCC Markup

7.3.7 TDL Transaction Data

This section describes the TLV data items required for the full spectrum Host/Terminal transaction sessions. Most transactions will consist of constructed tag FF8112 (tag ID 146 transaction request) or tag FF8113 (tag ID 147 transaction response) and one or more of the sub-tags defined in this section.

7.3.7.1 Terminal Identification

The Terminal Identification (TI) TLV identifies the terminal equipment involved in the transaction.

Tag	Value
DF64	Terminal Identifier (tag ID 100) is a variable length string identifying the terminal equipment in the transaction.

7.3.7.2 Transaction Type Identification

The Transaction Type Identification (TTI) TLV unambiguously identifies the transaction type associated with a message exchange through TDL.

Tag	Value
DF65	Transaction Type Identifier (tag ID 101). This TLV will defines the list of the basic transactions available through TDL (see Table 40)

TID	Description
1	Withdrawal Transaction
2	Balance Inquiry
3	Cash Transfer
4	Purchase Transaction

PRELIMINARY

5	Wireless Phone Refill
6	PIN Change
7	PIN Unlock
8	Bill Payment
9	Prepaid Utilities
10	POS

Table 40 TDL Transaction Identifiers

7.3.7.3 Account Type Definition

Account Type Identification (ATI) TLVs unambiguously identifies source and/or destination account types involved for the transaction associated with a message exchange through TDL[2].

Tag	Value
DF66	Source Account Identifier (tag ID 102). This TLV will define the list of account identifiers available through TDL (see Table 41) for the source account.

Tag	Value
DF67	Destination Account Identifier (tag ID 103). This TLV will define the list of account identifiers available through TDL (see Table 41) for the destination account.

ATI	Description
1	Checking
2	Savings
3	Credit
4	Brokerage
N	Etc.

Table 41 TDL Account Identifiers

7.3.7.4 Transaction Sequence Number

The Transaction Sequence Number (TSN) TLV identifies the message sequence for request/response correlation.

Tag	Value
DF68	Transaction Sequence number (tag ID 104) value type is ASCII represented hexadecimal integer.

7.3.7.5 Track Data Group

The Track Data (TDG) TLV defines the track data from the card used in the transaction. The terminal can provide a variable number of track data entries for the transaction as required for the specific feature.

Tag	Value		
FF69	Track Data Group (tag ID 105) Contains one or more tag 106 items.		
	Tag	Value	
	FF6A	Track Data Item (tag ID 106)	
		Tag	Value

[2] This account ID is used for transactions where source, destination or both account IDs are required, e.g. Cash Transfers etc.

Highly Confidential: Property of Triton Systems – Do Not Disclose

		DF6B	Track Number (tag ID 107)
		DF6C	Track Data (tag ID 108)

7.3.7.6 Encrypted PIN Block

The Encrypted PIN Block (EPB) TLV contains the ANSI standard encrypted PIN block.

Tag	Value
DF6D	ANSI standard encrypted PIN block (tag ID 109)

7.3.7.7 Transaction Amount

The Transaction Amount (TA) TLV is the amount requested in a transaction for withdrawal, purchase, or transfer. The value type is as ASCII represented double.

Tag	Value
DF6E	Requested amount (tag ID 110)

7.3.7.8 Transaction Date

The Transaction Date TLV is the date of transaction in UTC format. It is set by the processor. The value type is numeric. Ex. 2010-12-05T19:39:10.

Tag	Value
DF6F	Transaction Date (tag ID 111)

7.3.7.9 Authorization Code

The Authorization Code TLV describes the condition of the transaction. The value type is numeric.

Tag	Value
DF70	Authorization Code (tag ID 112) Integer

7.3.7.10 Surcharge Amount

The surcharge amount TLV is the surcharge amount specific to the requested transaction. The value type is as ASCII represented double.

Tag	Value
DF72	Surcharge amount (tag ID 114) Double

Table 42 Surcharge Amount

7.3.7.11 ATM Transaction Record

The ATM Transaction Record constructed TLV contains an ATM specific transaction. TLVs within this construct have already been defined and are specific to the transaction.

Tag	Value	
FF73	ATM Transaction Record (tag ID 115)	
	Tag	Value
	DF6F	Transaction Date (tag ID 111)
	DF65	Transaction Type Identifier (tag ID 101)
	DF66	Source Account Identifier (tag ID 102)

PRELIMINARY

	DF68	Transaction Sequence Number (tag ID 104)
	DF70	Authorization Code (tag ID 112)
	DF72	Surcharge Amount (tag ID 114)
	DF74	Account Number (tag ID 116)
	DF75	Authorization Number (tag ID 117)
	DF810B	Language Selected (tag ID 139)
	FFA858	Terminal Message Item (Store Message) (tag ID 5208)
	DF900E	Currency Code (tag ID 2062)
	DF76	Ledger Balance (tag ID 118) (if applicable)
	DF77	Available Balance (tag ID 119) (if applicable)
	DF78	Reversal Status (tag ID 120) (if applicable)
	DF79	Reversal Reason (tag ID 121) (if applicable)
	DF7A	Receipt Message (tag ID 122) (if applicable)
	FF7B	Extended Transaction Record Data (tag ID 123) (if applicable)
	FF8104	Notification Info (tag ID 132)

7.3.7.12 Account Number

The Account Number TLV specifies card number of the card holder's card used for the transaction. The value type is a 16 character String in the form: 000000========0000.

Tag	Value
DF74	Account Number (tag ID 116)

7.3.7.13 Authorization Number

The Authorization Number TLV specifies the transaction's authorization number. The value type is an 8 digit ASCII represented unsigned integer.

Tag	Value
DF75	Authorization Number (tag ID 117)

7.3.7.14 Ledger Balance

The Ledger Balance TLV specifies the ledger balance of the account used for the transaction. The value type is an ASCII represented double.

Tag	Value
DF76	Ledger Balance (tag ID 118)

7.3.7.15 Available Balance

The Available Balance TLV specifies the available balance of the account used for the transaction. The value type is an ASCII represented double.

Tag	Value
DF77	Available Balance (tag ID 119)

7.3.7.16 Reversal Status

The Reversal Status TLV is present only if the terminal needed to send a reversal. It specifies whether or not the reversal was successful. The value type is an ASCII represented unsigned integer.

PRELIMINARY

Tag	Value
DF78	Reversal Status (tag ID 120)

Valid Reversal Statuses:

Reversal Status	Description
1	Successful
2	Unsuccessful
3	In Progress
4	Denied by Host

Table 43 Reversal Statuses

7.3.7.17 Reversal Reason

The reversal reason is sent in reversal requests indicating the reason a transaction is to be reversed.

Tag	Value
DF79	Reversal Reason (tag ID 121) Integer – See Table 44

Reversal Reason	Description
1	Incorrect Dispense: The amount dispensed does not match amount requested. This may be due to either a mechanical failure or the lack of available bill(s) to make the requested amount
2	Protocol Error: A request message was sent from the terminal for which no valid response message was received from the host
3	Lost EOT: A valid response message was received by the terminal, but the EOT was not received. This assumes the terminal was configured to expect the EOT. If the terminal is configured to ignore EOT, there will be no reversal generated
4	Mac Error: The calculated MAC result does not match the MAC result returned in the response message
5	Customer Cancellation: Customer decided not to continue after reviewing the transaction fee
6	VAS Challenge/Response Error. An incorrect response was received in the 'vd' FID
7	ICC declined transaction
8	Offline PIN has not changed (the ATM could not communicate with the card)
9	Offline PIN Management Request incomplete (no response from card after issuer script sent to card)
10	Offline PIN Management failed (card sent message indicating the PIN Management Request Failed)
11	Customer did not remove card from reader

Table 44 Reversal Reason Enumerations

7.3.7.18 Receipt Message

The Receipt Message TLV contains a generic message to be printed onto the receipt. The value type is a variable length String.

Tag	Value
DF7A	Receipt Message (tag ID 122)

PRELIMINARY

7.3.7.19 Deposit Request Message

This section describes the TDL tag for a cash deposit request. For cash-in operations at a terminal designed for deposit, the raw message form would be as follows:

Tag	Value				
FF8374	CashDepositTransactionRequest (tag ID 500)				
	Tag	**Value**			
	FF8375	CustomerInfo (tag ID 501)			
		Tag	**Value**		
		FF69	TrackDataGroup (tag ID 105)		
			Tag	**Value**	
			FF6A	TrackDataItem (tag ID 106) Track 1	
				Tag	**Value**
				DF6B	TrackNumber (tag ID 107) Integer
				DF6C	TrackData (tag ID 108) String
			FF6A	TrackDataItem (tag ID 106) Track N	
	FF8376	DepositTransactionData (tag ID 502)			
		DF67	DestiationAccountIdentifier (tag ID 103) Integer		
		DF8377	TotalDepositAmountCash (tag ID 503) Double		
		FF8378	NoteCounts (tag ID 504)		
			DFA85C	CurrencyCode (tag ID 5212) Unsigned Integer	
			FF8379	NoteCount (tag ID 505) **Entry 1**	
				Tag	**Value**
				DF837A	Denomination (tag ID 506) Double
				DF837B	Count (tag ID 507) Integer
			FF8379	NoteCount (tag ID 505) **Entry N**	

This is primarily an example and may contain more/less tags as required for tailoring to an implementation's needs. The request message tag elements specific to deposit type transaction apart from the higher layer tag IDs 500 and 501 are the deposit transaction data (tag ID 502), total deposit amount (tag ID 503) and the note count tags (tag ID 504). The total deposit amount is the total amount deposited. The note counts, one per denomination, include the denomination and count per denomination for each different denomination accounted for in the deposit. The sum of all note counts, count (tag ID 507) times denomination (tag ID 506) must equal the total deposit amount.

7.3.7.20 Deposit Transaction Response

This section describes the TDL tag for a cash deposit transaction response. For cash-in operations at a terminal designed for deposit, the raw message form would be as follows:

Tag	Value	
FF837C	CashDepositTransactionResponse (tag ID 508)	
	Tag	**Value**
	DF8377	TotalDepositAmountCash (tag ID 503) Double

7.3.7.21 Deposit Totals Reporting

This section assumes the terminal is capable of tracking deposit transaction totals as follows:

1. Total Amount Deposited Cash – This is a total of all cash deposits successfully made since the last reporting period
2. Total Number of Cash Deposit Transactions – The total number of attempts made to deposit cash since the last reporting period, (i.e. the total number of deposit transactions that included cash acceptance at the terminal)
3. Total Amount Deposited Cash per Denomination - this is a total amount deposited successfully per denomination since the last reporting period. The total amount is evenly divided by the denomination.

The deposit count totals data are reported through the REPORT-DATA Request/Response tags.

Tag	Value		
FF837D	DepositTotalsInformation (tag ID 509)		
	Tag	**Value**	
	FF837F	CashDepositTotals (tag ID 511)	
		Tag	**Value**
		DF8400	TotalAmountDeposited (tag ID 512) Double
		DF8401	TotalTransactions (tag ID 513) Unsigned
		FF8378	NoteCounts (tag ID 504)[3]

The response to a report providing totals for deposit transactions can stipulate whether or not current totals can be cleared by the terminal.

Tag	Value	
FF837E	CashDepositTotalsAcknowledge (tag ID 510)	
	Tag	**Value**
	DF8403	ClearTotalsFlag (tag ID 514) Boolean

7.3.7.22 Extended Transaction Record Data

This response TLV contains the record information requested by the host.

Tag	Value	
FF7B	Extended Transaction Record Data (tag ID 123)	
	Tag	**Value**
	VAR	Constructed TLV based on request

Extended Transaction Record TLVs:

Tag	Value
FF8116	Withdrawal Transaction Record (tag ID 150)
FF8117	Transfer Transaction Record (tag ID 151)

Table 45 Extended Transaction Record TLVs

7.3.7.23 Withdrawal Transaction Record

The Withdrawal Transaction Record constructed TLV contains a transaction record of a withdrawal transaction.

Tag	Value

[3] Same tag as defined for the cash deposit transaction request but contain the cumulative totals per denomination since last cleared.

FF8116	Withdrawal Transaction Record (tag ID 150)	
	Tag	**Value**
	DF6E	Transaction Amount (tag ID 110)
	DF7C	Dispensed Amount (tag ID 124)
	DF7D	Dispense Status (tag ID 125)
	FF7E	Dispensed Count by Cassette (tag ID 126)
	FF7F	Rejected Count by Cassette (tag ID 127)

7.3.7.24 Transfer Transaction Record

The Transfer Transaction Record constructed TLV contains a transaction record of a transfer transaction.

Tag	**Value**	
FF8117	Transfer Transaction Record (tag ID 151)	
	Tag	**Value**
	DF6E	Transaction Amount (tag ID 110)
	DF67	Destination Account Identifier (tag ID 103)

7.3.7.25 Dispensed Amount

The Dispensed Amount TLV specifies the amount of money dispensed during the transaction. The value type is an ASCII represented double.

Tag	**Value**
DF7C	Dispensed Amount (tag ID 124)

7.3.7.26 Dispense Status

The Dispense Status TLV contains the status of the dispense. The value type is an ASCII represented integer.

Tag	**Value**
DF7D	Dispense Status (tag ID 125)

7.3.7.27 Dispensed Count by Cassette

The Dispensed Count by Cassette constructed TLV contains the amount of bills each cassette dispensed during the transaction.

Tag	**Value**	
FF7E	Dispensed Count by Cassette (tag ID 126)	
	Tag	**Value**
	DF8100	Cassette A (tag ID 128)
	DF8101	Cassette B (tag ID 129)
	DF8102	Cassette C (tag ID 130)
	DF8103	Cassette D (tag ID 131)

7.3.7.28 Rejected Count by Cassette

The Rejected Count by Cassette constructed TLV contains the amount of bills each cassette rejected during the transaction.

Tag	**Value**

PRELIMINARY

FF7F	Rejected Count by Cassette (tag ID 127)	
	Tag	**Value**
	DF8100	Cassette A (tag ID 128)
	DF8101	Cassette B (tag ID 129)
	DF8102	Cassette C (tag ID 130)
	DF8103	Cassette D (tag ID 131)

7.3.7.29 Cassette A

The Cassette A TLV specifies the amount of bills dispensed from cassette A during the transaction. The value type is an ASCII represented unsigned integer.

Tag	Value
DF8100	Cassette A (tag ID 128)

7.3.7.30 Cassette B

The Cassette B TLV specifies the amount of bills dispensed from cassette B during the transaction. The value type is an ASCII represented unsigned integer.

Tag	Value
DF8101	Cassette B (tag ID 129)

7.3.7.31 Cassette C

The Cassette C TLV specifies the amount of bills dispensed from cassette C during the transaction. The value type is an ASCII represented unsigned integer.

Tag	Value
DF8102	Cassette C (tag ID 130)

7.3.7.32 Cassette D

The Cassette D TLV specifies the amount of bills dispensed from cassette D during the transaction. The value type is an ASCII represented unsigned integer.

Tag	Value
DF8103	Cassette D (tag ID 131)

7.3.7.33 Notification Info

The Notification Info constructed TLV contains information the host needs to send an electronic receipt.

Tag	Value	
FF8104	Notification Info (tag ID 132)	
	Tag	**Value**
	DF8105	Message Type (tag ID 133)
	DF8106	Recipient Address (tag ID 134)
	DF810F	Opt In Flag (tag ID 143)

7.3.7.34 Message Type

The Message Type TLV is an enumeration used by the terminal to specify what type of message the host should send. The value type is an ASCII represented unsigned integer.

PRELIMINARY

Tag	Value
DF8105	Message Type (tag ID 133)

Valid Message Types:

Message Type ID	Description
1	Short Message Service (SMS text message)
2	Email

Table 46 E-Receipt Message Types

7.3.7.35 Recipient Address

The Recipient Address TLV specifies the location to send the electronic receipt. It may be an email address or a phone number. The value type is a variable length String.

Tag	Value
DF8106	Recipient Address (tag ID 134)

7.3.7.36 Unique Identifier

The Unique Identifier TLV specifies the unique identifier used by Versalink to identify a terminal. The value type is a variable length String.

Tag	Value
DF8107	Unique Identifier (tag ID 135)

7.3.7.37 Disable SMS Command

This command TLV allows the host to disable the terminal's ability to offer electronic receipts to be sent by SMS.

Tag	Value
FF8108	Disable SMS Command (tag ID 136)

The response to this command will be a Continue Session message. The session indicator will be set to SUCCESS if the operation was successful.

Exceptions:

- If any mandatory elements are missing from the message, a Continue Session message with a FAILED Session Indicator and Error Code 9004 – Parse Error will be returned.

7.3.7.38 Working Key

The working key TLV is constructed of a key type identifier and the key value. The working key value consists of 32 ASCII characters. Each character represents a 4-bit hexadecimal numeral (0-9, A-F).

Tag	Value	
FF8109	Working Key (tag ID 137)	
	Tag	**Value**
	DF810A	Working Key ID (tag ID 138) Integer (see Table 48)
	DF71	Working Key (tag ID 113) String

Table 47 Working Key

PRELIMINARY

Working Key ID	Description
1	PIN Working Key
2	MAC Working Key

Table 48 Working Key ID

7.3.7.39 Language Selected

The Language Selected TLV specifies the language used for the transaction. The value type is ASCII represented unsigned integer.

Tag	Value
DF810B	Language Selected (tag ID 139)

Valid Languages:

Language	Description
1	English
2	Spanish
3	French
4	German

Table 49 Languages

7.3.7.40 Enable SMS Message

This message TLV allows the terminal to ask the host whether or not it can enable SMS functionality.

Tag	Value
FF810C	Enable SMS Message (tag ID 140)

The response to this command will be an Enable SMS Response message.

7.3.7.41 Enable SMS Response

This message TLV allows the host to indicate to the terminal whether or not this terminal can offer SMS functionality.

Tag	Value	
FF810D	Enable SMS Response Message (tag ID 141)	
	Tag	Value
	DF810E	Enable SMS (tag ID 142)

The response to this command will be a Continue Session message. The session indicator will be set to SUCCESS if the operation was successful.

Exceptions:

- If any mandatory elements are missing from the message, a Continue Session message with a FAILED Session Indicator and Error Code 9004 – Parse Error will be returned.

7.3.7.42 Enable SMS

The Enable SMS TLV specifies the whether or not the terminal has permission from the host to support SMS functionality. The value type is an ASCII represented Boolean.

Tag	Value

PRELIMINARY

DF810E	Enable SMS (tag ID 142)

7.3.7.43 Opt In Flag

The Opt In Flag TLV specifies whether or not the electronic receipt recipient is giving permission to share his/her email address. The value will always be false (0) if the message type is SMS. The value type is an ASCII Boolean.

Tag	Value
DF810F	Opt In Flag (tag ID 143)

7.3.7.44 Business Date

The business date of the transaction as set by the processor. The value type is of type Date.

Tag	Value
DF8110	Business Date (tag ID 144)

7.3.7.45 ATM Report Record

The ATM Report Record constructed TLV contains an ATM management functions report text to be emailed to the address within the Notification Info construct.

Tag	Value	
FF8111	ATM Report Record (tag ID 145)	
	Tag	**Value**
	DF9003	Text (tag ID 2051)
	FF8104	Notification Info (tag ID 132)

7.3.7.46 EMV Data

Contains the EMV data elements required for the transaction. The value type is a String. Please see section 10.10 for more information. This tag is equivalent to the 'ud' FID.

Tag	Value
DF8114	EMV Data (tag ID 148)

7.3.7.47 Triton Specific EMV Data

Contains the Triton Specific EMV data elements required for the transaction. The value type is a String. Please see section 10.10 for more information. This tag is equivalent to the 'uh' FID.

Tag	Value
DF8115	Triton Specific EMV Data (tag ID 149)

7.3.8 Prepaid/Gift Card Services

This section describes the information definitions for supporting prepaid/gift card (PPGC) services with TDL. When a Triton terminal supports purchase transactions of items that can be scanned for card ID information, these data items can be used for the required activation/deactivation operations with a specified host. The operations are performed as SET-DATA Request/Response primitives (see [2]) carried as application data in session request/response PDUs (see [2])

PRELIMINARY

7.3.8.1 Credentials

When requesting an activation or deactivation request, PPGC services provides credentials in the form of a Message Authentication Code (MAC) in the session tags to the activation host. Refer to [2] for details on how the session credentials are built.

7.3.8.2 Gift Card Operation Failure Information

When a gift card activation or deactivation operation fails the SET-DATA Response code enumeration tag DF16 (ID 22) will be set to the value 5 indicating operation failure. The activation host will include tag FFAE79 (ID 6009) in the SET_DATA Response as the feature/data specific failure tag.

Tag: FFAE79
Primitives: SET-DATA Response
Technical Description (Gift Card Operation Failure Information):

Tag	Value	
FFAE79	Gift Card Operation Failure Information (tag ID 6009)	
	Tag	**Value**
	FFAE7A	1st Gift Card Failure Entry (tag ID 6010)
	FFAE7A	Nth Gift Card Failure Entry (tag ID 6010)

Table 50 Gift Card Operation Failure Information

Tag: FFAE7A
Primitives: SET-DATA Response
Technical Description (Gift Card Failure Entry):

Tag	Value	
FFAE7A	Gift Card Failure Entry (tag ID 6010)	
	Tag	**Value**
	DFAE7B	Gift Card Serial Number (tag ID 6011) String
	DFAE7C	Gift Card Failure Code (tag ID 6012) Integer
	DFAE7D	Gift Card Failure Description (tag ID 6013) String

Table 51 List of Gift Card Serial Numbers

Response Code	Description	Additional Notes
1	Invalid Card Number	
2	Prior Activation	Serial number activated prior to current activation request
3	General Error	
4	Prior Deactivation	Serial number deactivated prior to current deactivation request
5	Internal Provider Error	Indicates a configuration issue on activation host's side. (e.g. A card missing a portion of its configuration data)
6	External Provider Error	Indicates that an error has occurred with either 1to1 or TxVia
7	Unable to Authenticate	
8	No prior activation	An attempt to deactivate a card not previously activated

Table 52 Gift Card Failure Codes

The gift card failure description tag (ID 6013) is optional at the host's discretion.

7.3.8.3 Gift Card Activation

The gift card activation information is supplied in tag ID 6000. This section describes the information encoded in tag ID 6000.

Highly Confidential: Property of Triton Systems – Do Not Disclose

Tag: FFAE70
Primitives: SET-DATA Request
Technical Description (Gift Card Activation):

Tag	Value	
FFAE70	Gift Card Activation (tag ID 6000)	
	Tag	**Value**
	DFAE7E	Requested Document Count (tag ID 6014)
	DFAE7F	Dispensed Document Count (tag ID 6015)
	DFAF00	Activation Status (tag ID 6016)
	FFAE71	Gift Card Activation List (tag ID 6001)

Table 53 Gift Card Activation

This data item provides information for activating gift cards purchased at a terminal. It also provides information in the presence of dispense and scan errors. The requested document count (tag ID 6014) is always the number of documents required to satisfy the purchase request. The dispensed document count (tag ID 6015) reflects the number documents delivered to the customer. The activation status (tag ID 5016) indicates the dispense/scan status at the terminal at the time the activation request is initiated.

Activation Status	Description
1	Success – all gift cards delivered with no rejects and all codes scanned
2	Scan Errors – all gift cards delivered with no rejects and not all codes scanned
3	Notes Rejected – Single rejects were detected in the dispense command, the values for dispensed document count are dependent on the device.
4	Cassette Empty – The cassette was emptied during the transaction
5	Dispense Errors – Device errors detected in the dispense command, the document dispensed count will be 0 and the activation list will be empty
6	Unknown Dispense – An error that prevents the terminal from knowing with 100% assurance that the customer did or did not receive their purchase. This can happen if for instance a dispenser's response was garbled or communications with the device timed out.

Table 54 Activation Status

Tag: FFAE71
Primitives: SET-DATA Request
Technical Description (Gift Card Activation List):

Tag	Value	
FFAE71	Gift Card Activation List (tag ID 6001)	
	Tag	**Value**
	FFAE72	1st Gift Card Activation Entry (tag ID 6002)
	FFAE72	Nth Gift Card Activation Entry (tag ID 6002)

Table 55 Gift Card Activation List

PRELIMINARY

Tag: FFAE72
Primitives: SET-DATA Request
Technical Description (Gift Card Activation Entry):

Tag	Value	
FFAE72	Gift Card Activation Entry (tag ID 6002)	
	Tag	**Value**
	DFAE73	Card ID (tag ID 6003) String
	DFAE74	Amount(tag ID 6004) Double
	DFA86D	Currency Code (Tag ID 5229) String

Table 56 Gift Card Activation Entry

See section 9.5 for gift card activation encoding examples.

7.3.8.4 Gift Card Deactivation

The gift card deactivation information is supplied in tag ID 6005. This section describes the information encoded in tag ID 6005.

Tag: FFAE75
Primitives: SET-DATA Request
Technical Description (Gift Card Deactivation):

Tag	Value	
FFAE75	Gift Card Deactivation (tag ID 6005)	
	Tag	**Value**
	FFAE76	Gift Card Deactivation List (tag ID 6006)

Table 57 Gift Card Deactivation

This data item provides information for deactivating gift cards purchased at a terminal.

Tag: FFAE76
Primitives: SET-DATA Request
Technical Description (Gift Card Deactivation List):

Tag	Value	
FFAE76	Gift Card Deactivation List (tag ID 6006)	
	Tag	**Value**
	FFAE77	1st Gift Card Deactivation Entry (tag ID 6007)
	FFAE77	Nth Gift Card Deactivation Entry (tag ID 6007)

Table 58 Gift Card Deactivation List

Tag: FFAE77
Primitives: SET-DATA Request
Technical Description (Gift Card Deactivation Entry):

Tag	Value	
FFAE77	Gift Card Deactivation Entry (tag ID 6007)	
	Tag	**Value**
	DFAE73	Card ID (tag ID 6003) String
	DFAE78	Reason (tag ID 6008) String

Table 59 Gift Card Deactivation Entry

See section 9.5 for gift card deactivation encoding examples.

PRELIMINARY

7.3.8.5 Performance Monitoring Statistics

The gift card service is capable of reporting simple counters that can be used for performance monitoring (PM). This section describes the message structure and counts currently supported by TDL.

Tag: FFAF01
Primitives: EVENT-Notify
Technical Description (Gift Card PM Statistics):

Tag	Value	
FFAF01	Gift Card PM Statistics (tag ID 6017)	
	Tag	**Value**
	DFAF02	Scan Error Count (tag ID 6018) Unsigned Integer
	DFAF03	Reject Event Count (tag ID 6019) Unsigned Integer

Table 60 Gift Card PM Statistics

The scan error count (tag 6018) records the current count of scan errors detected during gift card dispensing operations. The reject event count (tag 6018) records the current count of single reject events detected during gift card dispensing operations. The counts roll over to 0 after reaching MAX unsigned integer value (4294967295 decimal).

7.3.9 Cardless Transactions

This section describes the information definitions for supporting cardless transactions.

7.3.9.1 Cardless Transaction Request

Cardless messages exchanged between the terminal and cardless host will be TDL PDUs using session wrappers. The application data included in the session PDU (the payload of DFA710 tag ID 5008) for a Cardless Transaction Request will be a GET DATA Request. This is a request for the transaction amount.

Session PDU:

Tag	Value	
FFA708	Session PDU (tag ID 5000)	
	Tag	**Value**
	DFA709	Destination SAP (tag ID 5001) Integer (9 = Just.Cash Cardless)
	DFA70A	Session Message ID (tag ID 5002) Integer (7 = Single session message request)
	DFA70B	Source SAP (tag ID 5003) Integer (9 = Just.Cash Cardless)
	FFA70C	Session Credentials (tag ID 5004) (required for cardless transactions)
	DFA710	Application Data (tag ID 5008)

Table 61 Session PDU

Session Credentials:

Tag	Value	
FFA70C	Session Credentials (tag ID 5004)	
	Tag	Value
	DF811C	Cardless Session Key (tag ID 156) – String (R)

Table 62 Session Credentials

GET-DATA Request (this construct is immediately following the App Data (tag ID 5008)):

Tag	Value	
FF0D	GET-DATA Request (tag ID 13)	
	Tag	Value
	FF28	Correlation Information (tag ID 40)
	FF8118	Cardless Transaction Request (tag ID 152)

Table 63 GET-DATA Request

Correlation Information:

Tag	Value	
FF28	Correlation Information (tag ID 40)	
	Tag	Value
	DF64	Terminal Identifier (tag ID 100) String
	DF68	Transaction Sequence Number (tag ID 104) Hexadecimal Unsigned Integer.
	DF29	Correlation Timestamp (tag ID 41) Date-Time
	FFAA5B	Status Monitor (tag ID 5467) (optional)

Table 64 Correlation Information

Cardless Transaction Request:

Tag	Value	
FF8118	Cardless Transaction Request (tag ID 152)	
	Tag	Value
	DF811D	Cash Code (tag ID 157) – String (R)

Table 65 Cardless Transaction Request

7.3.9.2 Cardless Transaction Request Example

Messages to the cardless host will be sent using raw binary packing. For readability, in this document examples are shown using base64 packing.

T10001000084/6clXd+nCQE536cKATffpwsBOf+nDAnfgRwFSEVZMjTfpxA9/w06/ygm32QCSjHfaAgwMDAwMDEzON8pEzIwMTUtMDgtMTQgMDg6NTk6Mzf/gRgN34EdCTEyMzQ1Njc4OQ==

Unpacked Payload: The unpacked payload here shows the TLV structure. The payload of Tag ID 5008 (DFA710) is the GET-DATA Request TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 5D (Session Packet, length 93)
   DFA709 01 9 (Destination SAP, length 1, value "9")
   DFA70A 01 7 (Session Message ID, length 1, value "7")
   DFA70B 01 9 (Source SAP, length 1, value "9")
   FFA70C 09 (Session Credentials, length 9)
      DF811C 05 HEY24 (Session Key (Tag ID 156), length 5, value "HEY24")
   DFA710 3D FF0D3AFF2826DF64
            024A31DF68083030
            303030313338DF29
            13323031352D3038
            2D31342030383A35
            393A3337FF81180D
            DF811D0931323334
            3536373839   --->(Session APP Data, length 61, value "TLV Data in HEX ASCII format")
```

The decoded application data (payload of DFA710, tag ID 5008) is as follows:

```
FF0D 3A (GET-DATA Request (Tag ID 13), length 58)
   FF28 26 (Correlation Information, length 38)
      DF64 02 J1 (Terminal ID, length 2, value "J1")
      DF68 08 00000138 (Transaction Sequence Number, length 8, value "00000138")
```


Highly Confidential: Property of Triton Systems – Do Not Disclose

```
   DF29 13 2015-08-14 08:59:37 (Timestamp, length 19, value "2015-08-14 08:59:37")
FF8118 0D (Cardless Transaction Request (Tag ID 152), length 13)
   DF811D 09 123456789 (Cash Code (Tag ID 157), length 9, value "123456789")
```

In this example, terminal "J1" is sending a Cardless Transaction Request with Session Key "HEY24", Sequence Number "312" (notice that it is sent in hex in the message), and Cash Code "123456789".

7.3.9.3 Cardless Transaction Response

Cardless messages exchanged between the terminal and cardless host will be TDL PDUs using session wrappers. The application data included in the session PDU (the payload of DFA710 tag ID 5008) for a Cardless Transaction Response will be a GET DATA Response. This response will contain the requested transaction amount.

Session PDU:

Tag	Value	
FFA708	Session PDU (tag ID 5000)	
	Tag	**Value**
	DFA709	Destination SAP (tag ID 5001) Integer (9 = Just.Cash Cardless)
	DFA70A	Session Message ID (tag ID 5002) Integer (8 = Single session message response)
	DFA70B	Source SAP (tag ID 5003) Integer (9 = Just.Cash Cardless)
	FFA70C	Session Credentials (tag ID 5004)
	DFA710	Application Data (tag ID 5008)

Table 66 Session PDU

GET-DATA Response:

Tag	Value	
FF0E	GET-DATA Response (tag ID 14)	
	Tag	Value
	FF28	Correlation Information (tag ID 40) (same format as it is in the request)
	FF8119	Cardless Transaction Response (tag ID 153)

Table 67 GET-DATA Response

Cardless Transaction Response:

Tag	Value	
FF8119	Cardless Transaction Response (tag ID 153)	
	Tag	Value
	DF70	Authorization/Response Code (tag ID 112) Integer
	DF6E	Transaction Amount (tag ID 110) Ascii Represented Double
	DF6C	Track Data (tag ID 108) String (R)
	FF1C	Message Error (tag ID 28) (optional)

Table 68 Cardless Transaction Response

Valid Transaction Response Codes:

Reversal Status	Description
0	Approved
1	Cash Code Not Found
2	Cash Code Already Used

Table 69 Transaction Response Codes

PRELIMINARY

Tag	Value	
FF1C	Message Error (tag ID 28)	
	Tag	Value
	DF1D	Error Code (tag ID 29) - Integer
	DF1E	Error Description (tag ID 30) – String (R)

Table 70 Message Error

7.3.9.4 Cardless Transaction Response Example

Messages to the cardless host will be sent using raw binary packing. For readability, in this document examples are shown using base64 packing.

T1000100 00B4/6cIgYLfpwkBOd+nCgE336cLATn/pwwJ34EcBUhFWTI036cQYv8OX/8oJt9kAkox32gIMDAwMDAwMkbfKRMyMDE2LTA0LTE5IDEwOjE4OjQ5/4EZMt9wATDfbgYxMDAuMDDfbCI0NzYxNzM5MDAxMDEwMDEwPTEwMTIyMDExMTQzODc4MDg5

Unpacked Payload: The unpacked payload here shows the TLV structure. The payload of Tag ID 5008 (DFA710) is the GET-DATA Response TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 8182 (Session Packet, ID: 5000, length 130)
   DFA709 01 9 (Destination SAP, ID: 5001, length 1, value "9")
   DFA70A 01 7 (Session Message ID, ID: 5002, length 1, value "7")
   DFA70B 01 9 (Source SAP, ID: 5003, length 1, value "9")
   FFA70C 09 (Session Credentials, ID: 5004, length 9)
      DF811C 05 HEY24 (Cardless Session Key, ID: 156, length 5, value "HEY24")
   DFA710 62 FF0E5FFF2826DF64
             024A31DF68083030
             303030303246DF29
             13323031362D3034
             2D31392031303A31
             383A3439FF811932
             DF700130DF6E0631
             30302E3030DF6C22
             3437363137333930
             3031303130303130
             3D31303132323031
             3131343338373830
             3839   --->(Session APP Data, ID: 5008, length 98, value "TLV Data in HEX ASCII
format")
```

The decoded application data (payload of DFA710, tag ID 5008) is as follows:

```
FF0E 5F (GET-DATA Response, ID: 14, length 95)
   FF28 26 (Correlation Information, ID: 40, length 38)
      DF64 02 J1 (Terminal ID, ID: 100, length 2, value "J1")
      DF68 08 0000002F (Transaction Sequence Number, ID: 104, length 8, value "0000002F")
      DF29 13 2016-04-19 10:18:49 (Correlation Timestamp, ID: 41, length 19, value "2016-04-19
10:18:49")
   FF8119 32 (Cardless Trans Response, ID: 153, length 50)
      DF70 01 0 (Authorization Code, ID: 112, length 1, value "0")
      DF6E 06 100.00 (Requested Amount, ID: 110, length 6, value "100.00")
      DF6C 22 4761739001010010=10122011143878089 (Track Data, ID: 108, length 34, value
"4761739001010010=10122011143878089")
```

In this example, the host is responding to terminal J1's Cardless Transaction Request with a Cardless Transaction Response with Session Key "HEY24", Sequence Number "47" (notice that it is sent in hex in the message), and Authorization/Response Code "0", and Transaction Amount "100.00".

PRELIMINARY

7.3.9.5 Session Key Request

Cardless messages exchanged between the terminal and cardless host will be TDL PDUs using session wrappers. The application data included in the session PDU (the payload of DFA710 tag ID 5008) for a Cardless Session Key Request will be a GET DATA Request. This is a request for the cardless session key.

Session PDU:

Tag	Value	
FFA708	Session PDU (tag ID 5000)	
	Tag	**Value**
	DFA709	Destination SAP (tag ID 5001) Integer (9 = Just.Cash Cardless)
	DFA70A	Session Message ID (tag ID 5002) Integer (7 = Single session message request)
	DFA70B	Source SAP (tag ID 5003) Integer (9 = Just.Cash Cardless)
	FFA70C	Session Credentials (tag ID 5004) (optional) (not included for Session Key Request)
	DFA710	Application Data (tag ID 5008)

Table 71 Session PDU

GET-DATA Request:

Tag	Value	
FF0D	GET-DATA Request (tag ID 13)	
	Tag	**Value**
	FF28	Correlation Information (tag ID 40)
	FF811A	Cardless Session Key Request (tag ID 154)

Table 72 GET-DATA Request

Cardless Session Key Request:

Tag	Value	
FF811A	Cardless Session Key Request (tag ID 154)	
	Tag	**Value**
	FFA858	Terminal Message Item (Store Message) (tag ID 5208)

Table 73 Cardless Session Key Request

Terminal Message Item:

Tag	Value	
FFA858	Terminal Message Item (tag ID 5208)	
	Tag	Value
	DFA859	Terminal Message Type (tag ID 5209), Integer (R) (see Table 23)
	DFA85A	Terminal Message (tag ID 5210) – String (R/W)

Table 74 Terminal Message Item

Terminal Message Type	Description
1	Store message
2	Welcome Message
3	Marketing Message
4	Exit Message

Table 75 Terminal Message Type

7.3.9.6 Session Key Request Example

Messages to the cardless host will be sent using raw binary packing. For readability, in this document examples are shown using base64 packing.

T10001000078/6clVd+nCQE536cKATffpwsBOd+nEEL/DT//KB7fZAJKMd9oAN8pEzIwMTUtMDYtMTEgMTM6NDY6MTf/gRoa/6hYFt+oWQEx36haDVN0b3JIIE1lc3NhZ2U=

Unpacked Payload: The unpacked payload here shows the TLV structure. The payload of Tag ID 5008 (DFA710) is the GET-DATA Request TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 55 (Session Packet, length 85)
       DFA709 01 9 (Destination SAP, length 1, value "9")
       DFA70A 01 7 (Session Message ID, length 1, value "7")
       DFA70B 01 9 (Source SAP, length 1, value "9")
       DFA710 42 FF0D3FFF281EDF64
                 024A31DF6800DF29
                 13323031352D3036
                 2D31312031333A34
                 363A3137FF811A1A
                 FFA85816DFA85901
                 31DFA85A0D53746F
                 7265204D65737361
                 6765   --->(Session APP Data, length 66, value "TLV Data in HEX ASCII format")
```

The decoded application data (payload of DFA710, tag ID 5008) is as follows:

```
FF0D 3F (GET DATA Request (Tag ID 13), length 63)
       FF28 1E (Correlation Information, length 30)
              DF64 02 J1 (Terminal ID, length 2, value "J1")
              DF68 00 (Transaction Sequence Number, length 0, value "")
              DF29 13 2015-06-11 13:46:17 (Timestamp, length 19, value "2015-06-11 13:46:17")
              FF811A 1A (Cardless Session Key Request (Tag ID 154), length 26)
                     FFA858 16 (Terminal Message Item (Tag ID 5208), length 22)
                            DFA859 01 1 (Msg Type (Tag ID 5209), length 1, value "1")
                            DFA85A 0D Store Message (Message (Tag ID 5210), length 13, value
"Store Message")
```

In this example, terminal "J1" is requesting to receive its Cardless Session Key with store details "Store Message".

7.3.9.7 Session Key Response

Cardless messages exchanged between the terminal and cardless host will be TDL PDUs using session wrappers. The application data included in the session PDU (the payload of DFA710 tag ID 5008) for a Cardless Session Key Response will be a GET DATA Response. This response will contain the requested cardless session key.

Session PDU:

Tag	Value	
FFA708	Session PDU (tag ID 5000)	
	Tag	**Value**
	DFA709	Destination SAP (tag ID 5001) Integer (9 = Just.Cash Cardless)
	DFA70A	Session Message ID (tag ID 5002) Integer (8 = Single session message response)
	DFA70B	Source SAP (tag ID 5003) Integer (9 = Just.Cash Cardless)
	FFA70C	Session Credentials (tag ID 5004) (optional) (not included for Session Key Response)
	DFA710	Application Data (tag ID 5008)

Table 76 Session PDU

GET-DATA Response:

Tag	Value	
FF0E	GET-DATA Response (tag ID 14)	
	Tag	**Value**
	FF28	Correlation Information (tag ID 40) (same format as it is in the request)
	FF811B	Cardless Session Key Response (tag ID 155)

Table 77 GET-DATA Response

Cardless Session Key Response:

Tag	Value	
FF811B	Cardless Session Key Response (tag ID 155)	
	Tag	**Value**
	DF811C	Cardless Session Key (tag ID 156) – String (R)
	FF1C	Message Error (tag ID 28) (optional)

Table 78 Cardless Session Key Response

7.3.9.8 Session Key Response Example

Messages to the cardless host will be sent using raw binary packing. For readability, in this document examples are shown using base64 packing.

T10001000060/6clRN+nCQE536cKATjfpwsBOd+nEDH/Di7/KB7fZAJKMd9oAN8pEzlwMTUtMDYtMTEgMTM6NDY6MTf/gRsJ34EcBUhFWTI0

Unpacked Payload: The unpacked payload here shows the TLV structure. The payload of Tag ID 5008 (DFA710) is the GET-DATA Response TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 44 (Session Packet, length 68)
   DFA709 01 9 (Destination SAP, length 1, value "9")
   DFA70A 01 8 (Session Message ID, length 1, value "8")
   DFA70B 01 9 (Source SAP, length 1, value "9")
   DFA710 31 FF0E2EFF281EDF64
             024A31DF6800DF29
             13323031352D3036
             2D31312031333A34
             363A3137FF811B09
             DF811C0548455932
             34   --->(Session APP Data, length 49, value "TLV Data in HEX ASCII format")
```

The decoded application data (payload of DFA710, tag ID 5008) is as follows:

```
FF0E 2E (GET-DATA Response (Tag ID 14), length 46)
   FF28 1E (Correlation Information, length 30)
      DF64 02 J1 (Terminal ID, length 2, value "J1")
      DF68 00 (Transaction Sequence Number, length 0, value "")
      DF29 13 2015-06-11 13:46:17 (Timestamp, length 19, value "2015-06-11 13:46:17")
   FF811B 09 (Cardless Session Key Response (Tag ID 155), length 9)
      DF811C 05 HEY24 (Cardless Session Key (Tag ID 156), length 5, value "HEY24")
```

In this example, the cardless host has responded to terminal "J1" with "HEY24" as the Cardless Session Key.

7.3.10 Digital Currency Purchase Transactions

This section describes the information definitions for supporting digital currency purchase transactions.

7.3.10.1 Digital Currency Purchase Request

Digital currency purchase messages exchanged between the terminal and Digital Currency host will be TDL PDUs using session wrappers. The application data included in the session PDU (the payload of DFA710 tag ID 5008) for a Digital Currency Purchase Request will be a SET DATA Request. The requested amount will be the cash amount of digital currency to be purchased.

Session PDU:

Tag	Value	
FFA708	Session PDU (tag ID 5000)	
	Tag	**Value**
	DFA709	Destination SAP (tag ID 5001) Integer (11 = Digital Currency Purchase)
	DFA70A	Session Message ID (tag ID 5002) Integer (7 = Single session message request)
	DFA70B	Source SAP (tag ID 5003) Integer (11 = Digital Currency Purchase)
	FFA70C	Session Credentials (tag ID 5004)
	DFA710	Application Data (tag ID 5008)

Table 79: Session PDU

Session Credentials:

Tag	Value	
FFA70C	Session Credentials (tag ID 5004)	
	Tag	Value
	DF811C	Session Key (tag ID 156) – String (R)

Table 80: Session Credentials

SET-DATA Request:

Tag	Value		
FF0F	SET-DATA Request (tag ID 15)		
	Tag	**Value**	
	FF28	Correlation Information (tag ID 40) – See Table 4	
	FFB043	Digital Currency Purchase Request (tag ID 6211)	
		Tag	**Value**
		DF6E	Transaction (Requested) Amount (tag ID 110) – Double
		DFB04C	Currency Type (tag ID 6220) – Integer
		DFB04D	Customer ID (tag ID 6221) - String
		DF8126	Customer Name (tag ID 166) – String (Optional)

Table 81: Digital Currency Purchase Request Encoded as SET-DATA Request

Correlation Information:

Tag	Value	
FF28	Correlation Information (tag ID 40)	
	Tag	Value
	DF64	Terminal Identifier (tag ID 100) String
	DF68	Transaction Sequence Number (tag ID 104) Hexadecimal Unsigned Integer.
	DF29	Correlation Timestamp (tag ID 41) Date-Time

Table 82: Correlation Information

Currency Types:

Code	Description
1	Bitcoin
2	Ethereum

Table 83: Session PDU

Highly Confidential: Property of Triton Systems – Do Not Disclose

7.3.10.2 Digital Currency Purchase Request Example

Messages to the digital currency host will be sent using raw binary packing. For readability, in this document examples are shown using base64 packing.

T100010000A8/6cled+nCQIxMd+nCgE336cLAjEx/6cMCd+BHAVlRVkyNN+nEFf/D1T/KCbfZAJKMd9oCDAwMDAwMERF3ykTMjAx
Ny0wOS0wOCAwOTo0NDo1Of+wQyffbgU0MC4wMN+wTAEx37BNCFNhbXBsZUlE34EmCkpvaG4gU21pdGg=

Unpacked Payload: The unpacked payload here shows the TLV structure. The payload of Tag ID 5008 (DFA710) is the SET-DATA Request TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 79 (Session Packet, ID: 5000, length 121)
   DFA709 02 11 (Destination SAP, ID: 5001, length 2, value "11")
   DFA70A 01 7 (Session Message ID, ID: 5002, length 1, value "7")
   DFA70B 02 11 (Source SAP, ID: 5003, length 2, value "11")
   FFA70C 09 (Session Credentials, ID: 5004, length 9)
      DF811C 05 HEY24 (Cardless Session Key, ID: 156, length 5, value "HEY24")
   DFA710 57 FF0F54FF2826DF64
             024A31DF68083030
             303030304445DF29
             13323031372D3039
             2D30382030393A34
             343A3539FFB04327
             DF6E0534302E3030
             DFB04C0131DFB04D
             0853616D706C6549
             44DF81260A4A6F68
             6E20536D697468   --->(Session APP Data, ID: 5008, length 87
```

The decoded application data (payload of DFA710, tag ID 5008) is as follows:

```
FF0F 54 (SET-DATA Request, ID: 15, length 84)
   FF28 26 (Correlation Information, ID: 40, length 38)
      DF64 02 J1 (Terminal ID, ID: 100, length 2, value "J1")
      DF68 08 000000DE (Transaction Sequence Number, ID: 104, length 8, value "000000DE")
      DF29 13 2017-09-08 09:44:59 (Time, ID: 41, length 19, value "2017-09-08 09:44:59")
   FFB043 27 (Digital Currency Purchase Request, Tag ID: 6211, length 39)
      DF6E 05 40.00 (Requested Amount, ID: 110, length 5, value "40.00")
      DFB04C 01 1 (Currency Type, Tag ID: 6220, length 1, value "1")
      DFB04D 08 SampleID (Customer ID, Tag ID: 6221, length 8, value "SampleID")
      DF8126 0A John Smith (Customer Name, Tag ID: 166, length 10, value "John Smith")
```

In this example, customer John Smith using terminal J1 is requesting $40.00 worth of bitcoin.

7.3.10.3 Digital Currency Purchase Response

Digital Currency purchase messages exchanged between the terminal and digital currency host will be TDL PDUs using session wrappers. The application data included in the session PDU (the payload of DFA710 tag ID 5008) for a Digital Currency Purchase Response will be a SET DATA Response.

Note that for an Ethereum purchase, the Private Key tag will contain a Pass Phrase that should be printed on the receipt. The wallet tag will contain information that should be printed as the second QR code on the bitcoin purchase receipt.

Session PDU:

PRELIMINARY

Tag	Value	
FFA708	Session PDU (tag ID 5000)	
	Tag	**Value**
	DFA709	Destination SAP (tag ID 5001) Integer (11 = Digital Currency Purchase)
	DFA70A	Session Message ID (tag ID 5002) Integer (8 = Single session message response)
	DFA70B	Source SAP (tag ID 5003) Integer (11 = Digital Currency Purchase)
	FFA70C	Session Credentials (tag ID 5004)
	DFA710	Application Data (tag ID 5008)

Table 84: Session PDU

SET-DATA Response:

Tag	Value	
FF10	SET-DATA Response (tag ID 16)	
	Tag	**Value**
	FF28	Correlation Information (tag ID 40) – See Table 4
	FF15	SET-DATA Response Code Construct (tag ID 21)
	DF14	Tag ID value (tag ID 20), Integer (Reference to request in this case 6211)
	DF16	SET-DATA Response Code (tag ID 22), Integer
	FFB044	Digital Currency Purchase Response (tag ID 6212)
	Tag	**Value**
	DFB045	Price (tag ID 6213) - Double
	DFB046	Quantity (tag ID 6214) - Double
	DFB047	Whole Unit Price (tag ID 6215) - Double
	DFB048	Public Key (tag ID 6216) - String
	DFB049	Private Key (tag ID 6217) - String
	DFB04F	Purchase Limit (tag ID 6223) - Double
	DFB04E	Wallet (tag ID 6222) – String (optional)
	FF1C	Message Error (tag ID 28) (optional)

Table 85: Digital Currency Purchase Response Encoded as SET-DATA Response

SET-DATA Response Code:

This response code enumeration is specific to the SET-DATA operation. It indicates whether or not the SET-DATA request was accepted and will be one of the following:

Response Code	Description
1	Success
2	Invalid tag or tags associated with the requested object
3	Access violation (read-write infraction)
4	Insufficient access privileges
5	Operation Failure

Table 86: SET-DATA Response Code Enumeration

7.3.10.4 Digital Currency Purchase Response Example

Messages to the digital currency host will be sent using raw binary packing. For readability, in this document examples are shown using base64 packing.

T10001000134/6cIgeDfpwkCMTHfpwoBON+nCwIxMf+nDAnfgRwFSEVZMjTfpxCBvf8Qgbn/KCbfZAJKMd9oCDAwMDAwMEUz3yk
TMjAxNy0wOS0wOCAxMDozMzowM/8VC98UBDYyMTHfFgEx/7BEft+wRQU0MC4wMN+wRggwLjAwOTEyMt+wRwc0Mzg1LjAw3
7BIlm10WUhQNWU1Y1hDUzJjOGJlalVTcjlOckh3WUdIRUNSQlnfsEk0eFZXOHVQUFV2TUZEN3BRWXVid3c1WUtBRVlEdG5kYn
RpSng3YkhSdXR0M2pkNHJnY3FLeA==

Unpacked Payload: The unpacked payload here shows the TLV structure. The payload of Tag ID 5008 (DFA710) is the SET-DATA Response TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 81E0 (Session Packet, ID: 5000, length 224)
   DFA709 02 11 (Destination SAP, ID: 5001, length 2, value "11")
   DFA70A 01 8 (Session Message ID, ID: 5002, length 1, value "8")
   DFA70B 02 11 (Source SAP, ID: 5003, length 2, value "11")
   FFA70C 09 (Session Credentials, ID: 5004, length 9)
      DF811C 05 HEY24 (Cardless Session Key, ID: 156, length 5, value "HEY24")
   DFA710 81BD FF1081B9FF2826DF
               64024A31DF680830
               30303030304533DF
               2913323031372D30
               392D30382031303A
               33333A3033FF150B
               DF140436323131DF
               160131FFB0447EDF
               B0450534302E3030
               DFB04608302E3030
               39313232DFB04707
               343338352E3030DF
               B048226D74594850
               3565356358435332
               633862486A555372
               394E724877594748
               4543524259DFB049
               3478565738755050
               55764D4644377051
               597562777735594B
               41455944746E6462
               74694A7837624852
               757474336A643472
               6763714B78   --->(Session APP Data, ID: 5008, length 189
```

The decoded application data (payload of DFA710, tag ID 5008) is as follows:

```
FF10 81B9 (SET-DATA Response, ID: 16, length 185)
   FF28 26 (Correlation Information, ID: 40, length 38)
      DF64 02 J1 (Terminal ID, ID: 100, length 2, value "J1")
      DF68 08 000000E3 (Transaction Sequence Number, ID: 104, length 8, value "000000E3")
      DF29 13 2017-09-08 10:33:03 (Time, ID: 41, length 19, value "2017-09-08 10:33:03")
   FF15 0B (SET-DATA Response Code, ID: 21, length 11)
      DF14 04 6211 (Reference Tag ID, ID: 20, length 4, value "6211")
      DF16 01 1 (SET-DATA Response Code Enumeration, ID: 22, length 1, value "1")
   FFB044 7E (Digital Currency Purchase Response, Tag ID: 6212, length 126)
      DFB045 05 40.00 (Price, Tag ID: 6213, length 5, value "40.00")
      DFB046 08 0.009122 (Quantity, Tag ID: 6214, length 8, value "0.009122")
      DFB047 07 4385.00 (Whole Unit Price, Tag ID: 6215, length 7, value "4385.00")
      DFB048 22 mtYHP5e5cXCS2c8bHjUSr9NrHwYGHECRBY (Public Key, Tag ID: 6216, length 34,
        value "mtYHP5e5cXCS2c8bHjUSr9NrHwYGHECRBY")
      DFB049 34 xVW8uPPUvMFD7pQYubww5YKAEYDtndbtiJx7bHRutt3jd4rgcqKx (Private Key, Tag    ID:
6217, length 52, value "xVW8uPPUvMFD7pQYubww5YKAEYDtndbtiJx7bHRutt3jd4rgcqKx")
```

In this example, the digital currency host has provided the Price, Quantity, Whole Unit Price, Public Key, and Private Key for a Digital Currency purchase of $40.00 worth of bitcoin.

7.3.10.5 Digital Currency Commitment Request

Digital currency commitment messages exchanged between the terminal and Digital Currency host will be TDL PDUs using session wrappers. The application data included in the session PDU (the payload of DFA710 tag ID 5008) for a Digital Currency Commitment Request will be a SET DATA Request.

Session PDU:

PRELIMINARY

FFA708	Session PDU (tag ID 5000)	
	Tag	**Value**
	DFA709	Destination SAP (tag ID 5001) Integer (11 = Digital Currency Purchase)
	DFA70A	Session Message ID (tag ID 5002) Integer (7 = Single session message request)
	DFA70B	Source SAP (tag ID 5003) Integer (11 = Digital Currency Purchase)
	FFA70C	Session Credentials (tag ID 5004)
	DFA710	Application Data (tag ID 5008)

Table 87: Session PDU

SET-DATA:

Tag	**Value**		
FF0F	SET-DATA Request (tag ID 15)		
	Tag	**Value**	
	FF28	Correlation Information (tag ID 40)	
	FFB04A	Digital Currency Commitment Request (tag TD 6218)	
		Tag	**Value**
		DF70	Authorization Code (tag ID 112) – Integer
		DFB048	Digital Currency Public Key (tag TD 6216) - String

Table 88: Digital Currency Commitment Request Encoded as SET-DATA Request

Authorization Code:

Code	Description
0	Customer Canceled Transaction
1	Customer Approved Digital Currency Purchase

Table 89: Authorization Code

7.3.10.6 Digital Currency Commitment Request Example

Messages to the digital currency host will be sent using raw binary packing. For readability, in this document examples are shown using base64 packing.

T1000100008C/6cIY9+nCQIxMd+nCgE336cLAjEx/6cMCd+BHAVlRVkyNN+nEEH/Dz7/KCbfZAJKMd9oCDAwMDAwMDUx3ykTMjA
xNy0wNC0yNyAwNzozNzo0MP+wShHfcAEx37BICVB1YmxpY0tleQ==

Unpacked Payload: The unpacked payload here shows the TLV structure. The payload of Tag ID 5008 (DFA710) is the SET-DATA Request TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 63 (Session Packet, ID: 5000, length 99)
   DFA709 02 11 (Destination SAP, ID: 5001, length 2, value "11")
   DFA70A 01 7 (Session Message ID, ID: 5002, length 1, value "7")
   DFA70B 02 11 (Source SAP, ID: 5003, length 2, value "11")
   FFA70C 09 (Session Credentials, ID: 5004, length 9)
      DF811C 05 HEY24 (Cardless Session Key, ID: 156, length 5, value "HEY24")
   DFA710 41 FF0F3EFF2826DF64
             024A31DF68083030
             303030303531DF29
             13323031372D3034
             2D32372030373A33
             373A3430FFB04A11
             DF700131DFB04809
             5075626C69634B65
             79   --->(Session APP Data, ID: 5008, length 65, value
```

The decoded application data (payload of DFA710, tag ID 5008) is as follows:

```
FF0F 3E (SET-DATA Request, ID: 15, length 62)
```

```
FF28 26 (Correlation Information, ID: 40, length 38)
   DF64 02 J1 (Terminal ID, ID: 100, length 2, value "J1")
   DF68 08 00000051 (Transaction Sequence Number, ID: 104, length 8, value "00000051")
   DF29 13 2017-04-27 07:37:40 (Time, ID: 41, length 19, value "2017-04-27 07:37:40")
FFB04A 11 (Digital Currency Commitment Request, ID: 6218, length 17)
   DF70 01 1 (Authorization Code, ID: 112, length 1, value "1")
   DFB048 09 PublicKey (Public Key, ID: 6216, length 9, value "PublicKey")
```

7.3.10.7 Digital Currency Commitment Response

Digital currency commitment messages exchanged between the terminal and digital currency host will be TDL PDUs using session wrappers. The application data included in the session PDU (the payload of DFA710 tag ID 5008) for a Digital Currency Purchase Commitment Response will be a SET DATA Response.

Session PDU:

Tag	Value	
FFA708	Session PDU (tag ID 5000)	
	Tag	**Value**
	DFA709	Destination SAP (tag ID 5001) Integer (11 = Digital Currency Purchase)
	DFA70A	Session Message ID (tag ID 5002) Integer (8 = Single session message response)
	DFA70B	Source SAP (tag ID 5003) Integer (11 = Digital Currency Purchase)
	FFA70C	Session Credentials (tag ID 5004)
	DFA710	Application Data (tag ID 5008)

Table 90: Session PDU

SET-DATA Response:

Tag	Value		
FF10	SET-DATA Response (tag ID 16)		
	Tag	**Value**	
	FF28	Correlation Information (tag ID 40)	
	FF15	SET-DATA Response Code Construct (tag ID 21)	
		DF14	Tag ID value (tag ID 20), Integer (Reference to request tag ID)
		DF16	SET-DATA Response Code (tag ID 22), Integer
	FFB04B	Digital Currency Commitment Response (tag TD 6219)	
		Tag	**Value**
		FF1C	Message Error (tag ID 28) (optional)

Table 91: Digital Currency Commitment Response Encoded as SET-DATA Response

7.3.10.8 Digital Currency Commitment Response Example

Messages to the digital currency host will be sent using raw binary packing. For readability, in this document examples are shown using base64 packing.

T10001000088/6clYN+nCQIxMd+nCgE436cLAjEx/6cMCd+BHAVlRVkyNN+nED7/EDv/KCbfZAJKMd9oCDAwMDAwMDUx3ykTMjAxNy0wNC0yNyAwOTozNzo1Nf8VC98UBDYyMTjfFgEx/7BLAA==

Unpacked Payload: The unpacked payload here shows the TLV structure. The payload of Tag ID 5008 (DFA710) is the SET-DATA Request TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 60 (Session Packet, ID: 5000, length 96)
   DFA709 02 11 (Destination SAP, ID: 5001, length 2, value "11")
   DFA70A 01 8 (Session Message ID, ID: 5002, length 1, value "8")
   DFA70B 02 11 (Source SAP, ID: 5003, length 2, value "11")
   FFA70C 09 (Session Credentials, ID: 5004, length 9)
      DF811C 05 HEY24 (Cardless Session Key, ID: 156, length 5, value "HEY24")
   DFA710 3E FF103BFF2826DF64
```

```
024A31DF68083030
303030303531DF29
13323031372D3034
2D32372030393A33
373A3535FF150BDF
140436323138DF16
0131FFB04B00    --->(Session APP Data, ID: 5008, length 62, value
```

The decoded application data (payload of DFA710, tag ID 5008) is as follows:

```
FF10 3B (SET-DATA Response, ID: 16, length 59)
   FF28 26 (Correlation Information, ID: 40, length 38)
      DF64 02 J1 (Terminal ID, ID: 100, length 2, value "J1")
      DF68 08 00000051 (Transaction Sequence Number, ID: 104, length 8, value "00000051")
      DF29 13 2017-04-27 09:37:55 (Time, ID: 41, length 19, value "2017-04-27 09:37:55")
   FF15 0B (SET-DATA Response Code, ID: 21, length 11)
      DF14 04 6218 (Reference Tag ID, ID: 20, length 4, value "6218")
      DF16 01 1 (SET-DATA Response Code Enumeration, ID: 22, length 1, value "1")
   FFB04B 00 (Digital Currency Commitment Response, ID: 6219, length 0)
```

8 Allocated Tag Enumerations

This table lists the Triton Proprietary tag enumerations to be assigned for primitive and constructed TLV data items exchanged in host/terminal communications. These are only to be used in TLV encodings with the tag class PRIVATE, and are assigned by Triton Systems LLC.

Tag Enumeration	Tag	TLV Type	Description
1		Constructed	Transaction Options
2		Primitive	Label String
3		Primitive	Boolean State
4		Constructed	OAR Data
5		Primitive	OAR Return Code
6		Primitive	OAR Screen Text
7		Primitive	OAR Button Label
8		Primitive	OAR Customer Selected Label
9		Constructed	Custom Screen Text
10		Primitive	Text String
11		Primitive	Host Withdrawal Surcharge Amount
12		Primitive	Host Balance Surcharge Amount
28	FF1C	Constructed	Message Error
29	DF1D	Primitive	Error Code
30	DF1E	Primitive	Error Description
31		Primitive	Donation Offer Capability Flag
32		Primitive	Donation Amount
Range 100 (Transaction TLVs)			
100	DF64	Primitive	Terminal Identifier
101	DF65	Primitive	Transaction Type Identifier
102	DF66	Primitive	Source Account Type
103	DF67	Primitive	Destination Account Type
104	DF68	Primitive	Transaction Sequence Number
105	FF69	Constructed	Track Data Group
106	FF6A	Constructed	Track Data Entry
107	DF6B	Primitive	Track Number
108	DF6C	Primitive	Track Data
109	DF6D	Primitive	Encrypted PIN Block
110	DF6E	Primitive	Transaction Amount
111	DF6F	Primitive	Transaction Date

PRELIMINARY

Tag Enumeration	Tag	TLV Type	Description
112	DF70	Primitive	Authorization Code
113	DF71	Primitive	Working Key Value
114	DF72	Primitive	Surcharge Amount
115	FF73	Constructed	ATM Transaction Record
116	DF74	Primitive	Account Number
117	DF75	Primitive	Authorization Number
118	DF76	Primitive	Ledger Balance
119	DF77	Primitive	Available Balance
120	DF78	Primitive	Reversal Status
121	DF79	Primitive	Reversal Reason
122	DF7A	Primitive	Receipt Message
123	FF7B	Constructed	Extended Transaction Record Data
124	DF7C	Primitive	Dispensed Amount
125	DF7D	Primitive	Dispense Status
126	FF7E	Constructed	Dispensed Count by Cassette
127	FF7F	Constructed	Rejected Count by Cassette
128	DF8100	Primitive	Cassette A
129	DF8101	Primitive	Cassette B
130	DF8102	Primitive	Cassette C
131	DF8103	Primitive	Cassette D
132	FF8104	Constructed	Notification Info
133	DF8105	Primitive	Message Type
134	DF8106	Primitive	Recipient Address
135	DF8107	Primitive	Unique Identifier
136	FF8108	Constructed	Disable SMS Command
137	FF8109	Constructed	Working Key
138	DF810A	Primitive	Working Key ID
139	DF810B	Primitive	Language Selected
140	FF810C	Constructed	Enable SMS Message
141	FF810D	Constructed	Enable SMS Response Message
142	DF810E	Primitive	Enable SMS
143	DF810F	Primitive	Opt In Flag
144	DF8110	Primitive	Business Date
145	FF8111	Constructed	ATM Report Record
146	FF8112	Constructed	Standard Transaction Request
147	FF8113	Constructed	Standard Transaction Response
148	DF8114	Primitive	EMV Data
149	DF8115	Primitive	Triton Specific EMV Data
150	FF8116	Constructed	Withdrawal Transaction Record
151	FF8117	Constructed	Transfer Transaction Record
152	FF8118	Constructed	Cardless Transaction Request
153	FF8119	Constructed	Cardless Transaction Response
154	FF811A	Constructed	Cardless Transaction Session Key Request
155	FF811B	Constructed	Cardless Transaction Session Key Response
156	FF811C	Primitive	Cardless Session Key
157	FF811D	Primitive	Cardless Pay Code
158	FF811E	Constructed	Cardless Reversal Request
159	FF811F	Constructed	Cardless Reversal Response
160	FF8120	Constructed	Transaction Fees
161	DF8121	Primitive	IVA (Impuesto al Valor Agregado) Tax for Withdrawal
162	DF8122	Primitive	Surcharge plus IVA tax for a withdrawal
163	DF8123	Primitive	Line Fee
164	DF8124	Primitive	Loyalty Fee
165	DF8125	Primitive	Bank Name

PRELIMINARY

Tag Enumeration	Tag	TLV Type	Description
166	DF8126	Primitive	Customer Name
167	DF8127	Primitive	EMV Fallback State Card Type
168	DF8128	Primitive	EMV Fallback State Chip Present
169	DF8129	Primitive	EMV Fallback State Reason
170	DF812A	Primitive	Card-less Session Key Password
171	DF812B	Primitive	TR-31 key block
172	DF812C	Primitive	TR-31 key block processing supported flag
500		Constructed	Cash Deposit Transaction Request
501		Constructed	Customer Info
502		Constructed	Deposit Transaction Data
503		Primitive	Total Deposit Amount Cash
504		Constructed	Note Counts
505		Constructed	Note Count
506		Primitive	Denomination
507		Primitive	Count
508		Constructed	Cash Deposit Transaction Response
509		Constructed	Deposit Totals Information
510		Constructed	Cash Deposit Totals Acknowledge
511		Constructed	Cash Deposit Totals
512		Primitive	Total Amount Deposited
513		Primitive	Total Transactions
514		Primitive	Clear Totals Flag
Range 1000 – 1999 (FX-DCC)			
1000		Constructed	Group of cassette configuration tags (ID 1001)
1001		Primitive	Dispenser mechanism ID
1002		Constructed	Cassette configuration information
1003		Primitive	Cassette ID
1005		Primitive	Cassette State
1006		Primitive	FX-DCC eligibility flag
1007		Constructed	FX-DCC offer lookup request PDU
1008		Primitive	Requested withdrawal amount
1009		Constructed	FX-DCC offer lookup response PDU
1010		Constructed	FX-DCC offer
1011		Primitive	FX-DCC offer type
1012	DF8774	Primitive	FX-DCC currency code for converted amount
1013	DF8775	Primitive	FX-DCC converted surcharge amount
1014	DF8776	Primitive	FX-DCC converted amount
1015		Primitive	FX-DCC commission
1016		Constructed	FX-DCC Fast Cash Configuration List
1024		Primitive	Current language in Use
1025		Primitive	Maximum Characters for Disclaimer Region
1026		Constructed	FX-DCC Eligibility Request
1027		Primitive	FX- DCC Primary Account Number
1028		Constructed	FX-DCC Eligibility Response
1029		Primitive	FX-DCC Surcharge Type
1030		Constructed	FX-DCC Disclaimer Scheme
1031		Constructed	FX-DCC Transaction Request
1032		Constructed	FX-DCC Transaction Response
1033		Primitive	FX-DCC Offer Acceptance Flag
1034		Primitive	FX-DCC Presented Amount
1035		Constructed	FX-DCC PIN Working Key Request
1036		Constructed	FX-DCC PIN Working Key Response
1037		Constructed	FX-DCC Close Request
1038		Constructed	FX-DCC Close Response

PRELIMINARY

Tag Enumeration	Tag	TLV Type	Description
1039		Primitive	FX-DCC Close Type
1040		Primitive	FX-DCC Transaction Count
1041		Primitive	FX-DCC Settlement Total
1042		Primitive	FX-DCC Close Reset Flag
1044		Primitive	FX-DCC Dispensed Amount
1045		Primitive	FX-DCC Markup
1046		Constructed	FX-DCC Reversal Request
1047		Constructed	FX-DCC Reversal Response
1048	DF8818	Primitive	FX-DCC Exchange Fee Amount
1049	DF8819	Primitive	FX-DCC Converted Exchange Fee Amount
1050		Primitive	FX-DCC Converted IVA Tax Amount (Withdrawal)
Range 2000 – N (VersaSafe Monitoring)			
2000			
2001			
2002	FF8F52	Constructed	Records Request
2003	FF8F53	Constructed	Records Message
2004			
2005	FF8F55	Constructed	Status Message
2006	FF8F56	Constructed	VersaSafe Users Request
2007	FF8F57	Constructed	VersaSafe Users Message
2008			
2009			
2010			
2011			
2012			
2013			
2014			
2015	FF8F5F	Constructed	Cash Deposit Transaction Record
2016	FF8F60	Constructed	Drop Deposit Transaction Record
2017	FF8F61	Constructed	Day Close Record
2018	FF8F62	Constructed	Pickup Close Record
2019	FF8F63	Constructed	Drop Close Record
2020	FF8F64	Constructed	Bill Acceptor Close Record
2021	FF8F65	Constructed	Shift Close Record
2022	FF8F66	Constructed	Device Status
2023	FF8F67	Constructed	VersaSafe User
2024	DF8F68	Primitive	Totals Type Id
2025	DF8F69	Primitive	Filter Id
2026	DF8F6A	Primitive	Number of Transactions
2027	DF8F6B	Primitive	Number of Cash Deposit Transactions
2028	DF8F6C	Primitive	Number of Drop Deposit Transactions
2029	DF8F6D	Primitive	Total Amount
2030	DF8F6E	Primitive	Amount Cash Deposited
2031	DF8F6F	Primitive	Amount Cash Dropped
2032	DF8F70	Primitive	Amount Checks Dropped
2033	DF8F71	Primitive	Number of Cash Drop Items
2034	DF8F72	Primitive	Number of Check Drop Items
2035	DF8F73	Primitive	Total Number of Bills Deposited
2036	FF8F74	Constructed	Bill Acceptor Bills Deposited
2037	DF8F75	Primitive	Device Id
2038	DF8F76	Primitive	User Id
2039	FF8F77	Constructed	Denomination Deposited
2040	DF8F78	Primitive	Denomination Value

PRELIMINARY

Tag Enumeration	Tag	TLV Type	Description
2041	DF8F79	Primitive	Record Type Id
2042	FF8F7A	Constructed	Filter Range
2043	FF8F7B	Constructed	Session Request
2044	FF8F7C	Constructed	Start Session
2045	DF8F7D	Primitive	Last X
2046	FF8F7E	Constructed	Date-Time Range
2047	DF8F7F	Primitive	Start Date-Time
2048	DF9000	Primitive	End Date-Time
2049	DF9001	Primitive	Date-Time Occurred
2050	FF9002	Constructed	All Unaudited
2051	DF9003	Primitive	Text
2052	DF9004	Primitive	Previous Value
2053	DF9005	Primitive	New Value
2054	FF9006	Constructed	End Session
2055	DF9007	Primitive	Password
2056	FF9008	Constructed	VersaSafe User
2057	DF9009	Primitive	User Name
2058	DF900A	Primitive	Transaction Id
2059			
2060	FF900C	Constructed	Continue Session
2061	DF900D	Primitive	Session Indicator
2062	DF900E	Primitive	Currency Code
2063	FF900F	Constructed	Versions Request
2064	FF9010	Constructed	Versions Message
2065	FF9011	Constructed	Version Information
2066	DF9012	Primitive	Version Label
2067	DF9013	Primitive	Version Number
2068	DF9014	Primitive	Record Id
Range 6000 – N (Prepaid/Gift Card Services)			
6000		Constructed	Prepaid/gift card activation
6001		Constructed	Prepaid/gift card activation list
6002		Constructed	Prepaid/gift card activation entry
6003		Primitive	Card ID
6004		Primitive	Amount
6005		Constructed	Prepaid/gift card deactivation
6006		Constructed	Prepaid/gift card deactivation list
6007		Constructed	Prepaid/gift card deactivation entry
6008		Primitive	Reason
6009		Constructed	Gift card operation failure information
6010		Primitive	Gift card failure entry
6011		Primitive	Gift card serial number
6012		Primitive	Gift card failure code
6013		Primitive	Gift card failure description
6014		Primitive	Requested document count
6015		Primitive	Dispensed document count
6016		Primitive	Activation status
6017		Constructed	Gift card PM information
6018		Primitive	Scan error count
6019		Primitive	Reject event count
Range 6200 – N (Digital Currency Purchase)			
6200		Primitive	CCC Anti-Money Laundering Unique Customer Identification Code
6201		Primitive	CCC Crypto currency code
6202		Primitive	CCC Settlement currency code
6203		Primitive	CCC Transaction type customer is eligible for

PRELIMINARY

Tag Enumeration	Tag	TLV Type	Description
6204		Primitive	CCC Balance information
6205		Primitive	CCC Settlement currency balance information
6206		Primitive	CCC Crypto currency balance information
6207		Primitive	CCC Balance
6208		Primitive	CCC Available balance
6209		Primitive	CCC 24 hour bid transaction limit
6210		Primitive	CCC 24 hour ask transaction limit
6211		Constructed	Just Cash Bitcoin Purchase Request
6212		Constructed	Just Cash Bitcoin Purchase Response
6213		Primitive	Just Cash Bitcoin Price
6214		Primitive	Just Cash Bitcoin Quantity
6215		Primitive	Just Cash Bitcoin Whole Unit Price
6216		Primitive	Just Cash Bitcoin Public Key
6217		Primitive	Just Cash Bitcoin Private Key
6218		Constructed	Just Cash Bitcoin Confirmation Request
6219		Constructed	Just Cash Bitcoin Confirmation Response
6220		Primitive	Just Cash Bitcoin Currency Type
6221		Primitive	Just Cash Bitcoin Customer ID
6222		Primitive	Just Cash Bitcoin Wallet
6223		Primitive	Just Cash Bitcoin Purchase Limit

Table 92 - Allocated Tag Enumerations

PRELIMINARY

9 TDL Examples

TDL is defined to allow the host more control over the screen flow and specific options displayed to the customer. Some of the features that can be implemented using TDL include:

- Customer Specific Surcharge
- Overdraft or other Bank Fees Disclosure
- Multiple User Accounts
- Allow/Disallow Transactions per Customer
- Direct Currency Conversion
- Terminal Status Monitoring
- Terminal Configuration
- Smart Safe Transaction Support
- Terminal Firmware Update

9.1 Customer Specific Surcharge (TDL Legacy)

The host processor can indicate a specific surcharge to apply to a customer based on the customer account number by sending the surcharge in the TIR (Transaction Information Request) message response as in the following example:

TIR from terminal (after customer card is read):

[FS]TERMINALID12345[FS]97[FS] [FS]1234567890=123456789O123456789[FS] [FS]
[FS]XXXXXXXXXXXXXXXX[FS][FS]qaDF0B**00**[FS]

TIR response from host (indicating withdrawal surcharge amount of $1.50):

[FS]TERMINALID12345[FS]97[FS] [FS]000[FS]00000000[FS]010110[FS]120000[FS] [FS]
[FS][FS]qaDF0B041.50[FS]

9.2 Overdraft Fee Disclosure (TDL Legacy)

The host processor can indicate whether or not to display an overdraft fee message to the customer to allow the customer the option to decline the transaction by sending an overdraft fee flag in the TIR message response as in the following example:

TIR from terminal (after customer enters $100.00 withdrawal amount):

[FS]TERMINALID12345[FS]97[FS] [FS]1234567890=123456789O123456789[FS]00010000[FS]
[FS]XXXXXXXXXXXXXXXX[FS][FS]qaFF011ADF020Doverdraft FeeDF03010[FS]

TIR response from host (indicating to display overdraft fee screen):

[FS]TERMINALID12345[FS]97[FS] [FS]000[FS]00000000[FS]010110[FS]120000[FS] [FS]
[FS][FS]qaFF011ADF020Doverdraft FeeDF03010[FS]

PRELIMINARY

9.3 Multiple User Accounts (TDL Legacy)

The host processor can indicate multiple accounts to display to the customer for the case where a user has more than one checking or savings accounts. This can be done by sending the appropriate OAR response to a TIR as in the following example:

TIR from terminal (after customer selects account type checking or savings):

[FS]TERMINALID12345[FS]97[FS] [FS]1234567890=12345678901234567890[FS] [FS]
[FS]XXXXXXXXXXXXXXXX[FS][FS]qaFF0408DF050200[FS]

TIR response from host (indicating to display screen with 2 account number selections and enable the cancel key):

[FS]TERMINALID12345[FS]97[FS] [FS]000[FS]00000000[FS]010110[FS]120000[FS] [FS]
[FS][FS]qaFF0456DF050201DF0616Select Desired
AccountDF070D#111111111111DF070D#222222222222DF0706Cancel [FS]

TIR from terminal (customer selects option 1)

[FS]TERMINALID12345[FS]97[FS] [FS]1234567890=12345678901234567890[FS] [FS]
[FS]XXXXXXXXXXXXXXXX[FS][FS]qaFF041BDF050203DF080D#111111111111[FS]

TIR response from host (indicating to no more OAR data):

[FS]TERMINALID12345[FS]97[FS] [FS]000[FS]00000000[FS]010110[FS]120000[FS] [FS]
[FS][FS]qaFF0408DF050202[FS]

9.4 Terminal Management Examples

9.5 Gift Card Services Examples

This section provides example activation request/response PDUs. The bytes in red are the TDL header including the static 'T', TDL version, packing and encoding identifier bytes. The bytes in blue are the payload length bytes. The remainder of the PDU is the Base64 payload. The unpacked payload shows each tag in the PDU on a separate line with the tag and length octets displayed in ASCII hex and a description to the right in parenthesis. In the case of the application data for tag 5008, the value is represented as ASCII hex characters, and they are not repeated in the description area inside the parenthesis due to size.

9.5.1 Gift Card Activation Request PDU

T1000100 00F0/6cIga3fpwkBNN+nCgE336cLATTfpxMgOUMxMEYxODdGRDFENERGODg4NTFCMkU2MDE5NUQwOUbfpxBm/w
9j/ygs32QIR0MxMDAtMDHfaAgwMDAwMDAwMd8pEzIwMTEtMTItMjAgMTE6MjI6NDf/mAw/65xLP+ucijfmMUQkFUTTA0MTFNVD
EwMDAwMDAxODXfmQFMjUuMDDfqG0DVVNE/6cMDN+nFAhDM0MwMjI5NQ==

Unpacked Payload: The unpacked payload here shows the TLV structure. Tag ID 5008 (DFA710) is the SET-DATA Request TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 FFA70881AD (Session Packet, ID: 5000, length 173)
   DFA709 DFA70901 4 (Destination SAP, ID: 5001, length 1, value "4")
   DFA70A DFA70A01 7 (Session Message ID, ID: 5002, length 1, value "7")
   DFA70B DFA70B01 4 (Source SAP, ID: 5003, length 1, value "4")
   DFA713 DFA71320 9C10F187FD1D4DF88851B2E60195D09F (Session Endpoint ID, ID: 5011, length 32,
value "9C10F187FD1D4DF88851B2E60195D09F")
   DFA710 DFA71066 FF0F63FF282CDF64
                   0847433130302D30
                   31DF680830303030
                   30303031DF291332
                   3031312D31322D32
                   302031313A32323A
                   3437FFAE7030FFAE
                   712CFFAE7228DFAE
                   73144241544D3034
                   31314D5431303030
                   303030313835DFAE
                   740532352E3030DF
                   A86D03555344   --->(Session APP Data, ID: 5008, length 102, value "TLV Data in
HEX ASCII format")
   FFA70C FFA70C0C (Session Credentials, ID: 5004, length 12)
      DFA714 DFA71408 C3C02295 (CBC MAC, ID: 5012, length 8, value "C3C02295")
```

The decoded application data (tag DFA710, ID 5008) is as follows:

```
FF0F 63 (SET-DATA Request, length 99)
   FF28 2C (Correlation Information, length 44)
      DF64 08 GC100-01 (Terminal ID, length 8, value "GC100-01")
      DF68 08 00000001 (Transaction Sequence Number, length 8, value "00000001")
      DF29 13 2011-12-20 11:22:47 (Correlation Timestamp, length 19, value
                                   "2011-12-20 11:22:47")
   FFAE70 30 (Activation Request, length 48)
      FFAE71 2C (Activation List, length 44)
         FFAE72 28 (Activation Entry, length 40)
            DFAE73 14 BATM0411MT1000000185 (Gift Card Serial Number, length 20, value
                                            "BATM0411MT1000000185")
            DFAE74 05 25.00 (Gift Card Amount, length 5, value "25.00")
            DFA86D 03 USD (Alpha Currency Code, length 3, value "USD")
```

In this example the target of the operation was tag ID 6000 (gift card activation) for a single prepaid card valued at $25.00 in US currency.

9.5.2 Gift Card Activation Response PDU Success Case

T1000100 00BC/6cIgYffpwkBNN+nCgE436cLATTfpxMgOUMxMEYxODdGRDFENERGODg4NTFCMkU2MDE5NUQwOUbfpxBA/xA9/ygs32QIR0MxMDAtMDHfaAgwMDAwMDAwMd8pEzIwMTEtMTItMjAgMTE6MjI6NTD/FQvfFAQ2MDAw3xYBMf+nDAzfpxQIQjZFNzc4NEI=

Unpacked Payload: The unpacked payload here shows the TLV structure. Tag ID 5008 (DFA710) is the SET-DATA Response TLV from the host application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

When unpacked from the Base64 form, the payload looks like the following:

```
FFA708 8187 (Session Packet, length 135)
   DFA709 01 4 (Destination SAP, length 1, value "4")
   DFA70A 01 8 (Session Message ID, length 1, value "8")
   DFA70B 01 4 (Source SAP, length 1, value "4")
   DFA713 20 9C10F187FD1D4DF88851B2E60195D09F (Session Endpoint ID, length 32, value
                                               "9C10F187FD1D4DF88851B2E60195D09F")
   DFA710 40 FF103DFF282CDF64
```


Highly Confidential: Property of Triton Systems – Do Not Disclose

```
            0847433130302D30
            31DF680830303030
            30303031DF291332
            3031312D31322D32
            302031313A32323A
            3530FF150BDF1404
            36303030DF160131   --->(Session APP Data, length 64, value
                                    "TLV Data in HEX ASCII format")
   FFA70C 0C (Session Credentials, length 12)
      DFA714 08 B7EA8AA9 (CBC MAC, length 8, value "B7EA8AA9")
```

When gift card transactions are conducted over an SSL socket, the credentials tag in the response from the host is optional. If the socket is not SSL, the credentials tag must be present and evaluate to a correct MAC value given the provisioned MAC key. The decoded application TLV is as follows:

```
FF10 3D (SET-DATA Response, length 61)
   FF28 2C (Correlation Information, length 44)
      DF64 08 GC100-01 (Terminal ID, length 8, value "GC100-01")
      DF68 08 00000001 (Transaction Sequence Number, length 8, value "00000001")
      DF29 13 2011-12-20 11:22:50 (Correlation Timestamp, length 19, value
                                   "2011-12-20 11:22:50")
   FF15 0B (Response Code, length 11)
      DF14 04 6000 (Tag ID, length 4, value "6000")
      DF16 01 1 (Response Code Enumeration, length 1, value "1")
```

In this example the target of the operation was tag ID 6000 (gift card activation) and the response code enumeration indicates success.

9.5.3 Gift Card Activation Response PDU Failure Case

T1000100 00EC/6clgazfpwkBNN+nCgE436cLATTfpxMgOUMxMEYxODdGRDFENERGODg4NTFCMkU2MDE5NUQwOUbfpxBl/xBi/ygs32QIR0MxMDAtMDHfaAgwMDAwMDAwMd8pEzIwMTEtMTItMjAgMTE6MjI6NTD/FQvfFAQ2MDAw3xYBNf+ueSH/mod3657FEJBVE0wNDExTVQxMDAwMDAwMTg13658ATH/pwwM36cUCDRDNUEyRTJF

Unpacked Payload: The unpacked payload here shows the TLV structure. Tag ID 5008 (DFA710) is the SET-DATA Response TLV from the host application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 81AC (Session Packet, length 172)
   DFA709 01 4 (Destination SAP, length 1, value "4")
   DFA70A 01 8 (Session Message ID, length 1, value "8")
   DFA70B 01 4 (Source SAP, length 1, value "4")
   DFA713 20 9C10F187FD1D4DF88851B2E60195D09F (Session Endpoint ID, length 32, value
                                               "9C10F187FD1D4DF88851B2E60195D09F")
   DFA710 65 FF1062FF282CDF64
             0847433130302D30
             31DF680830303030
             30303031DF291332
             3031312D31322D32
             302031313A32323A
             3530FF150BDF1404
             36303030DF160135
             FFAE7921FFAE7A1D
             DFAE7B144241544D
             303431314D543130
             3030303030313835
             DFAE7C0131   --->(Session APP Data, length 101, value
                              "TLV Data in HEX ASCII format")
   FFA70C 0C (Session Credentials, length 12)
      DFA714 08 4C5A2E2E (CBC MAC, length 8, value "4C5A2E2E")
```

PRELIMINARY

The application data tag (SET-DATA Response TLV) includes a feature/data specific failure tag, in this case tag ID 6009 – Gift Card Operation Failure Information. Tag ID 6013 Gift Card Failure Description is optional and in this case, not supplied. When the application TLV is decode by the application layer, it produces the following TLV:

```
FF10 62 (SET-DATA Response, length 98)
   FF28 2C (Correlation Information, length 44)
      DF64 08 GC100-01 (Terminal ID, length 8, value "GC100-01")
      DF68 08 00000001 (Transaction Sequence Number, length 8, value "00000001")
      DF29 13 2011-12-20 11:22:50 (Correlation Timestamp, length 19, value
                                   "2011-12-20 11:22:50")
   FF15 0B (Response Code, length 11)
      DF14 04 6000 (Tag ID, length 4, value "6000")
      DF16 01 5 (Response Code Enumeration, length 1, value "5")
   FFAE79 21 (Gift Card Failure Operation Information, length 33)
      FFAE7A 1D (Gift Card Failure Entry, length 29)
         DFAE7B 14 BATM0411MT1000000185 (Gift Card Serial Number, length 20, value
                                         "BATM0411MT1000000185")
         DFAE7C 01 1 (Gift Card Failure Code, length 1, value "1")
```

In this example the target of the operation was tag ID 6000 (gift card activation) and the gift card failure code is 1, meaning invalid gift card serial number.

PRELIMINARY

9.5.4 Gift Card Deactivation Request PDU

T10001000114/6cIgcnfpwkBNN+nCgE336cLATTfpxMgOUMxMEYxODdGRDFENERGODg4NTFCMkU2MDE5NUQwOUbfpxCBgf8
Pfv8oLN9kCEdDMTAwLTAx32gIMDAwMDAwMDHfKRMyMDExLTEyLTA5IDA4OjQyOjM3/651S/+udkf/rndD365zFEJBVE0wNDExT
VQxMDAwMDAwMTg13654J1ByZXNlbnRlZCB0aGVuIHJldHJhY3RlZCBkdWUgdG8gdGltZW91dP+nDAzfpxQIQ0JGMzQzNzM=

Unpacked Payload: The unpacked payload here shows the TLV structure. Tag ID 5008 (DFA710) is the SET-DATA Request TLV from the terminal application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 81C9 (Session Packet, length 201)
   DFA709 01 4 (Destination SAP, length 1, value "4")
   DFA70A 01 7 (Session Message ID, length 1, value "7")
   DFA70B 01 4 (Source SAP, length 1, value "4")
   DFA713 20 9C10F187FD1D4DF88851B2E60195D09F (Session Endpoint ID, length 32, value
                                              "9C10F187FD1D4DF88851B2E60195D09F")
   DFA710 8181 FF0F7EFF282CDF64
               0847433130302D30
               31DF680830303030
               30303031DF291332
               3031312D31322D30
               392030383A34323A
               3337FFAE754BFFAE
               7647FFAE7743DFAE
               73144241544D3034
               31314D5431303030
               303030313835DFAE
               782750726573656E
               746564207468656E
               2072657472616374
               6564206475652074
               6F2074696D656F75
               74   --->(Session APP Data, length 129, value
                         "TLV Data in HEX ASCII format")
   FFA70C 0C (Session Credentials, length 12)
```

The decoded application data (tag 5008) is as follows:

```
FF0F 7E (SET-DATA Request, length 126)
   FF28 2C (Correlation Information, length 44)
      DF64 08 GC100-01 (Terminal ID, length 8, value "GC100-01")
      DF68 08 00000001 (Transaction Sequence Number, length 8, value "00000001")
      DF29 13 2011-12-09 08:42:37 (Correlation Timestamp, length 19, value
                                   "2011-12-09 08:42:37")
   FFAE75 4B (Deactivation Request, length 75)
      FFAE76 47 (Deactivation List, length 71)
         FFAE77 43 (Deactivation Entry, length 67)
            DFAE73 14 BATM0411MT1000000185 (Gift Card Serial Number, length 20, value
                                            "BATM0411MT1000000185")
            DFAE78 27 Presented then retracted due to timeout (Deactivation Reason,
                                 length 39, value "Presented then retracted due to timeout")
```

In this example the terminal is attempting to deactivate a single gift card due to a timeout and retraction.

9.5.5 Gift Card Deactivation Response PDU Success Case

T10001000BC/6cIgYffpwkBNN+nCgE436cLATTfpxMgOUMxMEYxODdGRDFENERGODg4NTFCMkU2MDE5NUQwOUbfpxBA/xA
9/ygs32QIR0MxMDAtMDHfaAgwMDAwMDAwMd8pEzIwMTEtMTltMDkgMDg6NDI6NDD/FQvfFAQ2MDA13xYBMf+nDAzfpxQIOTM
0MDMxOUU=

Unpacked Payload: The unpacked payload here shows the TLV structure. Tag ID 5008 (DFA710) is the SET-DATA Response TLV from the host application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 8187 (Session Packet, length 135)
   DFA709 01 4 (Destination SAP, length 1, value "4")
   DFA70A 01 8 (Session Message ID, length 1, value "8")
   DFA70B 01 4 (Source SAP, length 1, value "4")
   DFA713 20 9C10F187FD1D4DF88851B2E60195D09F (Session Endpoint ID, length 32, value
                                               "9C10F187FD1D4DF88851B2E60195D09F")
   DFA710 40 FF103DFF282CDF64
             0847433130302D30
             31DF680830303030
             30303031DF291332
             3031312D31322D30
             392030383A34323A
             3430FF150BDF1404
             36303035DF160131   --->(Session APP Data, length 64, value
                                     "TLV Data in HEX ASCII format")
   FFA70C 0C (Session Credentials, length 12)
      DFA714 08 36AF9081 (CBC MAC, length 8, value "36AF9081")
```

The decoded application data (tag 5008) is as follows:

```
FF10 3D (SET-DATA Response, length 61)
   FF28 2C (Correlation Information, length 44)
      DF64 08 GC100-01 (Terminal ID, length 8, value "GC100-01")
      DF68 08 00000001 (Transaction Sequence Number, length 8, value "00000001")
      DF29 13 2011-12-09 08:42:40 (Correlation Timestamp, length 19, value
                                   "2011-12-09 08:42:40")
   FF15 0B (Response Code, length 11)
      DF14 04 6005 (Tag ID, length 4, value "6005")
      DF16 01 1 (Response Code Enumeration, length 1, value "1")
```

The deactivation response indicates the attempt to deactivate one or more prepaid/gift cards was successful.

9.5.6 Gift Card Deactivation Response PDU Failure Case

T1000100 00EC/6cIgazfpwkBNN+nCgE436cLATTfpxMgOUMxMEYxODdGRDFENERGODg4NTFCMkU2MDE5NUQwOUbfpxBl/xBi
/ygs32QIR0MxMDAtMDHfaAgwMDAwMDAwMd8pEzIwMTEtMTItMDkgMDg6NDI6NDL/FQvfFAQ2MDA13xYBNf+ueSH/mod3657F
EJBVE0wNDExTVQxMDAwMDAwMTg13658ATH/pwwM36cUCDdGQjM4REQz

Unpacked Payload: The unpacked payload here shows the TLV structure. Tag ID 5008 (DFA710) is the SET-DATA Response TLV from the host application (high-lighted in green). The tag and length octets are binary, so the entire application data TLV is represented in ASCII hex characters:

```
FFA708 81AC (Session Packet, length 172)
   DFA709 01 4 (Destination SAP, length 1, value "4")
   DFA70A 01 8 (Session Message ID, length 1, value "8")
   DFA70B 01 4 (Source SAP, length 1, value "4")
   DFA713 20 9C10F187FD1D4DF88851B2E60195D09F (Session Endpoint ID, length 32, value
                                               "9C10F187FD1D4DF88851B2E60195D09F")
   DFA710 65 FF1062FF282CDF64
             0847433130302D30
             31DF680830303030
             30303031DF291332
             3031312D31322D30
             392030383A34323A
             3432FF150BDF1404
             36303035DF160135
             FFAE7921FFAE7A1D
             DFAE7B144241544D
```

PRELIMINARY

```
            303431314D543130
            3030303030313835
            DFAE7C0131   --->(Session APP Data, length 101, value
                              "TLV Data in HEX ASCII format")
   FFA70C 0C (Session Credentials, length 12)
      DFA714 08 7FB38DD3 (CBC MAC, length 8, value "7FB38DD3")
```

The decoded application data (tag 5008) is as follows:

```
FF10 62 (SET-DATA Response, length 98)
   FF28 2C (Correlation Information, length 44)
      DF64 08 GC100-01 (Terminal ID, length 8, value "GC100-01")
      DF68 08 00000001 (Transaction Sequence Number, length 8, value "00000001")
      DF29 13 2011-12-09 08:42:42 (Correlation Timestamp, length 19, value
                                   "2011-12-09 08:42:42")
   FF15 0B (Response Code, length 11)
      DF14 04 6005 (Tag ID, length 4, value "6005")
      DF16 01 5 (Response Code Enumeration, length 1, value "5")
   FFAE79 21 (Gift Card Failure Operation Information, length 33)
      FFAE7A 1D (Gift Card Failure Entry, length 29)
         DFAE7B 14 BATM0411MT1000000185 (Gift Card Serial Number, length 20, value
                                         "BATM0411MT1000000185")
         DFAE7C 01 1 (Gift Card Failure Code, length 1, value "1")
```

Int his example the activation attempt for the gift card failed with an invalid serial number.

PRELIMINARY

10 LEGACY

10.1 Legacy Message Extensions

This version of TDL™ supports backwards compatibility with *Triton Standard Communications Definition*, version 5.41B. This section describes additions that are only supported within TDL™.

10.1.1 Extended Transaction Codes

When using message support within TDL™, the following extended transaction codes can be used:

Code	Description
97	Transaction Information Request (TIR). Indicates the transaction request is meant for obtaining information related to the customer transaction. The sequence number, Amount 1, and Amount 2 fields will be space filled in the transaction request. The host may space fill Amount 1 in the response or it can contain the account balance as before. Amount 2 will be ignored.

10.1.2 Extended FIDs

When using message support within TDL™, the following miscellaneous FIDs can be used:

Field ID (FID)	Length/Type	Description
qa	VAR	TDL Payload – see section 7.1. Use this FID to wrap TDL payload information within a legacy message request/response. The tag and length conventions from section 10.1.3 apply to data encoded in the 'qa' FID. The TDL header is not included.
qc	VAR	TDL Payload - The value for 'qc' FIDs are complete TDL messages, including the header, conforming to the encoding conventions as defined in section 7.1. Tag and length octets are binary unlike the 'qa' FID. If use with modem or similar communications is a concern, the implementation should specify a Packing Identifier (PI) utilizing Base64 payload packing.

10.1.3 Tag and Length Conventions For TDL Legacy Payloads

TDL TLV constructs for TDL legacy mode encode the tag and length octets as described in section 7.1 but are converted to ASCII hexadecimal characters. The purpose for this transformation is to preserve back compatibility with terminals that use modem communications configured as 7-bit ASCII. Therefore both tag and length octets will be transmitted as pairs of their ASCII equivalents in the ranges ['0'-'9] and ['A'-'F']. Two such ASCII hexadecimal characters are required to assemble each tag or length octet. Take the transaction option encoding for example:

"FF01**19**DF02**0C**Option LabelDF03**01**0"

PRELIMINARY

There is a single enclosing constructed tag "FF01" that includes the 2 sub-TLV tags "DF01" and "DF03". The lengths for the TLVs are "19", "0C", and "01" respectively (indicated in bold print). Each ASCII hexadecimal tag and length field should be converted to its binary equivalent, starting with the constructed tag "FF01". These ASCII characters should be converted to binary 0xFF and 0x01. The length is then converted from "19" to 0x19. The value of the enclosing constructed tag 0xFF01 spans both tag encodings for 0xDF02 and 0xDF03. The process involved in converting ASCII hexadecimal characters to binary tag and length octets can be expressed in pseudo-code as follows (example is the tag "FF01"):

Step 1. Read octet 1 'F' from line
Step 2. Convert to upper case
Step 3. Check ASCII value against ranges ['0'..'9'] and ['A'..'F']
Step 4. If byte 1 is in range
Subtract 0x30 from octet 1 (['0'..'9']) to get 0x00..0x09 OR
Subtract 0x41 from octet 1 (['A'..'F']) and add 0xA to get 0x0A..0x0F
Step 5. Read Byte 2 'F' from line
Step 6. Convert to upper case
Step 7. Check ASCII value against ranges ['0'..'9'] and ['A'..'F']
Step 8. If byte 2 is in range
Subtract 0x30 from byte 2 (['0'..'9']) to get 0x00..0x09 OR
Subtract 0x41 from byte 2 (['A'..'F']) and add 0xA to get 0x0A..0x0F
Step 9. Shift byte 1 0x0F left 4 bits and bitwise OR with byte 2 0x0F resulting in binary hex 0xFF
Step 10. Repeat steps 1 through 9 for the next 2 ASCII characters "01" and the accumulated 2 binary octets 0xFF and 0x01 for the tag 0xFF01

The same basic steps are repeated for all tag and length octet assembly. Furthermore, the cumulative length of enclosing constructed tags account for the length of the sub-TLVs in terms of their ASCII hexadecimal encoding, not the converted binary equivalent.

10.2 MODEM COMMUNICATION PARAMETERS

The modem communications parameters are:

- 300, 1200, or 2400 bps
- 7 data bits
- even parity
- 1 stop bit
- Asynchronous

The modem complies with Bell 212A/103 and CCITT V.22 standards.

The LRC (Longitudinal Redundancy Check) is always the last character of the message. This character is the Exclusive OR of all characters in the message, starting with the first character after the STX, through and including the ETX.

The maximum length of a typical transmitted message cannot exceed 2048 bytes including the STX, ETX, and the LRC.

For terminal software versions that support large message buffers the maximum length of a transmitted message cannot exceed 2000 bytes including the STX, ETX, and the LRC for Z180 based SW. The CE SW implementations are not subject to the 2000 byte limitation. For protocol problems "Terminal ACK Not Received By Host" and

PRELIMINARY

"Terminal Does Not Acknowledge Response From Host" the 3-second timeout counter should start **after** the LRC is transmitted from the MODEM.

†**Preliminary**.

PRELIMINARY

10.2.1 Control Character Table

STX	02h
ETX	03h
EOT	04h
ENQ	05h
ACK	06h
NAK	15h
FS	1Ch

Table 93 – Control Character Table

PRELIMINARY

10.2.2 Transaction Request Message

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Communications Identifier	8 ASCII Char.	F		X*	A text string used to identify the 'owner' of the message to a communications service provider. Default string is "XXXXXX^^". ^ characters are spaces and are required. 'X' characters represent identification string for processor.
Terminal Identifier	2 ASCII Char.	F		X*	Lower case 't' and lower case 'd' or 's', indicating to host the type of terminal
Software Version No.	2 ASCII Char.	F		X*	Two digit Base 36 number representing the version of software installed on the terminal. Starts at '00'
Encryption Mode Flag	1 ASCII Char.	F		X*	0 = Single DES encryption of PIN block. 1 = Reserved. 2 = Triple DES encryption of PIN block.
Information Header	7 ASCII Char.	F		X*	Seven byte field, unused as of this release. Must be space filled.
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		Code indicating type of transaction being done. See Transaction Codes table.
FS	1 ASCII Char.	F	X		Field separator
Sequence Number	4 Numeric	F	X****		Consecutive number from 1-9999 set by terminal. Rolls over at 9999. Tags a transaction as a unique event. Each subsequent transaction is guaranteed to have different sequence number.
FS	1 ASCII Char.	F	X		Field separator
Track 2	37 Numeric	V	X		Complete track 2 data, excluding the Start Sentinel, End Sentinel, and LRC
FS	1 ASCII Char.	F	X		Field separator
Amount 1	8 Numeric	F	X		Withdraw or transfer amount. Zero filled to the left. Represents the amount in the smallest possible unit of currency. Note: will be space filled when ATM is configured for extended amounts. This amount is stored in Miscellaneous Field 'e' for this case.
FS	1 ASCII Char.	F	X		Field separator
Amount 2	8 Numeric	F	X		Surcharge amount as set by terminal. Zero filled to the left. Represents the amount in the smallest possible unit of currency. An ASCII minus sign (e.g. "-0000100") in the first position represents a credit.

PRELIMINARY

FS	1 ASCII Char.	F	X		Field separator
PIN Block	16 Alphanumeric	F	X		ANSI standard encrypted PIN block. (See 10.4)
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous 1	Variable Alphanumeric	V		X	Can contain user defined information.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous 2	Variable Alphanumeric	V		X	Can contain user defined information.
FS	1 ASCII Char.	F	X		Field separator
Status Monitoring Field	XX ASCII Chars.	F		X**	Contains status information about terminal. Refer to table for full field format later in this document.
FS	1 ASCII Char.	F		X**	Field separator
Miscellaneous X	Variable Alphanumeric	V		X***	Can contain user defined information.
FS	1 ASCII Char.	F		X***	Field separator
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 94 – Transaction Request Message

* These fields are optional only as a group. The terminal contains a management function to enable or disable the sending of this group of fields. If the field group is not enabled on the terminal, these fields will not be sent. If they are not sent, the corresponding fields in the response messages should not be returned.

** This field (Status Monitoring Field) is optional and can be enabled or disabled at the terminal level. If enabled, it is sent on *ALL* messages up to the host. No response is expected to the sending of this information. The first character of the Status Monitoring Field will always be a capital letter (A – Z).

*** Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

**** If 12 digit sequence number is being sent as a miscellaneous field, this field should be padded with 4 spaces.

PRELIMINARY

10.2.3 Transaction Response Message

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Information Header	12 ASCII Char.	F		X*	Not used as of this release, must be space filled
FS	1 ASCII Char.	F		X*	Field separator
Multi-Block Indicator	1 Numeric	F	X		0 = single block response 1= another block will follow this block. Last block must always be 0.
FS	1 ASCII Char.	F	X		Field separator
Terminal ID	15 Alphanumeric	F	X		Copied from request message.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		Copied from request message.
FS	1 ASCII Char.	F	X		Field separator
Sequence Number	4 Numeric	F	X		Copied from request message.
FS	1 ASCII Char.	F	X		Field separator
Response Code	3 Numeric	F	X		Result of request. Approved = 000. Non-0000 = declined. See Authorization Codes for list of decline codes.
FS	1 ASCII Char.	F	X		Field separator
Authorization Num.	8 Numeric	F	X		Authorization number generated by the processor for the transaction. Right justified, zero filled to the left if not 8 characters. (note: This field is alphanumeric for Quasicash type transaction responses, Zero(030h) or space(020h) filled to the left if less than 8 characters.)
FS	1 ASCII Char.	F	X		Field separator
Transaction Date	6 Numeric	F	X		Date of transaction in MMDDYY format. Set by processor.
FS	1 ASCII Char.	F	X		Field separator
Transaction Time	6 Numeric	F	X		Time of the transaction in HHMMSS format. Set by processor.
FS	1 ASCII Char.	F	X		Field separator
Business Date	6 Numeric	F	X		Business date of the transaction in MMDDYY format. Set by processor.
FS	1 ASCII Char.	F	X		Field separator
Amount 1	8 Numeric	F	X		**Balance amount**. Zero filled to the left, right justified. If space filled, no balance available (for W/D transactions) Represents the amount in the smallest possible unit of currency. **Amount to dispense**. Use for transaction codes:

PRELIMINARY

					"48" – Check Cashing Withdrawal. "49" – Receive Money.
FS	1 ASCII Char.	F	X		Field separator
Amount 2	8 Numeric	F	X		Amount of actual surcharge Zero filled to the left, right justified. Represents the amount in the smallest possible unit of currency. An ASCII minus sign (e.g. "-0000100") in the first position represents a credit.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous 1	Variable Alphanumeric	V		X	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous 2	Variable Alphanumeric	V		X	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS	1 ASCII Char.	F	X		Field separator.
Miscellaneous X**	Variable Alphanumeric	V		X**	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS**	1 ASCII Char.	F		X**	Field separator.
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 95 - Transaction Request Message -2

* These fields are optional only as a group. They should only be included in responses to requests that contain the corresponding optional fields. Refer to the request message table for further information.

If the FID 'p' is returned in either MISC. 1 or MISC. 2, the data will be printed on all approved and declined transactions.

** More than two miscellaneous fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

PRELIMINARY

10.2.4 Configuration Table Download Request Message

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Communications Identifier	8 ASCII Char.	F		X*	A text string used to identify the 'owner' of the message to a communications service provider. Default string is "XXXXXX^^". ^ characters are spaces and are required. 'X' characters represent identification string for processor.

PRELIMINARY

Terminal Identifier	2 ASCII Char.	F		X*	Lower case 't' and lower case 'd' or 's', indicating to host the type of terminal.
Software Version No.	2 ASCII Char.	F		X*	Two digit Base 36 number representing the version of software installed on the terminal. Starts at '00'
Encryption Mode Flag	1 ASCII Char.	F		X*	0 = Single DES encryption of PIN block. 1 = Reserved. 2 = Triple DES encryption of PIN block.
Information Header	7 ASCII Char.	F		X*	Seven byte field, unused as of this release. Must be space filled.
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		Code indicating type of transaction being done. Transaction code = 60 for Configuration Table Download.
FS	1 ASCII Char.	F	X		Field separator
Status Monitoring Field	XX ASCII Chars.	F		X**	Contains status information about terminal. Refer to table for full field format later in this document.
FS	1 ASCII Char.	F		X**	Field separator
Miscellaneous X	Variable Alphanumeric	V		X***	Can contain user defined information.
FS	1 ASCII Char.	F		X***	Field separator
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 96 – Configuration Table Download Request Message

* These fields are optional only as a group. The terminal contains a management function to enable or disable the sending of this group of fields. If the field group is not enabled on the terminal, these fields will not be sent. If they are not sent, the corresponding fields in the response messages should not be returned.

** This field (Status Monitoring Field) is optional and can be enabled or disabled at the terminal level. If enabled, it is sent on *ALL* messages up to the host. No response is expected to the sending of this information. The first character of the Status Monitoring Field will always be a capital letter (A – Z).

*** Multiple Miscellaneous fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

10.2.4.1 TR-31 Key Block Support

The main use for the configuration table download request has traditionally been requesting transport of working keys (PIN and MAC) from the host to the terminal encoded as legacy FIDs (standard format). In addition to working keys sent in the standard format, any terminal supporting TR-31 key block encoded format may indicate that support by sending a 'qc' FID with a TDL message payload. This payload would typically be a single primitive tag with ID 172 (TAG DF812C) which is a Boolean value indicating TRUE when TR-31 key block format is supported and FALSE when it is not supported.

Highly Confidential: Property of Triton Systems – Do Not Disclose

10.2.5 Configuration Table Download Response Message

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Information Header	12 ASCII Char.	F		X*	Not used as of this release, must be space filled
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		Code indicating type of transaction being done. Transaction code = 60 for Configuration Table Download.
FS	1 ASCII Char.	F	X		Field separator
Field ID Code	'~'	F	X		The '~' character indicates the field is the Pin Working Key 1. This field is required for standard (non TR-31) working keys. For TR-31, the '~' should NOT be sent.
Encrypted PIN Key	16 Alphanumeric	F	X		PIN Working Key 1. This key is encrypted by the PIN Master Key before being transmitted to the terminal.
FS	1 ASCII Char.	F	X		Field separator
Field ID Code	'!'	F		X	The '!' character indicates the field is the maximum allowed surcharge amount.
Surcharge amount	8 Numeric	F		X	The surcharge amount that the terminal will display to the customer and charge for the transaction(Note: actual surcharge may be less than this amount). Right justified, zero filled to the left. Represents the amount in the smallest possible unit of currency.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous X**	Variable Alphanumeric	V		X	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS	1 ASCII Char.	F		X	Field separator, needed only if Miscellaneous field X is present.
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 97 - Configuration Table Download Request Message

NOTE:

* These fields are optional only as a group. They should only be included in responses to requests that contain the corresponding optional fields. Refer to the request message table for father information.

** More than two miscellaneous fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

PRELIMINARY

Fields will be added to the end of this message as needed. Each field will have a Field ID Code associated with it. Fields do not have to be in any order, and are not required in each message. The terminal will use the ID code to process all fields sent.

10.2.5.1 TR-31 Key Block Support

The main use for the configuration table download response has traditionally been for the transport of working keys (PIN and MAC) from the host to the terminal encoded as legacy FIDs (standard format). In addition to working keys sent in the standard format, for any terminal supporting TR-31 key block encoded format support the host may send a 'qc' FID with a TDL message payload with one or more TR-31 key blocks encoded with as ID 171 TLVs (TAG DF812B).

Tag DF812B is a binary data value encoded as a primitive tag. The binary data contains the Key Block Header (KBH) and confidential data formatted according to ANSI X9 TR-32 specifications using the key block binding method using variants.

PRELIMINARY

10.2.6 Host Totals Download Request Message

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Communications Identifier	8 ASCII Char.	F		X*	A text string used to identify the 'owner' of the message to a communications service provider. Default string is "XXXXXX^^". ^ characters are spaces and are required. 'X' characters represent identification string for processor.
Terminal Identifier	2 ASCII Char.	F		X*	Lower case 't' and lower case 'd' or 's', indicating to host the type of terminal
Software Version No.	2 ASCII Char.	F		X*	Two digit Base 36 number representing the version of software installed on the terminal. Starts at '00'
Encryption Mode Flag	1 ASCII Char.	F		X*	0 = Single DES encryption of PIN block. 1 = Reserved. 2 = Triple DES encryption of PIN block.
Information Header	7 ASCII Char.	F		X*	Seven byte field, unused as of this release. Must be space filled.
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		Code indicating type of transaction being done. See Transaction Codes table. For Host totals, code is either 50 or 51.
FS	1 ASCII Char.	F	X		Field separator
Status Monitoring Field	XX ASCII Chars.	F		X**	Contains status information about terminal. Refer to table for full field format later in this document.
FS	1 ASCII Char.	F		X**	Field separator
No. of Withdrawals (Terminal)	4 Numeric	F		X***	Total number of withdrawals since the last request for totals. Zero filled to the left.
No. of Inquiries (Terminal)	4 Numeric	F		X***	Total number of inquiries since the last request for totals. Zero filled to the left.
No. of Transfers (Terminal)	4 Numeric	F		X***	Total number of transfers since the last request for totals. Zero filled to the left.
Settlement (Terminal)	12 Numeric	F		X***	Total amount of all withdrawals. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency. Note: will be space filled when ATM is configured for full extended amounts. This amount is stored in Miscellaneous Field 'ux' for this case
FS	1 ASCII Char.	F		X***	Field separator

PRELIMINARY

Miscellaneous X	Variable Alphanumeric	V		X****	Can contain user defined information.
FS	1 ASCII Char.	F		X****	Field separator
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 98 – Host Totals Download Request Message

* These fields are optional only as a group. The terminal contains a management function to enable or disable the sending of this group of fields. If the field group is not enabled on the terminal, these fields will not be sent. If they are not sent, the corresponding fields in the response messages should not be returned.

** This field (Status Monitoring Field) is optional and can be enabled or disabled at the terminal level. If enabled, it is sent on *ALL* messages up to the host. No response is expected to the sending of this information. The first character of the Status Monitoring Field will always be a capital letter (A – Z).

*** This field group (Terminal Totals Fields) is optional and can be enabled or disabled at the terminal level by a management function. If enabled, the Terminal Totals Field is sent with the Host Totals Download Request Message up to the host. The first character, and all characters, of the Terminal Totals Fields will always be numeric (0 – 9).

**** Multiple Miscellaneous Fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

PRELIMINARY

10.2.7 Host Totals Response Message

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Information Header	12 ASCII Char.	F		X*	Not used as of this release, must be space filled
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		Code indicating type of transaction being done. See Transaction Codes table.
FS	1 ASCII Char.	F	X		Field separator
Business Date	6 Numeric	F	X		Business Date of transactions in MMDDYY format.
FS	1 ASCII Char.	F	X		Field separator
No. of Withdrawals	4 Numeric	F	X		Total number of withdrawals since the last request for totals. Zero or space filled to the left.
No. of Inquiries	4 Numeric	F	X		Total number of inquiries since the last request for totals. Zero or space filled to the left.
No. of Transfers	4 Numeric	F	X		Total number of transfers since the last request for totals. Zero or space filled to the left.
Settlement	8 Numeric	F	X		Total amount of all withdrawals. Zero or space filled to left, right justified. Represents the amount in the smallest possible unit of currency.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous Field**	Variable alphanumeric	V		X**	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS	1 ASCII Char.	F		X**	Field separator
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 99 - Host Totals Download Request Message - 2

* These fields are optional only as a group. They should only be included in responses to requests that contain the corresponding optional fields. Refer to the request message table for further information.

** Only 'p', '^', and 'e' FIDS are supported in the in the host response. See Valid Misc. FID table.
Note: 'e' field, if used in this message, contains 12 digit Settlement amount.

PRELIMINARY

10.2.8 Reversal Request Message

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Communications Identifier	8 ASCII Char.	F		X*	A text string used to identify the 'owner' of the message to a communications service provider. Default string is "XXXXXX^^". ^ characters are spaces and are required. 'X' characters represent identification string for processor.
Terminal Identifier	2 ASCII Char.	F		X*	Lower case 't' and lower case 'd' or 's', indicating to host the type of terminal
Software Version No.	2 ASCII Char.	F		X*	Two digit Base 36 number representing the version of software installed on the terminal. Starts at '00'
Encryption Mode Flag	1 ASCII Char.	F		X*	0 = Single DES encryption of PIN block. 1 = Reserved. 2 = Triple DES encryption of PIN block.
Information Header	7 ASCII Char.	F		X*	Seven byte field, unused as of this release. Must be space filled.
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		"29" for reversal
FS	1 ASCII Char.	F	X		Field separator
Sequence Number	4 Numeric	F	X		Sequence number of the transaction to be reversed.
FS	1 ASCII Char.	F	X		Field separator
Track 2	37 Numeric	V	X		Complete track 2 data, excluding the Start Sentinel, End Sentinel, and LRC
FS	1 ASCII Char.	F	X		Field separator
Amount 1	8 Numeric	F	X		Requested Withdraw amount. Zero filled to the left. Represents the amount in the smallest possible unit of currency. Note: will be space filled when ATM is configured for full extended amounts. This amount is stored in Miscellaneous Field 'e' for this case.
FS	1 ASCII Char.	F	X		Field separator
Amount 2	8 Numeric	F	X		Surcharge amount as set by terminal. Zero filled to the left. Represents the amount in the smallest possible unit of currency.
FS	1 ASCII Char.	F	X		Field separator
Amount 3	8 Numeric	F	X		Actual Dispensed amount. Zero filled to the left. Represents the amount in the smallest possible unit of currency.

PRELIMINARY

					Note: will be space filled when ATM is configured for full extended amounts. This amount is stored in Miscellaneous Field 'ul' for this case.
FS	1 ASCII Char.	F	X		Field separator
Status Monitoring Field	XX ASCII Chars.	F	X	X**	Contains status information about terminal. Refer to table for full field format later in this document.
FS	1 ASCII Char.	F		X**	Field separator
Miscellaneous X**	Variable Alphanumeric	V		X***	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS**	1 ASCII Char.	F		X***	Field separator.
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 100 – Reversal Request Message

* These fields are optional only as a group. The terminal contains a management function to enable or disable the sending of this group of fields. If the field group is not enabled on the terminal, these fields will not be sent. If they are not sent, the corresponding fields in the response messages should not be returned.

** This field (Status Monitoring Field) is optional and can be enabled or disabled at the terminal level. If enabled, it is sent on *ALL* messages up to the host. No response is expected to the sending of this information. The first character of the Status Monitoring Field will always be a capital letter (A – Z).

*** Multiple Miscellaneous Fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

PRELIMINARY

10.2.9 Reversal Response Message

Field Name	Length/Type	Fixed / Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Information Header	12 ASCII Char.	F		X*	Not used as of this release, must be space filled
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Copied from request message.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		"29" from request message.
FS	1 ASCII Char.	F	X		Field separator
Sequence Number	4 Numeric	F	X		Copied from request message.
FS	1 ASCII Char.	F	X		Field separator
Response Code	3 Numeric	F	X		Result of request. Approved = 000. 111 = declined.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous X**	Variable Alphanumeric	V		X**	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS	1 ASCII Char.	F		X**	Field separator, needed only if Miscellaneous field X is present.
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 101 - Reversal Response Message

* These fields are optional only as a group. They should only be included in responses to requests that contain the corresponding optional fields. Refer to the request message table for father information.

** Multiple Miscellaneous Fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

PRELIMINARY

10.2.10 **RKT Request Message**

The RKT request message format is as follows:

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Communications Identifier	8 ASCII Char.	F		X*	A text string used to identify the 'owner' of the message to a communications service provider. Default string is "XXXXXX^^". ^ characters are spaces and are required. 'X' characters represent identification string for processor.
Terminal Identifier	2 ASCII Char.	F		X*	Lower case 't' and lower case 'd', indicating to host the type of terminal
Software Version No.	2 ASCII Char.	F		X*	Two digit Base 36 number representing the version of software installed on the terminal. Starts at '00'
Encryption Mode Flag	1 ASCII Char.	F		X*	0 = Single DES encryption of PIN block. 1 = Double DES encryption of PIN block.
Information Header	7 ASCII Char.	F		X*	Seven byte field, unused as of this release. Must be space filled.
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
RKT Message Identifier	2 Alphanumeric	F	X		Code to indicate that this is a RKT Request Message. This code is always '98'. This is used to avoid confusion with regular Transaction Request Messages.
FS	1 ASCII Char.	F	X		Field separator
Sequence Number	4 Numeric	F	X		Consecutive number from 1-9999 set by terminal. Rolls over at 9999. Tags a request as a unique event. Each subsequent request is guaranteed to have different sequence number.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous X	Variable Alphanumeric	V		X***	Can contain user defined information.
FS	1 ASCII Char.	F		X***	Field separator
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 102 - RKT Request Message

PRELIMINARY

* These fields are optional only as a group. They should only be included in responses to requests that contain the corresponding optional fields. Refer to the request message table for father information.

** Multiple Miscellaneous Fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

*** Multiple Miscellaneous Fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

10.2.11 **RKT Response message**

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Information Header	12 ASCII Char.	F		X*	Not used as of this release, must be space filled
FS	1 ASCII Char.	F		X*	Field separator
Multi-Block Indicator	1 Numeric	F	X		0 = single block response 1= another block will follow this block. Last block must always be 0.
FS	1 ASCII Char.	F	X		Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
RKT Message Identifier	2 Alphanumeric	F	X		Code to indicate that this is a RKT Response Message. This code is always '98'. This is used to avoid confusion with regular Transaction Response Messages.
FS	1 ASCII Char.	F	X		Field separator
Sequence Number	4 Numeric	F	X		Copied from request message.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous X	Variable Alphanumeric	V		X***	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS	1 ASCII Char.	F		X***	Field separator, needed only if Miscellaneous field X is present.
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 103 - RKT Response Message

* These fields are optional only as a group. They should only be included in responses to requests that contain the corresponding optional fields. Refer to the request message table for father information.

PRELIMINARY

*** Multiple Miscellaneous Fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

PRELIMINARY

10.2.12 Transaction Information Request

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Communications Identifier	8 ASCII Char.	F		X*	A text string used to identify the 'owner' of the message to a communications service provider. Default string is "XXXXXX^^". ^ characters are spaces and are required. 'X' characters represent identification string for processor.
Terminal Identifier	2 ASCII Char.	F		X*	Lower case 't' and lower case 'd' or 's', indicating to host the type of terminal
Software Version No.	2 ASCII Char.	F		X*	Two digit Base 36 number representing the version of software installed on the terminal. Starts at '00'
Encryption Mode Flag	1 ASCII Char.	F		X*	0 = Single DES encryption of PIN block. 1 = Reserved. 2 = Triple DES encryption of PIN block.
Information Header	7 ASCII Char.	F		X*	Seven byte field, unused as of this release. Must be space filled.
FS	1 ASCII Char.	F		X*	Field separator
Terminal ID	15 Alphanumeric	F	X		Terminal ID. Assigned by processor. Left justified, space filled to the right.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		Code indicating type of transaction being done. See Transaction Codes table.
FS	1 ASCII Char.	F	X		Field separator
Sequence Number	4 Numeric	F	X		This field is unused and space (020h) filled for the TIR.
FS	1 ASCII Char.	F	X		Field separator
Track 2	37 Numeric	V	X		Complete track 2 data, excluding the Start Sentinel, End Sentinel, and LRC
FS	1 ASCII Char.	F	X		Field separator
Amount 1	8 Numeric	F	X		This field is unused and space (020h) filled for the TIR.
FS	1 ASCII Char.	F	X		Field separator
Amount 2	8 Numeric	F	X		This field is unused and space (020h) filled for the TIR.
FS	1 ASCII Char.	F	X		Field separator
PIN Block	16 Alphanumeric	F	X		ANSI standard encrypted PIN block. (See Appendix A)
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous 1	Variable Alphanumeric	V		X	Can contain user defined information.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous 2	Variable Alphanumeric	V		X	Can contain user defined information.
FS	1 ASCII Char.	F	X		Field separator

PRELIMINARY

Miscellaneous X	Variable Alphanumeric	V		X**	Can contain user defined information.
FS	1 ASCII Char.	F		X**	Field separator
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 104 - Transaction Information Request Message

* These fields are optional only as a group. They should only be included in responses to requests that contain the corresponding optional fields. Refer to the request message table for father information.

** Multiple Miscellaneous Fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

10.2.13 Transaction Information Response

Field Name	Length/Type	Fixed/ Var.	Reqd	Opt.	Description
STX	1 ASCII Char.	F	X		Start of text character
Information Header	12 ASCII Char.	F		X*	Not used as of this release, must be space filled
FS	1 ASCII Char.	F		X*	Field separator
Multi-Block Indicator	1 Numeric	F	X		This field is unused and zero (030h) filled for TIR.
FS	1 ASCII Char.	F	X		Field separator
Terminal ID	15 Alphanumeric	F	X		Copied from request message.
FS	1 ASCII Char.	F	X		Field separator
Transaction Code	2 Alphanumeric	F	X		Copied from request message.
FS	1 ASCII Char.	F	X		Field separator
Sequence Number	4 Numeric	F	X		This field is unused and space (020h) filled for the TIR.
FS	1 ASCII Char.	F	X		Field separator
Response Code	3 Numeric	F	X		Result of request. Approved = 000. Non-0000 = declined. See Authorization Codes for list of decline codes.
FS	1 ASCII Char.	F	X		Field separator
Authorization Num.	8 Numeric	F	X		This field is unused and space (020h) filled for the TIR.
FS	1 ASCII Char.	F	X		Field separator
Transaction Date	6 Numeric	F	X		Date of transaction in MMDDYY format. Set by processor.
FS	1 ASCII Char.	F	X		Field separator
Transaction Time	6 Numeric	F	X		Time of the transaction in HHMMSS format. Set by processor.

PRELIMINARY

FS	1 ASCII Char.	F	X		Field separator
Business Date	6 Numeric	F	X		Business date of the transaction in MMDDYY format. Set by processor.
FS	1 ASCII Char.	F	X		Field separator
Amount 1	8 Numeric	F	X		This field is unused and blank space (020h) for the TIR.
FS	1 ASCII Char.	F	X		Field separator
Amount 2	8 Numeric	F	X		This field is unused and space (020h) filled for the TIR.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous 1	Variable Alphanumeric	V		X	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS	1 ASCII Char.	F	X		Field separator
Miscellaneous 2	Variable Alphanumeric	V		X	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS	1 ASCII Char.	F	X		Field separator.
Miscellaneous X**	Variable Alphanumeric	V		X**	Miscellaneous data. Refer to Miscellaneous Field ID codes for possible uses for this field.
FS**	1 ASCII Char.	F		X**	Field separator.
ETX	1 ASCII Char	F	X		End of text Character
LRC	1 Hexadecimal Char	F	X		Longitudinal Redundancy Check. Exclusive OR (XOR) of all characters starting with the first character after the STX, through and including the ETX.

Table 105 - Transaction Information Response

* These fields are optional only as a group. They should only be included in responses to requests that contain the corresponding optional fields. Refer to the request message table for father information.

** Multiple Miscellaneous Fields are allowed for. Each field must include a valid FID code and data and be terminated by a field separator. See Valid Misc. FID table.

10.2.14 **Status Monitoring Field**

Sub-Field Name	Length/Type	Description
Program Version Number	10 ASCII Chars.	Program version number in format XX-YYYY.ZZ, where XX indicates product, YYYY indicates processor version, and ZZ indicates revision level. For PC Based products the format is **XXMMNNNRRR, where XX = Product Code, MM = Major Version Number, NNN = Minor Version Number, RRR = Revision Version.** Refer to table for product identification.
Table Version Number	10 ASCII Chars.	Table version number in format XX-YYYY.ZZ, where XX indicates product, YYYY indicates processor version, and ZZ indicates revision level. For PC Based products the format is **XXYYYYZZZZ, where XX = Product Code, YYYY = Screen File ID (alpha-numeric), ZZZZ = Screen File Version.** Refer

PRELIMINARY

		to table for product identification.
Firmware Version Number	10 ASCII Chars.	EPROM version number in format WWXX.YYZAA, where WW is the family product code, XX is the major version number, YY is the minor version number, Z indicates presence of Electronic Journal (security module), and AA is revision of security module firmware For PC Based products the format is **WWXX.YYAAA, where WW = Product Code, XX = Main Board Major Version Number, YY = Main Board Minor Version Number, and AAA is revision of security module firmware.**
Alarm, Chest door open	1 ASCII Char.	0 – False, 1 – True, <Space> - Not supported, X – Not installed
Alarm, Top door open	1 ASCII Char.	0 – False, 1 – True, <Space> - Not supported, X – Not installed
Alarm, Supervisor active	1 ASCII Char.	0 – False, 1 – True, <Space> - Not supported
Receipt printer, paper status	1 ASCII Char.	0 – OK, 1 – Low, 2 – Out, <Space> - Not supported
Receipt printer, ribbon status	1 ASCII Char.	0 – OK, 1 – Low, 2 – Out, <Space> - Not supported, T – thermal
Journal printer, paper status	1 ASCII Char.	0 – OK, 1 – Low, 2 – Out, <Space> - Not supported, X – Not installed
Journal printer, ribbon status	1 ASCII Char.	0 – OK, 1 – Low, 2 – Out, <Space> - Not supported, T – thermal, X – Not installed
Note status, dispenser	1 ASCII Char.	0 – OK, 1 – Low, 2 – Out, 3 – Unknown, <Space> - Not supported (This is the cumulative summary of all cassettes in the dispenser. Empty individual cassettes are not reflected here). Note: The note status may not be determined in some error cases, this will be indicated by value 3.
Receipt printer	1 ASCII Char.	0 – OK, 1 – Needs attention, 2 – Out of service, <Space> - not supported, X – Not installed
Journal Printer	1 ASCII Char.	0 – OK, 1 – Needs attention, 2 – Out of service, <Space> - not supported, X – Not installed
Dispenser	1 ASCII Char.	0 – OK, 1 – Needs attention, 2 – Out of service, <Space> - not supported, X – Not installed
Communications System	1 ASCII Char.	0 – OK, 1 – Needs attention, 2 – Out of service
Card reader	1 ASCII Char.	0 – OK, 1 – Needs attention, 2 – Out of service
Cards retained	3 ASCII Char.	000-999, <Spaces> - Not supported
Electronics system	2 ASCII Chars.	XX represents status of electronics subsystem (Refer to table for listing of codes), <Spaces> - Not supported.
Current Error Code for Terminal	3 ASCII Chars.	XXX represents the value of the error code that existed in the terminal at the time the message was sent. '000' indicates no problems.
Communications failures	3 ASCII Chars.	Number of communications problems experienced since last reset of this value. Comms. Failure would be BUSY, NO CONNECT, etc.
Cassette A Denomination	3 ASCII Chars.	000 – 99K(M)(G). Values that are all numeric represent actual value of notes. Values suffixed with 'K', 'M', or 'G' are as follows: Range: 1000 to 99,999 — Min. 1K — Max. 99K Range: 100,000 to 9,999,999 — Min. 1M — Max. 99M Range: 10,000,000 to 999,999,999 — Min. 1G — Max. 99G
Cassette A notes loaded	4 ASCII Chars.	0000 – 9999. Numbers of notes in the cassette when it was installed. (Note: this values is manually entered).

PRELIMINARY

<table>
<tr><td>Cassette A notes dispensed</td><td>4 ASCII Chars.</td><td>0000 – 9999</td></tr>
<tr><td>Cassette A reject events</td><td>3 ASCII Chars.</td><td>000 – 999</td></tr>
<tr><td>Cassette B Denomination</td><td>3 ASCII Chars.</td><td>000 – 99K(M)(G). Values that are all numeric represent actual value of notes. Values suffixed with 'K', 'M', or 'G' are as follows:<table><tr><th>Range</th><th>Min.</th><th>Max.</th></tr><tr><td>1000 to 99,999</td><td>1K</td><td>99K</td></tr><tr><td>100,000 to 9,999,999</td><td>1M</td><td>99M</td></tr><tr><td>10,000,000 to 999,999,999</td><td>1G</td><td>99G</td></tr></table></td></tr>
<tr><td>Cassette B notes loaded</td><td>4 ASCII Chars.</td><td>0000 – 9999. Numbers of notes in the cassette when it was installed. (Note: this values is manually entered).</td></tr>
<tr><td>Cassette B notes dispensed</td><td>4 ASCII Chars.</td><td>0000 – 9999</td></tr>
<tr><td>Cassette B reject events</td><td>3 ASCII Chars.</td><td>000 – 999</td></tr>
<tr><td>Cassette C Denomination</td><td>3 ASCII Chars.</td><td>000 – 99K(M)(G). Values that are all numeric represent actual value of notes. Values suffixed with 'K', 'M', or 'G' are as follows:<table><tr><th>Range</th><th>Min.</th><th>Max.</th></tr><tr><td>1000 to 99,999</td><td>1K</td><td>99K</td></tr><tr><td>100,000 to 9,999,999</td><td>1M</td><td>99M</td></tr><tr><td>10,000,000 to 999,999,999</td><td>1G</td><td>99G</td></tr></table></td></tr>
<tr><td>Cassette C notes loaded</td><td>4 ASCII Chars.</td><td>0000 – 9999. Numbers of notes in the cassette when it was installed. (Note: this values is manually entered).</td></tr>
<tr><td>Cassette C notes dispensed</td><td>4 ASCII Chars.</td><td>0000 – 9999</td></tr>
<tr><td>Cassette C reject events</td><td>3 ASCII Chars.</td><td>000 – 999</td></tr>
<tr><td>Cassette D Denomination</td><td>3 ASCII Chars.</td><td>000 - 99K(M)(G). Values that are all numeric represent actual value of notes. Values suffixed with 'K', 'M', or 'G' are as follows:<table><tr><th>Range</th><th>Min.</th><th>Max.</th></tr><tr><td>1000 to 99,999</td><td>1K</td><td>99K</td></tr><tr><td>100,000 to 9,999,999</td><td>1M</td><td>99M</td></tr><tr><td>10,000,000 to 999,999,999</td><td>1G</td><td>99G</td></tr></table></td></tr>
<tr><td>Cassette D notes loaded</td><td>4 ASCII Chars.</td><td>0000 – 9999. Numbers of notes in the cassette when it was installed. (Note: this values is manually entered).</td></tr>
<tr><td>Cassette D notes dispensed</td><td>4 ASCII Chars.</td><td>0000 – 9999</td></tr>
<tr><td>Cassette D reject events</td><td>3 ASCII Chars.</td><td>000 – 999</td></tr>
<tr><td>Total notes purged</td><td>3 ASCII Chars.</td><td>000 - 999. Total notes exiting the dispenser in response to a purge command. Cassette origin is unknown for these notes.</td></tr>
</table>

This field is composed of individual subfields and is sent as a whole group, if enabled. Individual fields cannot be sent separately.

The first character of the Status Monitoring Field will always be a capital letter A(ASCII Char 41h) through Z(ASCII Char 5Ah).

Table 106 – Status Monitoring Field

PRELIMINARY

10.2.15 Product Code Table for Program / Table Version Numbers

CODE	DESCRIPTION
C'X'	Model 9500 family of single cassette dispensers. 'X' may be any letter.
D'X'	Model 9515 family of multiple cassette dispensers. 'X' may be any letter.
N'X'	Model 2000 Mako family. 'X' may be any letter.
P'X'	Model 9200 SuperScrip family. 'X' may be any letter.
S'X'	Model 9600 family of single cassette dispensers. 'X' may be any letter.
U'X'	Model 9640 family of single or multiple cassette dispensers. 'X' may be any letter.
T'X'	Model 9615 family of multiple cassette dispensers. 'X' may be any letter.
Z'X'	Model 9700 family of multiple cassette dispensers. 'X' may be any letter.
K'X'	Model 9100 family of single cassette dispensers. 'X' may be any letter.
W'X'	Model 9800 family of single or multiple cassette dispensers. 'X' may be any letter.
X'X'	Model 9600 family of single or multiple cassette dispensers. 'X' may be any letter.
R'X'	PC Based Products family of single or multiple cassette dispensers. 'X' may be any letter.

Table 107 – Product Code Table for Program/Table Version Numbers

10.2.16 Product Code Table for EPROM Version Numbers

CODE	DESCRIPTION
C'X'	Model 9500 family of single or multiple cassette dispensers. 'X' may be any letter.
N'X'	Model 2000 Mako and Model 9200 SuperScrip. 'X' may be any letter.
S'X'	Model 9600 family of single or multiple cassette dispensers. 'X' may be any letter.
Z'X'	Model 9700 family of single or multiple cassette dispensers. 'X' may be any letter.
K'X'	Model 9100 family of single cassette dispensers. 'X' may be any letter.
R'X'	PC Based Products family of single or multiple cassette dispensers. 'X' may be any letter.

Table 108 - Product Code Table for Program/Table Version Numbers

PRELIMINARY

10.2.17 Transaction Code Table

CODE	DESCRIPTION
01	Change PIN. Must send PIN Change PIN Block (Misc. field) along with this code. This transaction code is also used for PIN Management Services transactions for EMV. This includes both Unlock and Change. Unlock transactions will not send the PIN Change Block (Misc. field).
11	Cash Withdrawal from primary checking account
12	Cash Withdrawal from primary savings account
15	Cash Withdrawal from primary credit card account
21	Transfer from primary checking account to savings account
22	Transfer from primary savings account to checking account
25	Transfer from primary credit card account to primary checking account
29	Reverse Previous Withdrawal
31	Primary checking account balance inquiry
32	Primary Savings account balance inquiry
35	Primary Credit card balance inquiry
41	Non-Cash Withdrawal from primary checking account (POS type transaction)
42	Non-Cash Withdrawal from primary savings account (POS type transaction)
43†	Quasicash pinless purchase from primary credit account (POS type transaction)
44†	Non-Financial prepaid online transaction (POS type transaction)
45	Non-Cash Withdrawal from primary credit card account (POS type transaction)
46†	Send Money from Checking
47†	Send Money from Savings
48†	Check Cashing Withdrawal
49†	Receive Money
50	Get host totals (do not change business day or reset totals)
51	Get host totals (Change business date and reset totals)
52	Recharge existing long distance calling plan.
53	Purchase new long distance calling plan. Use with misc. field 'xr'.

PRELIMINARY

60	Download configuration table
61	Cash Withdrawal from primary checking account using Miscellaneous Field ID 'e' in request message for amount. Amount field 1 is set to spaces.
62	Cash Withdrawal from primary savings account using Miscellaneous Field ID 'e' in request message for amount. Amount field 1 is set to spaces.
65	Cash Withdrawal from primary credit card account using Miscellaneous Field ID 'e' in request message for amount. Amount field 1 is set to spaces.
71	Transfer from primary checking account to savings account using Miscellaneous Field ID 'e' in request message for amount. Amount field 1 is set to spaces.
72	Transfer from primary savings account to checking account using Miscellaneous Field ID 'e' in request message for amount. Amount field 1 is set to spaces.
75	Transfer from primary credit card account to checking account using Miscellaneous Field ID 'e' in request message for amount. Amount field 1 is set to spaces.
81	Primary checking account balance inquiry using Miscellaneous Field ID 'e' for balance. Amount 1 field in response is set to spaces.
82	Primary savings account balance inquiry using Miscellaneous Field ID 'e' for balance. Amount 1 field in response is set to spaces.
85	Primary credit card balance inquiry using Miscellaneous Field ID 'e' for balance. Amount 1 field in response is set to spaces.
91	Non-Cash Withdrawal from primary checking account (POS type transaction) using Miscellaneous Field ID 'e' in request message for amount. Amount field 1 is set to spaces.
92	Non-Cash Withdrawal from primary savings account (POS type transaction) using Miscellaneous Field ID 'e' in request message for amount. Amount field 1 is set to spaces.
95	Non-Cash Withdrawal from primary credit card account (POS type transaction) using Miscellaneous Field ID in request message 'e' for amount. Amount field 1 is set to spaces.
99†	Use Extended Transaction Code String – Miscellaneous FID 'l' (ASCII 6Ch).

Table 109 – Transaction Code Table

†**Preliminary**.

PRELIMINARY

10.2.18 **Authorization Codes Table**

CODE	DESCRIPTION
000	Transaction approved
001	Expired card
002	Unauthorized usage
003	PIN error
004	Invalid PIN
005	Bank unavailable
006	Card not supported
007	Insufficient funds
008	Ineligible transaction
009	Ineligible account
010	Number of daily withdrawals exceeded
011	Cannot process transaction
012	Amount too large
013	Account closed
014	PIN tries exceeded
015	Database problem
016	Withdrawal limit already reached
017	Invalid amount
018	External decline
019	System error
020	Contact card issuer
021	Routing lookup problem
022	Message edit error
023	Transaction not supported
024	Insufficient funds
025	Western Union sender data error

PRELIMINARY

CODE	DESCRIPTION
026	Western Union receiver data error
027	CRC error
028	Pre-pay transaction failed
029	Pre-pay transaction rejected
030	Invalid mobile phone number
031	Pre-pay account limit reached
032	Pre-pay system unavailable
033	Response would exceed message size limit
034	Necessary information missing to process transaction
035	Second Invalid PIN (second "invalid PIN" try in a row and one try is left before deactivation)
036	Fallback Not Allowed
037	Invalid Exchange Rate (for Currency Exchange transactions)
085	Request AAC from ICC (for Balance Inquiry and PIN Change chip transactions – see 10.10.1)
111	Reversal Declined
222	PIN Change Declined

PRELIMINARY

CODE	DESCRIPTION
223	PIN Unlock Declined

Table 110 – Authorization Codes Tables

10.2.19 Miscellaneous Field ID Descriptions

Field ID Code	DESCRIPTION
'!' (exclamation point)	Surcharge amount to be displayed to customer during Surcharge Notification screen for Transactions other than Balance Inquiry. 8 numeric ASCII characters, right justified and zero filled to the left. Represents the amount in the smallest possible unit of currency. Amount cannot be above "00032767". Example: [!00000100]
'/' (solidus)	MAC enable or disable field. 1 ASCII Character. An ASCII '0' disables MAC (disabled by default) and an ASCII '1' enables MAC. The MAC disable field '/0' is not allowed in Configuration Download Response Messages.
'^' (caret)	MAC (Message Authentication Code) Result. 9 ASCII characters calculated with the MAC key by encrypting the message starting from the STX and ending with the ETX, the LRC is ignored. The MAC is separated into two equal halves by an ASCII space. MAC result example: 'A562 34CD'. NOTE: Sending a MAC with 9 ASCII spaces will cause the terminal to fail the current message and perform a Configuration Table Download on the next call.
'a'	Currency Exchange information. Refer to supplemental document "TSCE Currency Exchange" for details on the format and use of this FID.
'b'	Supplemental balance field, 8 numeric ASCII characters long zero filled to left. Eleven space characters are used to pad the field to 20 positions. Example: [b00123456_ _ _ _ _ _ _ _ _ _ _] where '00123456' is 123456 units of available balance. Represents the amount in the smallest possible unit of currency.
'c'	Day close time is the time of day specified in hours and minutes when the scheduled day close will occur. The format is HHMM. The valid range is 0000(midnight) to 2359.
'd'	Message Download is used to download the programmable messages to the terminal. The first character after the FID is a character that defines which type of message is to be downloaded. The four message types currently supported are **s**tore, **m**arketing, **w**elcome and **e**xit. The first letter of each message type is used to identify it after the FID. The second character after the FID is the number of lines contained in the message. The text format is any ASCII character of 20 hex or greater. A maximum of 163 characters, including FID, message identifier, and line count, is allowed for the store and marketing messages. A maximum of 93 characters, including FID, message identifier, and line count, is allowed for welcome and exit messages. The following table lists the message types with associated REQUIRED line lengths.

PRELIMINARY

	Message Type	Message Identifier	Max. no. of Lines	Required number of characters per line
	Store	s	4	40
	Marketing	m	4	40
	Welcome	w	3	30
	Exit	e	3	30
'e'	Extended length amount field. When present in a Transaction Request Message, this field OVERRIDES the amount field 1 in the response message. Amount 1 will be space filled to indicate this. It can be used for balance inquiries. The format consists of the FID code 'e', followed by 12 ASCII digits, zero filled to the left. Example: [e000123456789] where '000123456789' represents 123,456,789 units of currency. (NOTE: the units always represent the smallest possible representation of the currency involved. For United States, this would be the number of pennies). When present in a Host Totals Response Message, this field represents the 12 digit Settlement Amount.			
'&'	28 Digit Account Number (i.e. Checking, Savings, etc.) Left justified, padded with spaces.			
'#'	12 Digit (alphanumeric) Sequence Number.			
'@'	PIN Change PIN Block. 16 alphanumeric. ANSI standard encrypted PIN block. (See 10.4) This is not include in EMV PIN Unlock transactions.			
'g'	Track 3 information, excluding the Start Sentinel, End Sentinel, and LRC. If this field is used, and if track 3 data is not present or is unreadable, this field will contain just the FID code 'g'.			
'h'	Heartbeat interval (minutes) is the length of time the terminal will sit idle (no transactions initiated) before it sends a Configuration Download Request. The format is 3 ASCII numeric characters, right justified and zero filled to the left. The valid range is 1 to 999.			
'i' ASCII Char – 69h.	Primary Account Number from track 3 on the magnetic stripe. If this field is used, and if track 3 data is not present or is unreadable, this field will contain just the FID code 'i'.			
'j'	Journal data – Ten ASCII numbers, optionally followed by size / journal data pairings, always ending with an ANSI CRC-16 checksum. - T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10} (T_{11} T_{12} T_{13})(Base64 data) T_{14} T_{15} T_{16} T_{17} 1 Numeric (T_1) - Status Field '0' = Disable '1' = Enable '2'=Incomplete record '3'=ACK '4'=NAK 2 Numeric (T_2 T_3) - Maximum records when sent from host or numbers of records to follow when sent to host. 3 Numeric (T_4 T_5 T_6) - Maximum journal data (256-999 bytes) when sent from host or total journal data (in bytes) to follow when sent to host. 4 Numeric (T_7 T_8 T_9 T_{10}) – Indicates the number of journal records that have not been audited on the terminal which will include any records included with this message. This field will be zero filled when being sent from the host to the terminal.			
	(T_{11} T_{12} T_{13})(Base64 data) - The remainder of the packet will be (T_2 T_3) pairs of 3 ASCII numbers followed by the Base64 representation of the journal record as defined in Triton Connect Journal Command Packet (*Triton Terminal Journal Record Format Specification.*). This part of the packet will only be transmitted from the terminal to the host. 3 Numeric (T_{11} T_{12} T_{13}) – size (in bytes) of 1 Base64 (RFC 3548) encoded journal record to follow. ASCII bytes- Base64 representation of the journal record as defined by *Triton Terminal Journal Record Format Specification.*			
	T_{14} T_{15} T_{16} T_{17} -4 ASCII characters that represent the CRC-16 two byte checksum calculated from and including the j fid to the last byte of Base64 data. Each character represents a 4-bit hexadecimal numeral (0-9, A-F). Uses the ANSI CRC-16 error checking protocol standard. See the sample code in 10.9.			

'k'	Issuer Fee/Credit to be displayed to customer during Surcharge Notification screen. Nine numeric ASCII characters, right justified and zero filled to the left. Represents the amount in the smallest possible unit of currency. An ASCII minus sign (e.g. "-00000100") in the first position represents a credit. The absence of this field implies an unknown Issuer Fee/Credit.
'l' † ASCII Char – 6Ch.	Extended Transaction Code String. – Six ASCII numbers. – T_1 T_2 T_3 T_4 T_5 T_6 2 Numeric (T_1 T_2) - Transaction Type Code 2 Numeric (T_3 T_4) - "From" Account Type Code (Code '00' not allowed) 2 Numeric (T_5 T_6) - "To" Account Type Code See 10.6 – Extended Transaction Code Table for a list of Transaction and Account Type Codes.
'm'	MAC (Message Authentication Code) key. 16 ASCII characters encrypted under Master key. Will be used starting with the *next* transaction.
	MAC (Message Authentication Code) Working Key Left Block for Triple-DES capable terminals. 16 ASCII characters encrypted under MAC Master key. Each character represents a 4-bit hexadecimal numeral (0-9, A-F). See 10.8 for the complete description of Encryption key sets. Will be used starting with the *next* transaction. The presence of only this FID in a Response Message (Transaction or Configuration Table Download) will enable Single-DES MAC Encryption.
'n'	Reason for reversal. Two ASCII numbers representing the following reasons for a reversal: '01' - Incorrect Dispense: The amount dispensed does not match amount requested. This may be due to either a mechanical failure or the lack of available bill(s) to make the requested amount. '02' – Protocol Error: A request message was sent from the terminal for which no valid response message was received from the host. '03' – Lost EOT: A valid response message was received by the terminal, but the EOT was not received. This assumes the terminal was configured to expect the EOT. If the terminal is configured to ignore EOT, there will be no reversal generated. '04' – Mac Error: The calculated MAC result does not match the MAC result returned in the response message. '05' - Customer Cancellation: Customer decided not to continue after reviewing the transaction fee. '06' – VAS Challenge/Response Error. An incorrect response was received in the 'vd' FID. '07' – ICC declined transaction. '08' – Offline PIN has not changed (the ATM could not communicate with the card) '09' – Offline PIN Management Request incomplete (no response from card after issuer script sent to card) '10' – Offline PIN Management failed (card sent message indicating the PIN Management Request Failed) '11' – Customer did not remove card from reader.

PRELIMINARY

<table>
<tr><td>'o' †
ASCII Char – 6Fh.</td><td colspan="2">Total number of CREDIT ADVANCE transactions and the total amount of all CREDIT ADVANCE transactions.

(12 Numeric total)

4 Numeric – Total number of CREDIT ADVANCE transactions since the last request for totals. Zero filled to the left.
8 Numeric – Total amount of all CREDIT ADVANCE transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.</td></tr>
<tr><td>'p'</td><td colspan="2">ASCII formatted printing text field. This field, if present, will be printed at the end of a successful transaction receipt, error receipt or host totals receipt. It must include as the second character (after the 'p') the number of lines to be printed (from '1' to '4'). Preformatted text follows this line count. All ASCII characters above 20 HEX (the space character) are allowed. The only control character allowed is the carriage return, 0d HEX. The printer is capable of printing up to forty characters per line. Each separate line of text must be terminated by a carriage return. No centering is done by the terminal; all text is printed as sent. Maximum length of this field is 166 characters, including the 'p'. Can also appear in host totals response message</td></tr>
<tr><td>'t'</td><td colspan="2">Current network time sent from the host processor to update the terminal's time. The format is 6 numeric ASCII characters in the form HHMMSS. The time will be in 24 hour military format.
May be used with the 'ua' FID to update both the time and date of the terminal. If the terminal software does not support the 'ua' FID, send the 't' FID alone to update only the terminal's time.
Both the time and date must reflect the local time zone of the terminal.</td></tr>
<tr><td rowspan="3">'u'
ASCII Char – 75h.</td><td colspan="2">Two character FIDs.
The first character after the FID 'u' is another character that defines which data element is to be transmitted. The table below describes each data element. A Field Separator must follow each field.</td></tr>
<tr><td>'a'</td><td>Current network date sent from the host processor to update the terminal's date. The format is 6 numeric ASCII characters in the form MMDDYY. The date format is month, day, year.
Must be used with the 't' FID to update both the time and date of the terminal.
Both the time and date must reflect the local time zone of the terminal.</td></tr>
<tr><td>'b'</td><td>16-bit CRC.
4 ASCII characters that represent the CRC-16 two byte checksum.
Each character represents a 4-bit hexadecimal numeral (0-9, A-F).
Uses the ANSI CRC-16 error checking protocol standard.

For a request message the 16-bit CRC calculation starts with the first character after the STX, and ends with the last character before the ETX, includes four ASCII zeros for the 'ub' field, and excludes the LRC. The result is the four ASCII characters that is placed in the 'ub' field in msb to lsb order. The host must also calculate the CRC by substituting the four ASCII character CRC result in the 'ub' field with four ASCII zeros, then compare the results. The converse should be performed for the response message.

The Triton TCP/IP protocol does not contain an STX, ETX, nor LRC, and includes a two-byte TCP/IP message length in the CRC Calculation. See "TDL0.52 - Part 1" for the complete description of the TCP message format.

See 10.9 for the complete ANSI 16-bit CRC description.</td></tr>
</table>

PRELIMINARY

	'd'	Encoded EMV data block ASCII characters. Variable length. See 10.10 for description of the formatting of this data field
	'e'	DATA Working Key Left Block. 16 ASCII characters. Each character represents a 4-bit hexadecimal numeral (0-9, A-F). See 10.8 for the complete description of Encryption key sets. Will be used starting with the *next* transaction.
	'f'	DATA Working Key Right Block. 16 ASCII characters. Each character represents a 4-bit hexadecimal numeral (0-9, A-F). See 10.8 for the complete description of Encryption key sets. Will be used starting with the *next* transaction.
	'g'	Host Transaction Trace Number 12 ASCII characters. Fixed length. Right justified, space filled to the left if less than 12 characters.
	'h'	Encoded EMV untagged-data block ASCII characters. Variable length. See 10.10 for description of the formatting of this data field.
	'i'	Motorized card reader commands 2 Numeric characters. Fixed length. This field is indented for use only on ATM's that have a motorized card reader. 01 – Cancel transaction and capture card. This command holds the customer's card inside a motorized card reader. 02 – Complete transaction and capture card. This command completes the transaction according to the authorization response and captures the card.
	'j'	Host Receipt – 2 ASCII numbers followed by a variable amount of ASCII data to be displayed exactly as formatted by the host. T_1 T_2 (T_3) 2 Numeric (T_1 T_2) – 2 ASCII numbers representing the character set selection. Currently '00' will always be sent indicating the default character set should be used. Variable Length ASCII (T_3) – Formatted ASCII text that will be printed exactly as it is sent from the host.
	'k'	Printer Capabilities- 10 ASCII numbers representing the current configuration of the terminal printer. T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10} 1 Numeric (T_1) – Represents the current printer mode '1' – Graphic '2'– Text. 3 Numeric (T_2 T_3 T_4) – This field is reserved for future use, but it will initially contain '000'. 2 Numeric (T_5 T_6) – The maximum number of characters per line. 2 Numeric (T_7 T_8) – The maximum number of lines per statement. 2 Numeric (T_9 T_{10}) – The default character set that is used by the system. '00' will be sent indicating the system default. Future enhancements may specify other character sets.
	'l' ASCII Char – 6Ch.	Extended actual dispensed amount field. When present in the reversal request message, the value accompanying this FID OVERRIDES the value in the Amount 3 field of the message. As a result, the value in the eight digit Amount 3 field is rendered inconsequential; it should be filled with spaces. Example: [ul000123456789] where '000123456789' represents 123,456,789 units of currency. (NOTE: the units always represent the smallest possible representation of the currency involved. For United States, this would be the number of pennies).

PRELIMINARY

	'x'	Extended the settlement amount in Host Totals. When present in the host totals request message, the value accompanying this FID overrides the value in the Amount 3 field of the message. As a result, the value in the eight digit Settlement field is rendered inconsequential; it should be filled with spaces. Example: [ux000123456789] where '000123456789' represents 123,456,789 units of currency. (NOTE: the units always represent the smallest possible representation of the currency involved. For United States, this would be the number of pennies).
	'!' ASCII Char 21h.	Surcharge amount to be displayed to customer during Surcharge Notification screen for Balance Inquiries. 8 numeric ASCII characters, right justified and zero filled to the left. Represents the amount in the smallest possible unit of currency. Amount cannot be above "00032767". Example: [u!00000100]
'v' ASCII Char – 76h.	Generic Value Added Service(VAS) fields. The first character after the FID 'v' is another character that defines which data element is to be transmitted. The table below describes each data element. A Field Separator must follow each field.	
	'a'	Value Added Service Terminal ID. 15 ASCII characters. Space filled to the right.
	'b'	Postal code of Terminal for Value Added Service. 12 ASCII characters. Space filled to the right.
	'c'	Loyalty card information for Value Added Service. 80 ASCII characters maximum. Variable length.
	'd'	VAS Challenge/Response used in transaction request/response messages. 20 ASCII characters. Each character represents a 4-bit hexadecimal numeral (0-9, A-F). You must obtain a four-digit Triton-waves identification number and a Triton-waves encryption key from Triton Systems. The ATM sends a twenty-digit hexadecimal string to the Triton-waves host processor in a transaction request. The first four digits should be ignored. The processor should DES encrypt the last sixteen digits with the Triton-waves encryption key. Then, the processor should place its four-digit Triton-waves identification number in the first four positions of the response string and the sixteen-digit encryption result in the last sixteen positions, and send the entire twenty-digit string to the ATM in a transaction response. Transactions that have an incorrect response will be reversed. The Triton-waves encryption key will be changed on a yearly basis.
'w'	Maximum withdrawal amount is the largest dollar amount allowed for a withdrawal transaction. The format is 3 or 4 ASCII numeric characters right justified and zero filled to the left. The valid range is $1 to $9999. This does not override any maximum set by the network.	
'x' † ASCII Char – 78h.	Prepaid online and/or prepaid offline fields. The first character after the FID 'x' is another character that defines which data element is to be transmitted. The table below describes each data element. A Field Separator must follow each field.	
	'a'	Telco-product Serial Number for prepaid offline transactions. 16 ASCII characters maximum
	'b'	User's mobile phone number for prepaid offline or online transactions when used in a transaction request. Activation phone number for prepaid offline or online transactions when used in a transaction response. 20 ASCII characters maximum
	'c'	Taxes 1 – (GST/HST/PST – General Sales Tax / Harmonized Sales Tax / Provencial Sales Tax for offline transactions) or general taxes for offline transactions. 8 numeric Zero filled to the left. Represents the amount in the smallest possible unit of currency.

PRELIMINARY

	'd'	Taxes 2 – General taxes for online transactions. 8 numeric Zero filled to the left. Represents the amount in the smallest possible unit of currency. The value sent in a transaction request is the total tax as calculated by the ATM, and the value received in the transaction response will be printed to the customers receipt. Amount cannot be above "00032767".
	'e'	Total number of prepaid online transactions and the total amount of all prepaid online transactions. (12 Numeric total) 4 Numeric – Total number of Wireless Phone Refill transactions since the last request for totals. Zero filled to the left. 8 Numeric – Total amount of all Wireless Phone Refill transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
	'f'	Contact phone number for prepaid online transactions. 20 ASCII characters maximum
	'g'	Service provider Name for prepaid online transactions. 24 ASCII characters maximum
	'h'	6 Digit (numeric) Expiration date (MMDDYY) for online transactions.
	'i'	Secret Number for prepaid offline transactions 20 ASCII characters maximum
	'n'	Actual amount debited from the customer's account for online transactions. 8 Digit (numeric) Zero filled to the left. Represents the amount in the smallest possible unit of currency.
	'p'	Voucher PIN number for online transactions. 25 ASCII characters maximum

PRELIMINARY

	'q'	Service provider name and fast cash amounts for wireless transactions. - 108 bytes maximum 24 Alphanumeric max – (required) Service provider name – (variable length). 1 RS character – (required) 0x1E 1 Numeric – (required) 1=enable provider name, 0=disable provider name. 1 RS character – (optional if disabling provider name) 0x1E 48 Numeric – (optional if disabling provider name) Six fast cash amounts, each amount is 8 Numeric characters. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency. 1 RS character – (optional - if disabling provider name) 0x1E 6 Numeric – (optional - if disabling provider name) Service provider ID 1 RS character – (optional - based on implementation) 0x1E 16 Numeric – (optional - based on implementation) Two more fast cash amounts, each amount is 8 Numeric characters. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency. 1 RS character – (optional - based on implementation) 0x1E 16 Numeric – (optional - based on implementation) Provider Surcharge Amount, 8 Numeric characters. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency. An ASCII zero alone in the service provider name will disable all previous programmed service provider names and fast cash amounts. Example: xq0[RS]0
	'r'	Purchase new long distance calling plan. Used when the transaction code is "53". 01 – US calling – penny plan. 02 – US calling – long talk plan. 03 – US calling – short talk plan. 04 – US calling – best value plan. 05 – Mexico calling – long talk plan. 06 – Mexico calling – short talk plan. 07 – Mexico calling – best value plan. 08 – International calling – long talk plan. 09 – International calling – short talk plan. 10 – International calling – best value plan.
	's'	Total number of long distance transactions and the total amount of all long distance transactions. (12 Numeric total) 4 Numeric – Total number of long distance transactions since the last request for totals. Zero filled to the left. 8 Numeric – Total amount of all long distance transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.

PRELIMINARY

	't'	Tax rate – General tax rate for wireless or long distance transactions. 4 numeric. Zero filled to the left. Represents a percentage from 00.00% to 99.99%. The decimal should not be included in the field. Amount cannot be above "9999". Will be used starting with the *next* transaction.
	'u'	Service provider ID. 6 numeric. fixed length.
	'v'	Service provider transaction totals and amounts. (18 Numeric total) 6 Numeric – Service Provider ID. 4 Numeric – Total number of service provider transactions since the last request for totals. Zero filled to the left. 8 Numeric – Total amount of all service provider transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
'y' † ASCII Char – 79h.	Bill Payment, Electricity Prepaid, and Check Cashing fields. The first character after the FID 'y' is another character that defines which data element is to be transmitted. The table below describes each data element. A Field Separator must follow each field.	
	'a'	Bill Payment account or electricity meter number. 39 ASCII characters maximum
	'b'	Receiver name. 35 ASCII characters maximum
	'c'	Check cashing Authorization code. 9 numeric. fixed length
	'd'	Check cashing PIN. 3 to 6 numeric.
	'e'	Total number of check cashing transactions and the total amount of all check cashing transactions. (12 Numeric total) 4 Numeric – Total number of check cashing transactions since the last request for totals. Zero filled to the left. 8 Numeric – Total amount of all check cashing transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
	'f'	Total number of Bill Payment transactions and the total amount of all Bill Payment transactions. (12 Numeric total) 4 Numeric – Total number of Bill Payment transactions since the last request for totals. Zero filled to the left. 8 Numeric – Total amount of all Bill Payment transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
'z' † ASCII Char – 7Ah.	Funds fields. The first character after the FID 'z' is another character that defines which data element is to be transmitted. The table below describes each data element. A Field Separator must follow each field.	
	'a'	16 Digit (alphanumeric) Sender ID for Funds transactions. For send transactions the card PAN is used for the PIN Block calculation. Sender ID is encrypted as an ANSI standard PIN Block. For receive transactions, code '49', the Sender ID element is sent in clear text (in its unencrypted form) and padded on the right with a character 'C'. The Sender ID must be constructed by padding the clear Sender ID to sixteen digits with the ASCII character C (ASCII hex 0x43). The clear Sender ID 2625335654 would appear as 2625335654CCCCCC.
	'b'	9 Digit (numeric) Preferred Customer ID for Funds transactions.

PRELIMINARY

	'c'	6 Digit (numeric) Confirmation ID for Funds transactions.
	'd'	8 Numeric – State tax value for Funds transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
	'e'	16 ASCII characters. Money Transfer Control Number (MTCN) for Funds transactions.
	'f'	1 ASCII character. Enable or disable funds field. For Configuration Download Response transaction message: ASCII '0' – Disable funds transactions (disabled by default) ASCII '1' – Enabled funds transactions.
	'g'	20 ASCII characters maximum. Assistance phone number for Funds transactions.
	'h'	8 Numeric – Maximum request amount for a Funds Transaction. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
	'i'	8 Numeric – Smallest denomination multiple for a Funds Transaction. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
	'j'	8 Numeric – Funds convenience fee. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
	'k'	Total number of funds-send transactions and the total amount of all funds-send transactions. (12 Numeric total) 4 Numeric – Total number of funds-send transactions since the last request for totals. Zero filled to the left. 8 Numeric – Total amount of all funds-send transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
	'm'	Total number of funds-receive transactions and the total amount of all funds- receive transactions. (12 Numeric total) 4 Numeric – Total number of funds- receive transactions since the last request for totals. Zero filled to the left. 8 Numeric – Total amount of all funds- receive transactions. Zero filled to left, right justified. Represents the amount in the smallest possible unit of currency.
	'n'	PAN used for Western Union Receive transaction request. 16 Numeric characters.
	'o'	15 Digit (numeric) Extended Length Confirmation ID for Funds transactions.

'~' (tilde) ASCII Char 7Eh	DES Communications key (PIN Working Key). 16 ASCII characters long, encrypted under Master key. Will be used starting with the *next* transaction. Same format as this field in configuration table download. Triple-DES Encrypted PIN Key 1 (*K1)* if the terminal is Triple-DES capable (See 10.7). Each character represents a 4-bit hexadecimal numeral (0-9, A-F).

PRELIMINARY

'{' † (left brace) ASCII Char 7Bh	Triple-DES Encrypted PIN Key 2 (*K2)* (See 10.7) when used in a Response Message. 16 ASCII characters. Each character represents a 4-bit hexadecimal numeral (0-9, A-F). The absence of this FID in a Response Message(Transaction or Configuration Table Download) will set K2 equal to K1 (*K2=K1)* and the '[' FID equal to zero(0) for all request messages if the terminal is Triple-DES capable, which is identical to Single-DES encryption of the PIN block. The presence of this FID in a Response Message(Transaction or Configuration Table Download) will set the '[' FID equal to two (2) for all request messages if the terminal is Triple-DES capable. Will be used starting with the *next* transaction.
'}' † (right brace) ASCII Char 7Dh	MAC (Message Authentication Code) Working Key Right Block for Triple-DES capable terminals. 16 ASCII characters encrypted under MAC Master key. Each character represents a 4-bit hexadecimal numeral (0-9, A-F). Will be used starting with the *next* transaction. See 10.8 for the complete description of Encryption key sets. The presence of both the 'm' FID and this FID in a Response Message (Transaction or Configuration Table Download) will enable Triple-DES MAC Encryption if the terminal is Triple-DES capable.
'[' † (left bracket) ASCII Char 5Bh	Terminal is Triple-DES capable / Encryption Mode Flag (EMF) (See 10.7) 1 ASCII character. 0(default) = Terminal is Triple-DES capable (reset to zero (0) by absence of the '{' FID in a Transaction or Configuration Table Download message from the host) 2 = Terminal is Triple-DES capable (set to two (2) by the presence of the '{' FID in a Transaction or Configuration Table Download message from the host) Terminals that are not Triple-DES capable will not send this FID, therefore its absence indicates the terminal is Single-DES only. The value of this FID (0 or 2) is also the value of the Encryption Mode Flag in the communication header of request messages.

Table 111 – Miscellaneous Field ID Descriptions

This table describes the Field ID codes that can appear in the miscellaneous fields in the response message. In all cases, the Field ID (FID) Code character *must* be the first character in the field, with the appropriate data following directly after the FID.
Any field described here may be placed in either field Miscellaneous 1, Miscellaneous 2, or Miscellaneous X, where appropriate.

PRELIMINARY

10.2.20 **Valid Miscellaneous Field ID Table**

FID		Transaction Message		Config. Message		Host Totals Message		Reversal Message	
1st char	2nd char	Request	Response	Request	Response	Request	Response	Request	Response
! – 21h			X		X				
# - 23h		X	X					X	X
& - 26h		X	X						
/ - 2Fh			X		X				
@ - 40h		X							
[- 5Bh				X		X		X	
^ - 5Eh		X	X			X	X	X	X
{ - 7Bh			X		X				
~ - 7Eh			X		X				
} – 7Dh			X		X				
a		X	X	X	X	X	X	X	X
b			X						
c			X						
d			X						
e		X	X						
f									
g		X						X	
h									
i		X						X	
j		X	X	X	X	X	X	X	X
k			X					X	
l		X	X						
m			X						
n								X	
o						X	X		
p			X				X		
q									
s									
t			X						
u	a		X						
	b	X	X	X	X	X	X	X	X
	c								
	d	X	X					X	
	e		X		X				
	f		X		X				

PRELIMINARY

FID		Transaction Message		Config. Message		Host Totals Message		Reversal Message	
1st char	2nd char	Request	Response	Request	Response	Request	Response	Request	Response
	g		X				X		X
	h	X	X					X	
	i		X						
	j		X						
	k			X					
	l							X	
	x					X			
	! – 21h		X		X				
v	a	X				X		X	
	b	X						X	
	c	X							
	d	X	X						
w			X						
x	a		X						
	b	X	X						
	c		X						
	d	X	X						
	e					X	X		
	f		X						
	g	X	X						
	h		X						
	i		X						
	n		X						
	p		X						
	q		X		X				
	r	X							
	s					X	X		
	t		X		X				
	u	X						X	
	v					X	X		
y	a	X							
	b	X							
	c	X							
	d	X							
	e					X	X		
	f					X	X		
z	a	X							

PRELIMINARY

FID		Transaction Message		Config. Message		Host Totals Message		Reversal Message	
1st char	2nd char	Request	Response	Request	Response	Request	Response	Request	Response
	b	X							
	c	X	X						
	d		X						
	e		X						
	f				X				
	g				X				
	h				X				
	i				X				
	j		X						
	k					X	X		
	m					X	X		
	n		X		X				

Table 112 – Valid Miscellaneous Field ID Tables

X = FID Allowed for message type.
See "Miscellaneous Field ID Descriptions" for data format and construction. The highlighted Field IDs are undefined.

PRELIMINARY

10.3 PROTOCOL SPECIFICATION

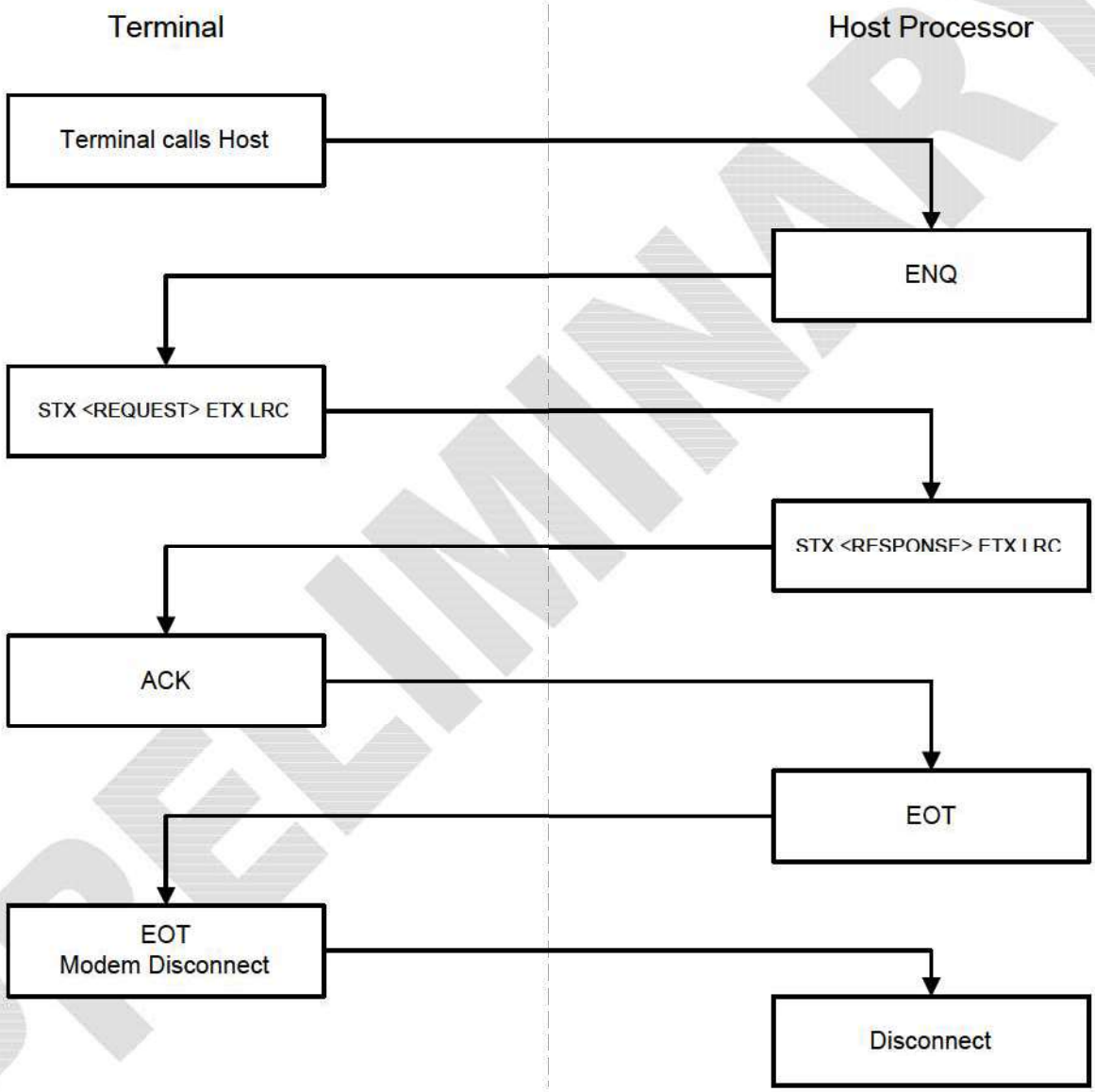
The Triton terminal communications protocol closely follows the VISA II POS Authorization protocol. The following diagrams illustrate the protocol level for messages sent between the terminal and the processor.

Diagrams are provided for the two modes of normal operation (terminal expects EOT, and terminal does not require EOT), as well as conditions that result in cancelled transactions.

PRELIMINARY

10.3.1 Normal Message w/EOT

This diagram illustrates the line protocol for a normal dispense cycle. Dispense occurs after Host EOT is received (default).

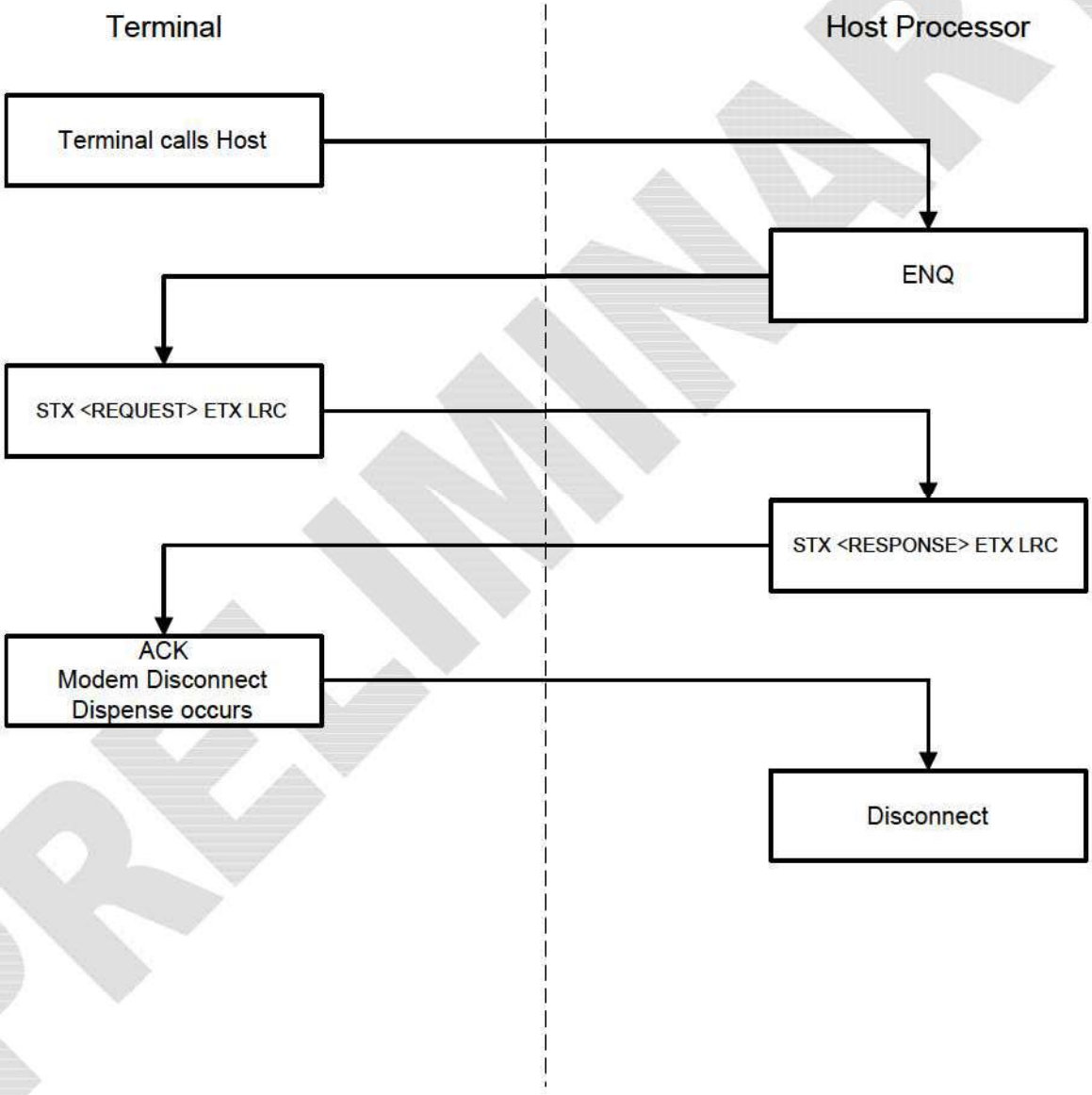


In this mode, the terminal will not dispense unless it receives the EOT from the host.

PRELIMINARY

10.3.2 Normal Message w/o EOT

This diagram illustrates the line protocol for a normal dispense cycle when the terminal is not expecting an EOT. Dispense occurs after the terminal sends the ACK to the HOST. If the host sends an EOT it will be ignored.

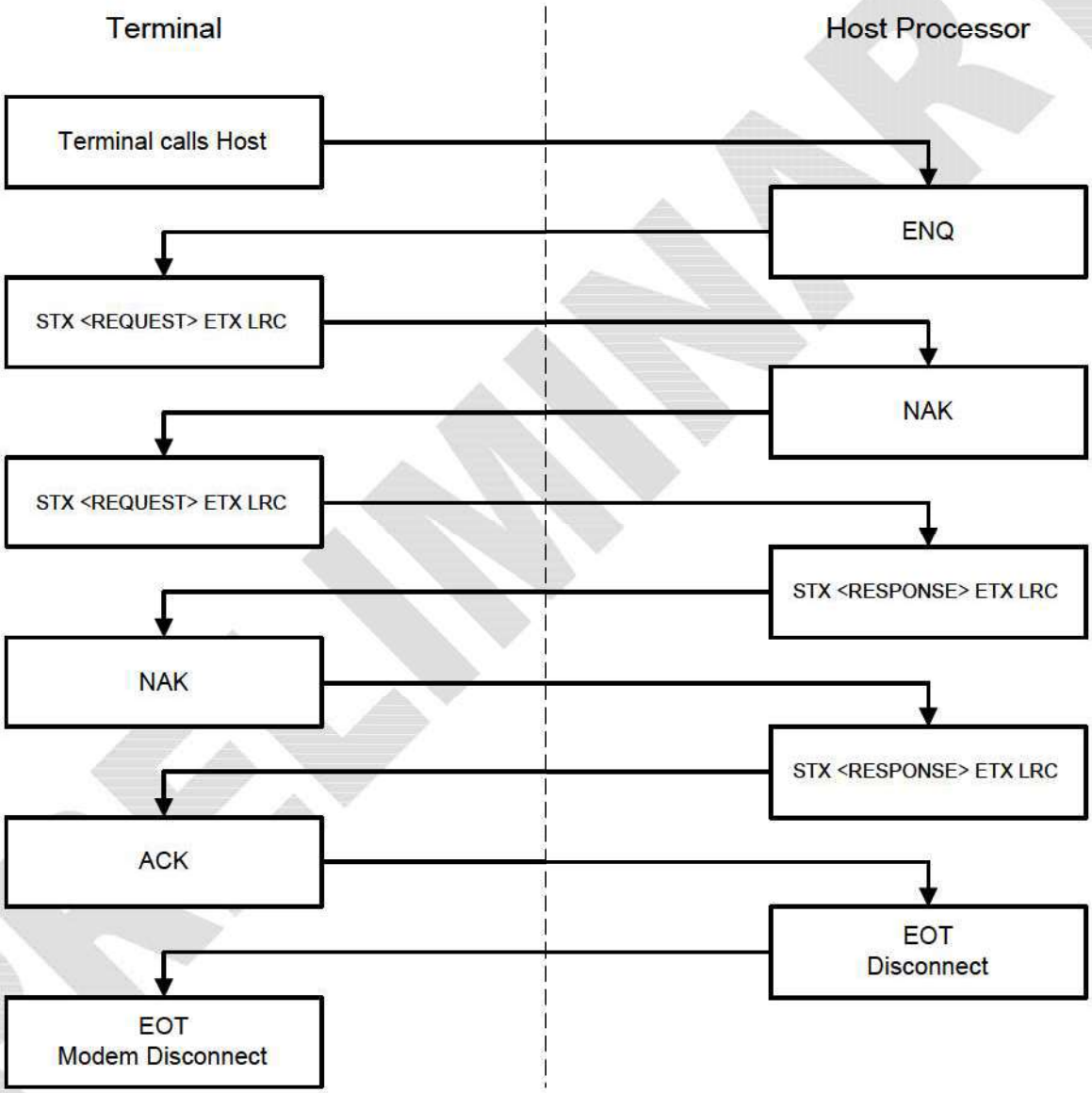


PRELIMINARY

Highly Confidential: Property of Triton Systems – Do Not Disclose

10.3.3 Correctable Request and Response Transmit Error

This diagram illustrates the line protocol for a correctable transmit error in the request and response messages. The NAK character is sent to indicate an error in the LRC.

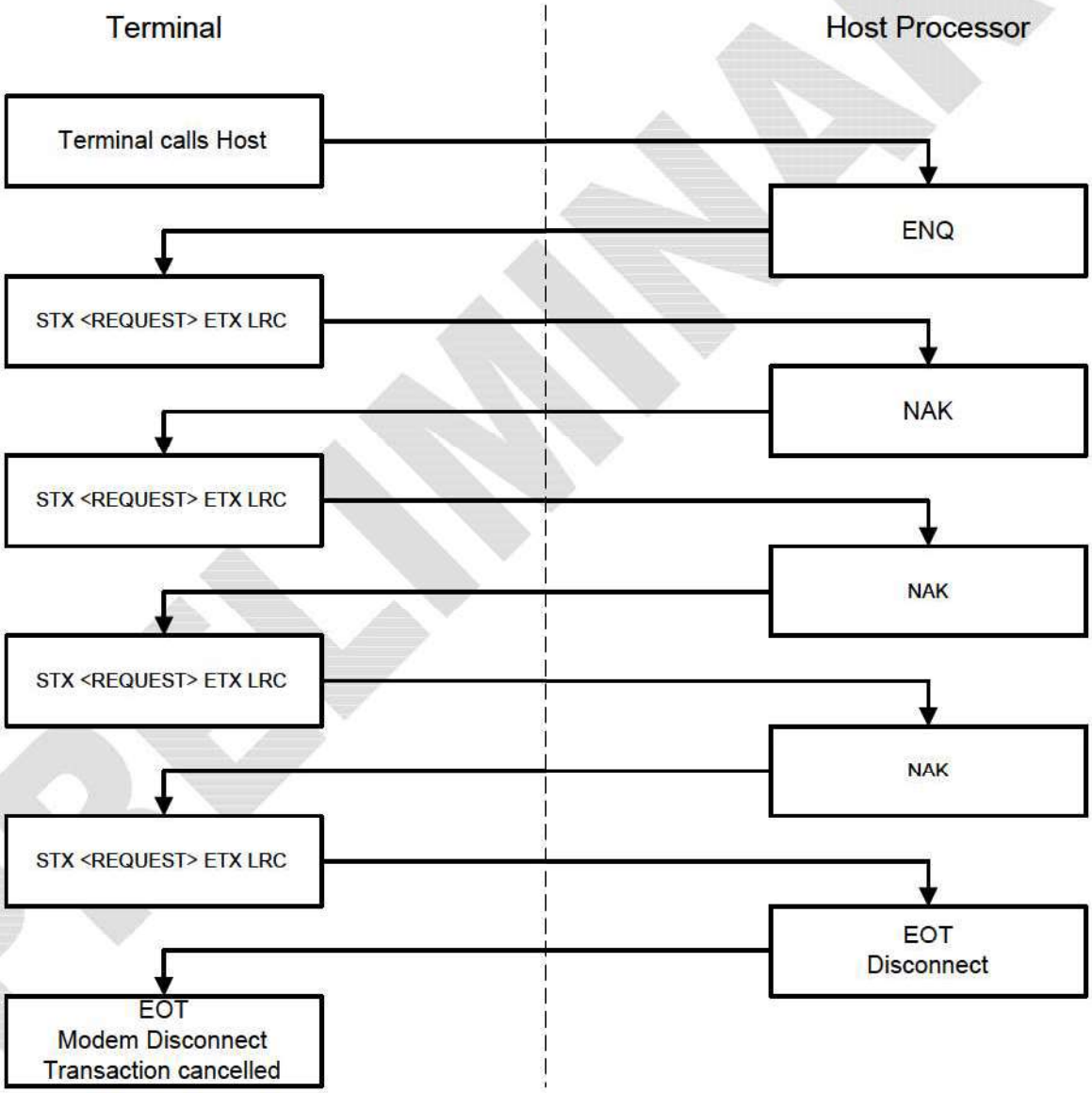


PRELIMINARY

Highly Confidential: Property of Triton Systems – Do Not Disclose

10.3.4 Uncorrectable Request Transmit Error

This diagram illustrates the line protocol for an uncorrectable transmit error in the request message. The NAK character is sent from the host to indicate an error in the LRC. The terminal will not dispense if this happens.

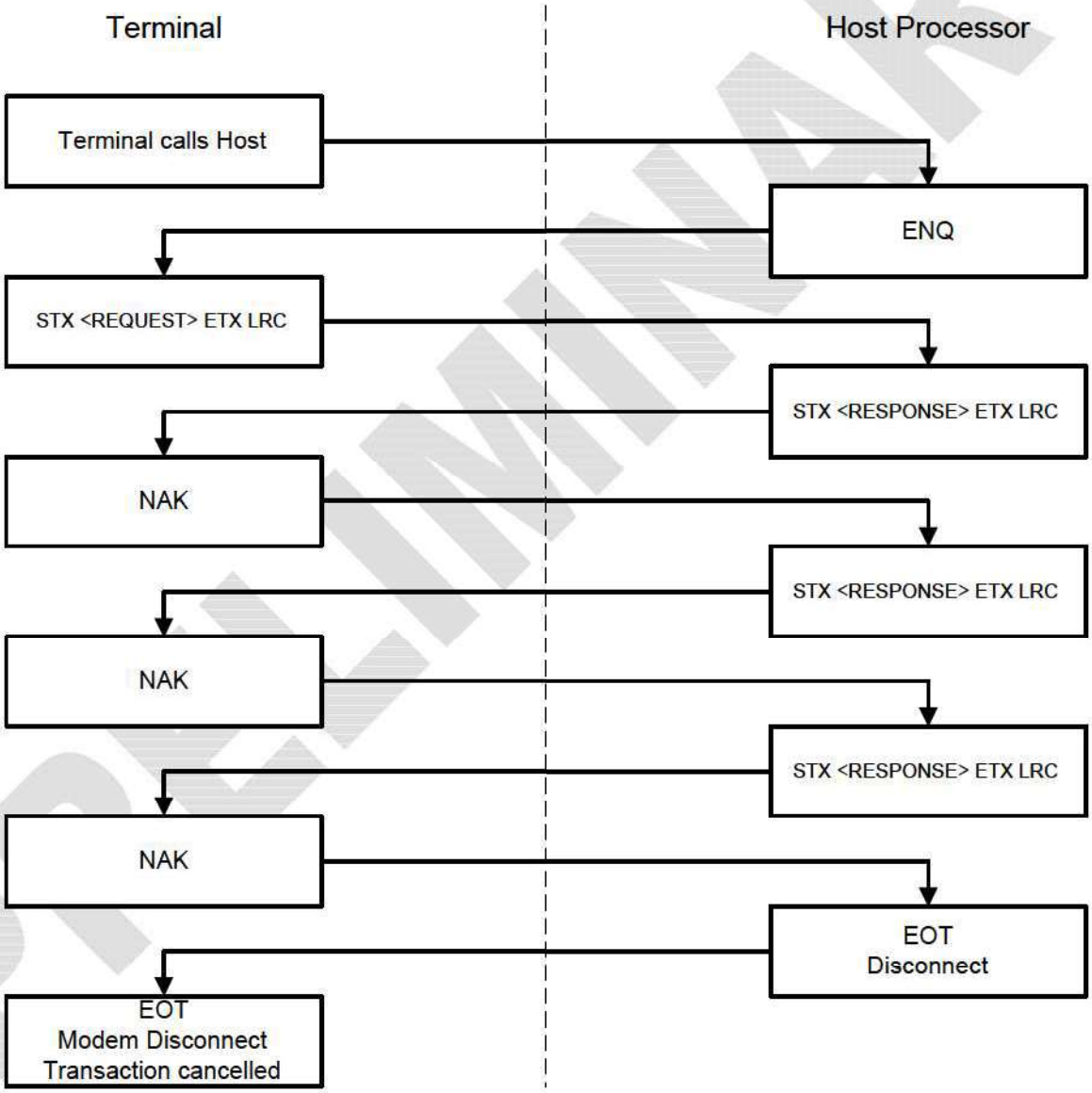


PRELIMINARY

Highly Confidential: Property of Triton Systems – Do Not Disclose

10.3.5 Uncorrectable Response Transmit Error

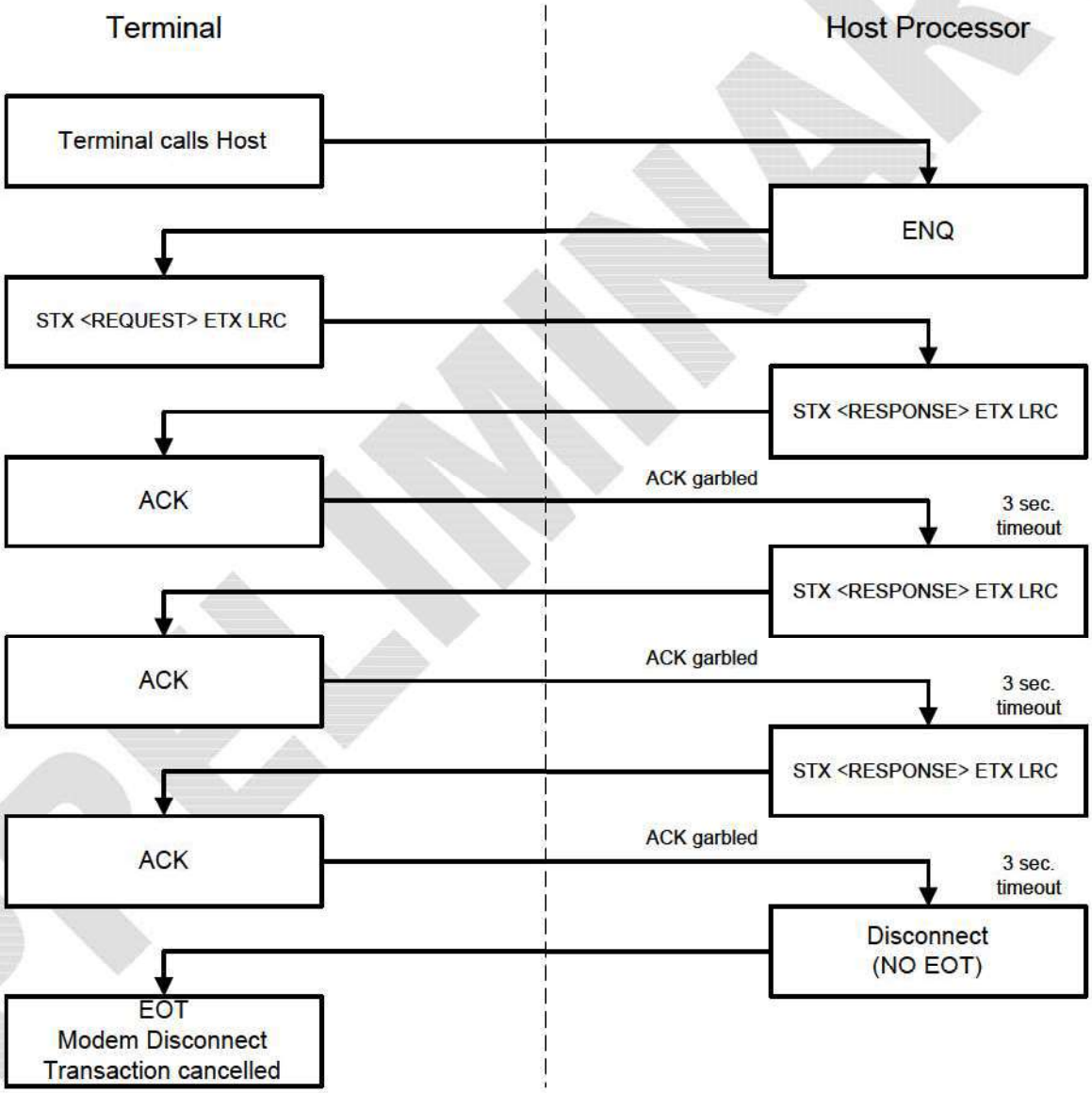
This diagram illustrates the line protocol for an uncorrectable transmit error in the response message. The NAK character is sent from the terminal to indicate an error in the LRC. The terminal will not dispense if this happens.



PRELIMINARY

10.3.6 Terminal ACK Not Received By Host

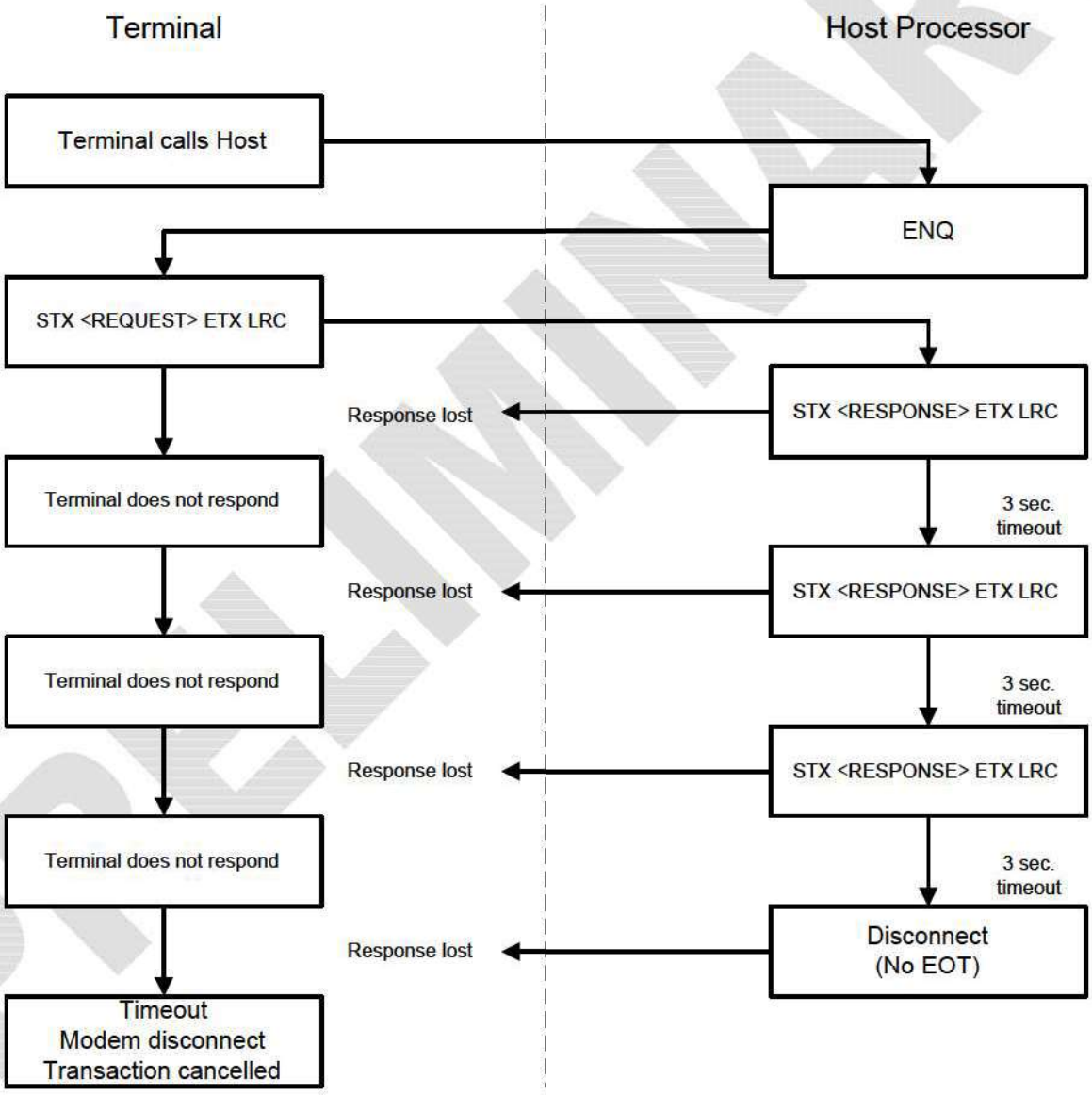
This diagram illustrates the line protocol when the host does not receive the ACK properly. In this scenario, the terminal will not dispense because no EOT was received (assuming the terminal is configured to expect EOT). The host should reverse the transaction.



PRELIMINARY

10.3.7 Terminal Does Not Acknowledge Response From Host

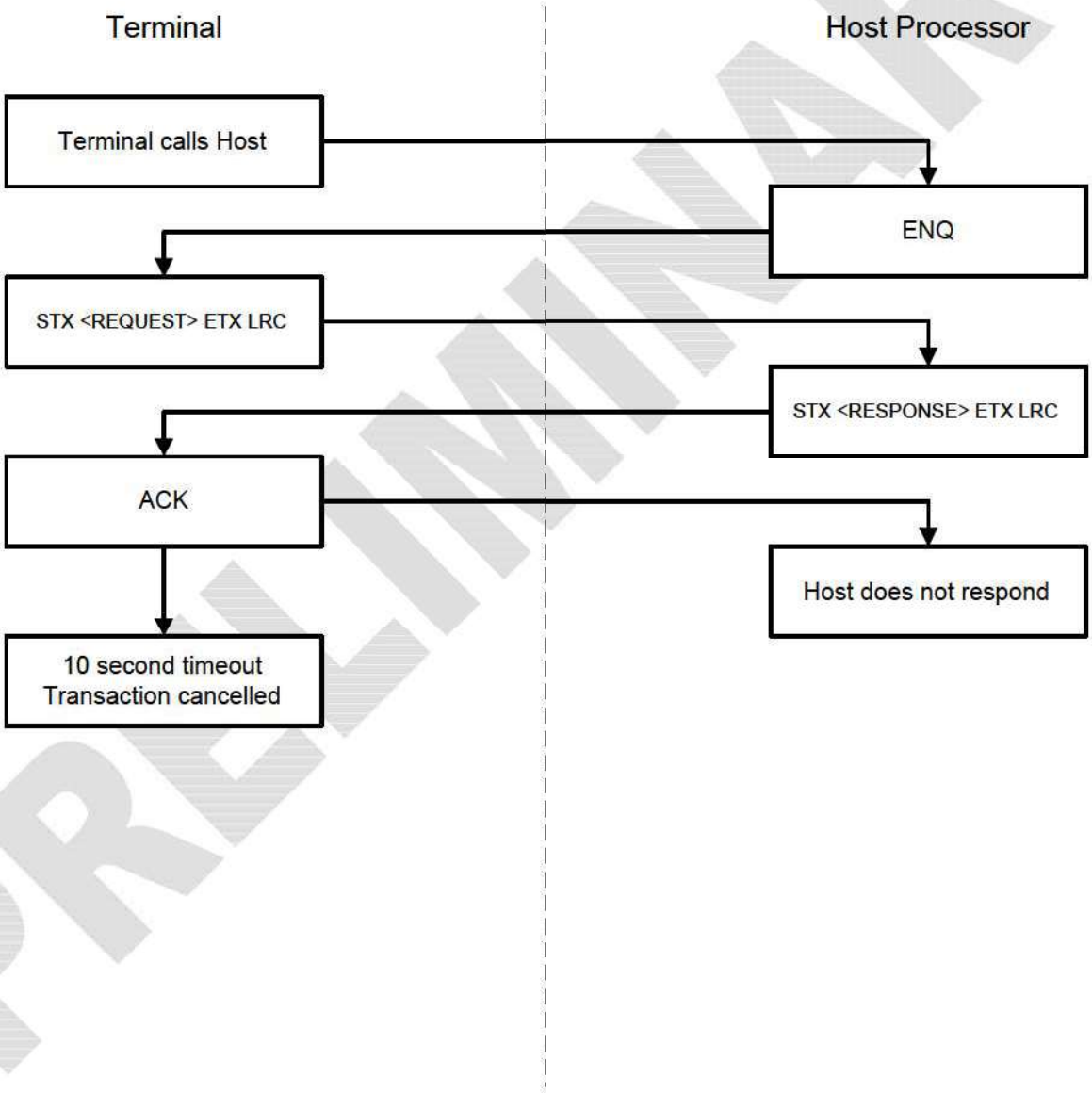
This diagram illustrates the line protocol when the terminal does not receive the response properly from the host. In this scenario, the terminal will not dispense the requested amount (if it was approved). The host should reverse the transaction.



PRELIMINARY

10.3.8 Host Does Not Respond to ACK from Terminal (Default Mode)

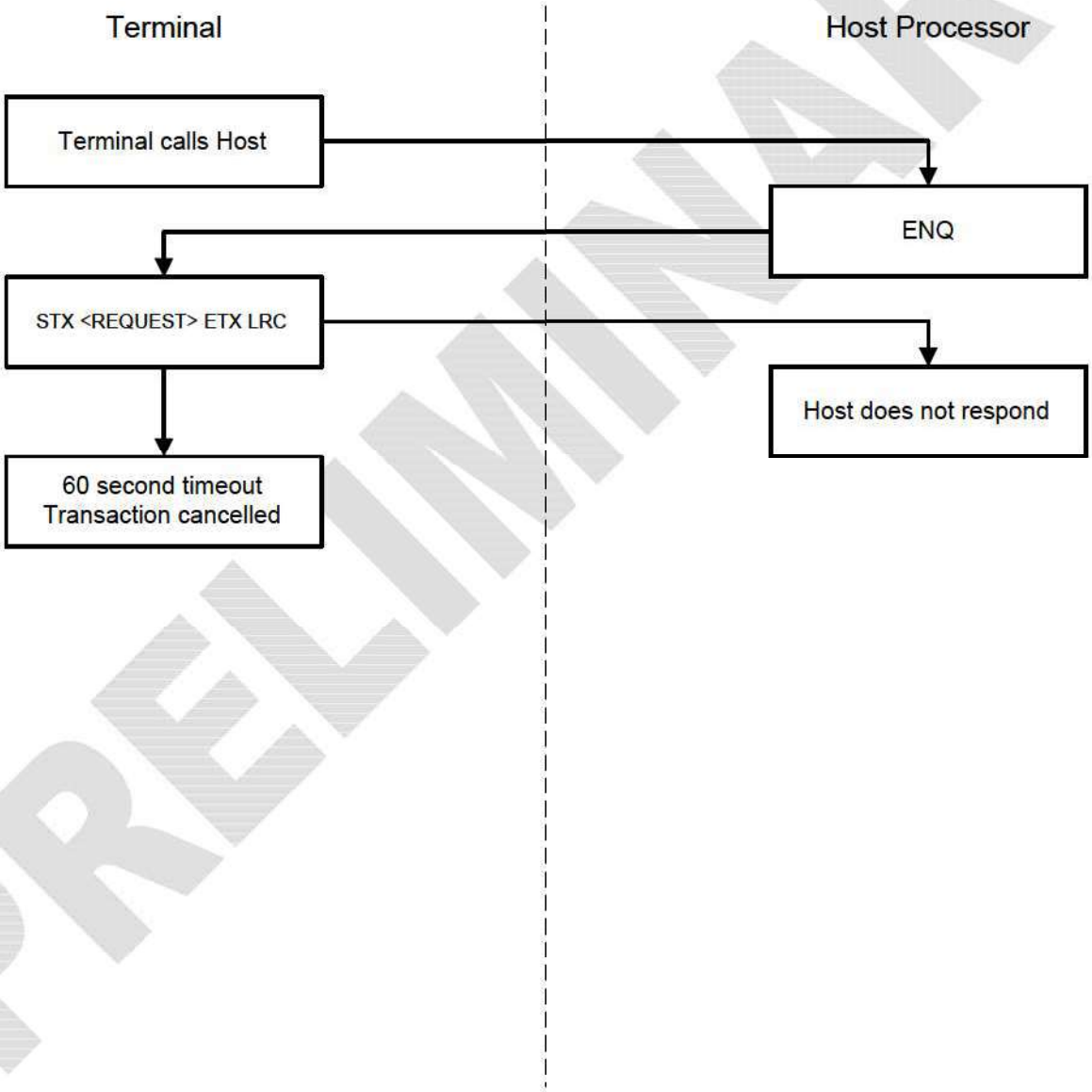
This diagram illustrates the line protocol when the host does not respond with an EOT to the ACK from the terminal when operating in default mode (terminal requires Host EOT to dispense). The terminal will not dispense in this scenario.



PRELIMINARY

10.3.9 Host Does Not Respond to Request Message

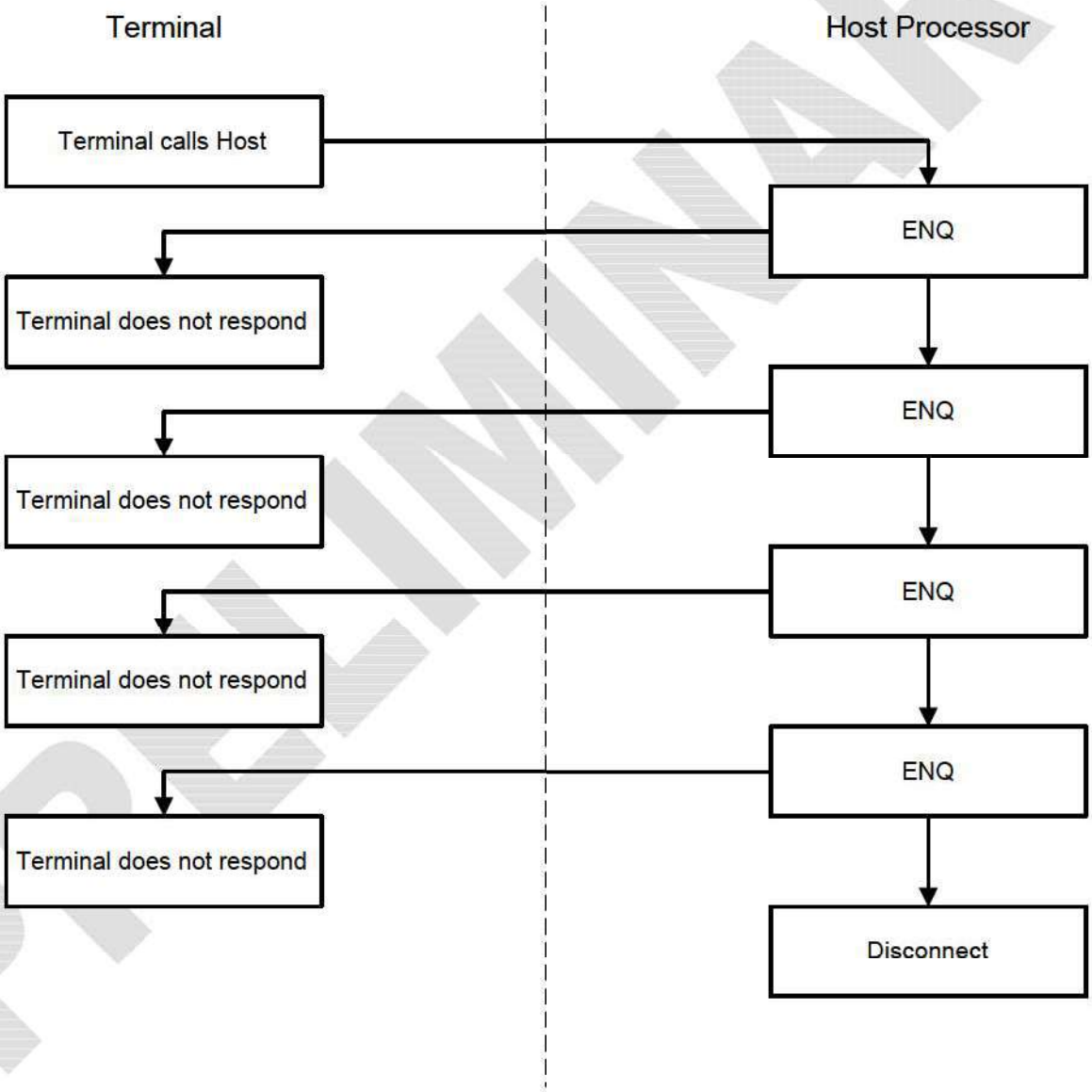
This diagram illustrates the line protocol when the host does not respond to the request message at all. The terminal will not dispense if this happens.



PRELIMINARY

10.3.10 Terminal Does Not Respond to Host ENQ

This diagram illustrates the line protocol when the terminal does not respond to the ENQ character. The host should send at least 4 ENQ characters, one second apart, although this is not required.



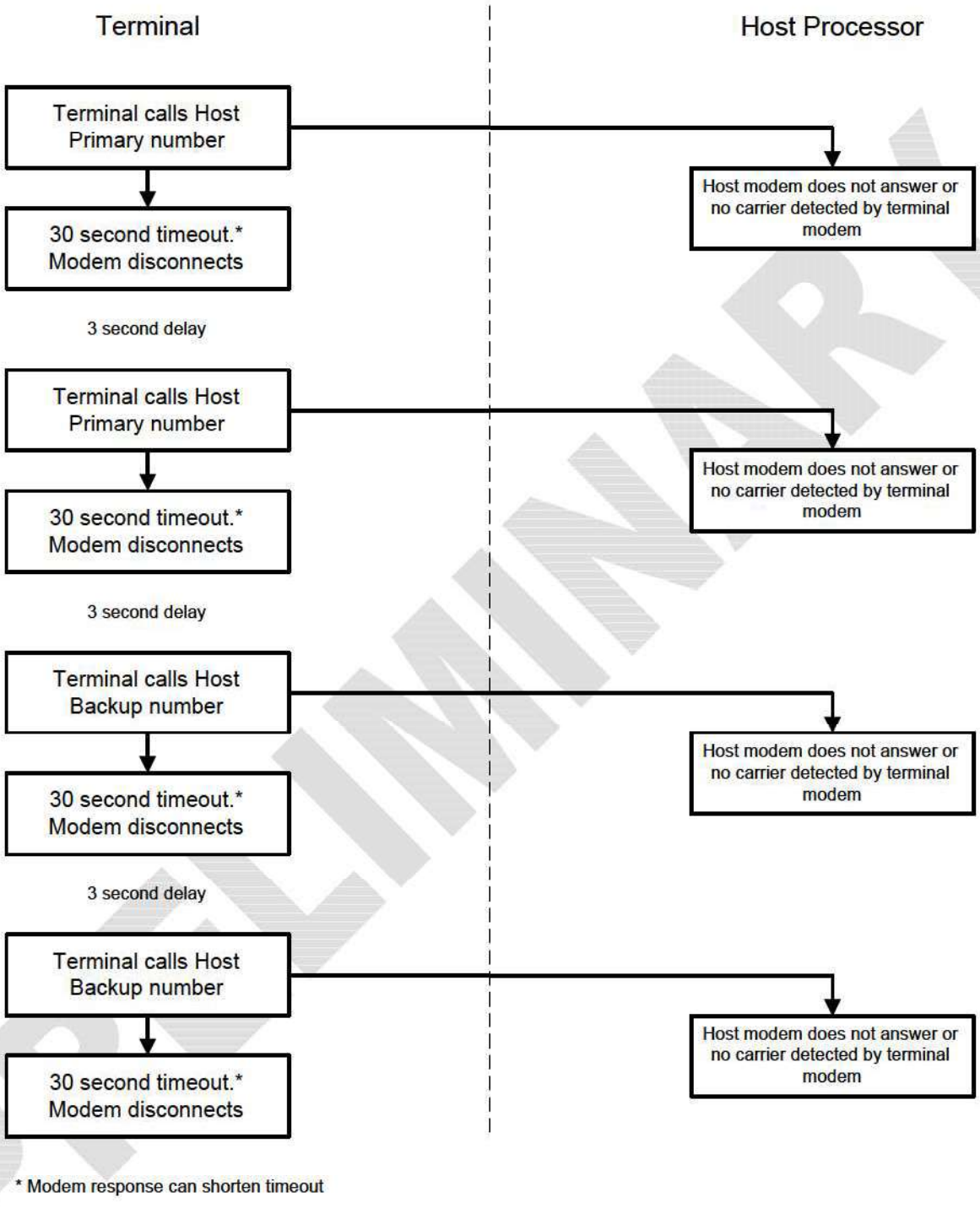
PRELIMINARY

Highly Confidential: Property of Triton Systems – Do Not Disclose

10.3.11 Host Does Not Answer Terminal Call

This diagram illustrates the line protocol when the host does not answer the call. The terminal will try the primary number twice, then the backup number (if programmed) twice. If no answer is received, the transaction will be cancelled.

PRELIMINARY



* Modem response can shorten timeout

PRELIMINARY

10.4 ANSI PIN Block Format

This section includes two sections from the American National Standards Institute (ANSI) document titled *American National Standard For Personal Identification Number (PIN) Management and Security X.9.8*. This document describes, among other things, the techniques used to secure transmissions of customer PINs over insecure communications pathways. It is presented as a reference only.

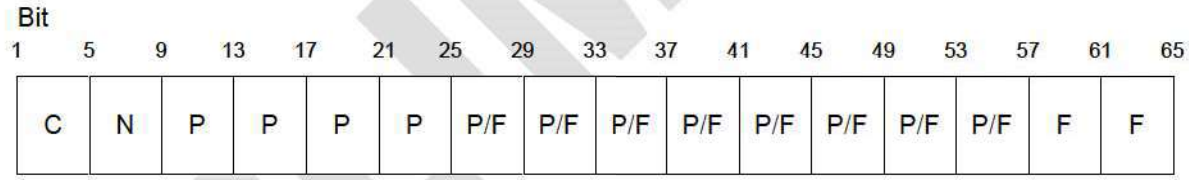
10.4.1 Clear-Text PIN Block Format

The PIN is assumed to consist of 4-12 decimal digits and *shall* be entered by a customer, in sequence as specified by Section 3.5.1.2, with the first digit entered referred to as digit 1 and the last referred to as digit N.

The clear-text PIN format specified *must* be used in interchange and may be used in the terminal to acquirer segment of the network. Bit positions are specified based on numbering the bits in the DEA input block as in Section 3.7.2.2.1.

The clear-text PIN and the customer's account number are constructed as follows:

PIN Block Construction



C - Control field. Currently, the only defined value is 0000 (binary) which designates the block as containing 4 to 12 four-bit decimal PIN digits representing user-entered characters according to Section 3.5.1.1. Future use of different values of this field could accommodate different character sets.

N - PIN length entered field. Four-bit binary number with permissible values of 0100 (=4) to 1100 (=12)

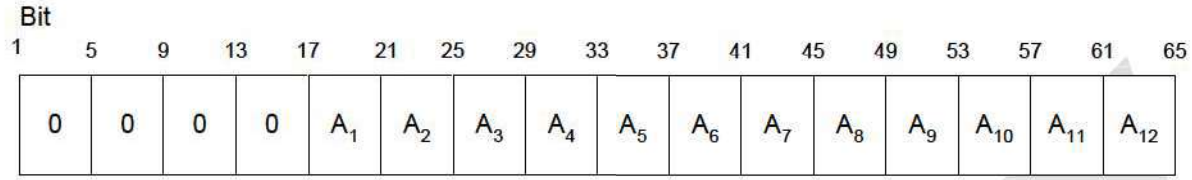
P - PIN digit. Four-bit field having possible values from 0000 to 1001 as defined by Section 3.5.1.1.

F- Fill digit. Four-bit field having fixed value 1111.

P/F - PIN digit or fill digit, as determined by PIN length entered field (N).

PRELIMINARY

Account Number Block Construction



A - The 12-right-most digits of the primary account number (PAN), excluding the check digit. A_1 is the most significant digit and A_{12} is the digit immediately preceding the PAN's check digit.

0 - A decimal pad digit-zero. The first 4-bit fields of the account number block are always padded with this value.

Formatted Clear-Text PIN Block

Next, the PIN and account number blocks are Exclusive OR'd before assembled in the DEA input register.

When the customer's account number is not available, other cryptographic techniques should be used to protect the PIN. 10.7 (ANSI X.9.8) illustrates one such method.

Any network or interchange node having access to the clear-text PIN block (i.e., during a decryption/re-encryption or during a PIN verification) should reject transactions having any of the following:

a. Control field value of other than 0000
b. A PIN length entered field of value less that 4 or greater that 12
c. Any PIN digit from P_1 through P_N of value greater than 9

This serves as a reasonable check on PIN encryption/decryption processing, but does not serve as any indication of validity of the customer-entered PIN.

10.4.2 Cipher-Text PIN Format

The formatted clear-text PIN block, defined above, then *shall* be encrypted using the DEA on Electronic Code-Book (ECB) mode as shown in Section 10.7. The 64-bit output cipher block is called the reversibly encrypted PIN (or encrypted PIN).

The encrypted PIN *shall* be transmitted between financial institutions as a 64-bit entity in bit oriented communications, as eight 8-bit bytes in 8-bit transparent character oriented communications, as sixteen characters representing hexadecimal digits (0, 1, …, 9, A, B, …, F), or by any other data representation agreed upon by sender and receiver. (Triton Note: The TDL legacy convention uses 16 hexadecimal digits represented as ASCII characters).

The order of transmission *shall* be for left to right, with respect to the diagram in Section 10.7. For example, in 8-bit transparent communications the first byte transmitted will contain bits 1 though 8 of the DEA output block; the eight byte transmitted will contain bits 57 through 64 of the DEA output block.

PRELIMINARY

PRELIMINARY

10.5 MAC Protocol Addendum

The MAC fields implemented in this specification follow the ANSI Standard X9.19 format.

The Triple-DES MAC Algorithm implemented in this specification uses "MAC Algorithm 3 - Padding Method 1" as defined in ISO/IEC 9797-1.

This section describes the actual implementation of the fields in the message.

1. Insert a Field ID '^' into one of the miscellaneous fields in the formatted message. Fill this field with 9 ASCII zero characters.
2. Create an 8-byte buffer in RAM called 'INPUT' and fill this block with hex zeroes.
3. Create an 8-byte buffer in RAM called 'OUTPUT' and fill this block with hex zeroes.
4. Begin at the STX character of the formatted message.
5. Take the next 8 ASCII characters from the formatted message (include STX first time through) and transfer them to the INPUT buffer. Use all characters up to and including the ETX. If less than 8 characters are available, pad the remaining position(s) with Binary zeroes.
6. Exclusive-OR the OUTPUT buffer with the INPUT buffer and store in the INPUT buffer.
7. Encrypt the INPUT buffer using DES, or Triple-DES if Triple-DES MAC is enabled and this is the last block(see Figure B1), with the MAC working key(s), MK1 and MK2, and store in the OUTPUT buffer.
8. Repeat steps 5 through 7 until the end of the message is reached (ETX).
9. Discard the right-most 32 bits of the OUTPUT buffer. Convert the remaining 32 bits into ASCII characters, 4 bits at a time to produce 8 ASCII characters.
10. Insert an ASCII space character (20h) after the fourth character.
11. The format of the string should be as follows: 'XXXX<sp>XXXX', where X is a hexadecimal ASCII character (0-F).
12. Insert this 9 character string in place of the 9 ASCII zeroes in the Field ID '^' created in step 1.
13. Calculate the LRC of the message with the MAC field in place.

PRELIMINARY

MAC Field Use

When enabled, the terminal will send the MAC field with all message requests except Configuration Table Download request, and will expect them in all message responses except Configuration Table Download response. If the terminal receives an invalid MAC field in a response message, it will immediately perform a Configuration Table Download on a separate call to update the MAC and/or COMMS key. If the failed MAC was contained in a Transaction Response Message and a withdraw or purchase was performed, a reversal will follow the Configuration Table Download, and if a withdraw was performed, no cash will be dispensed in this case.

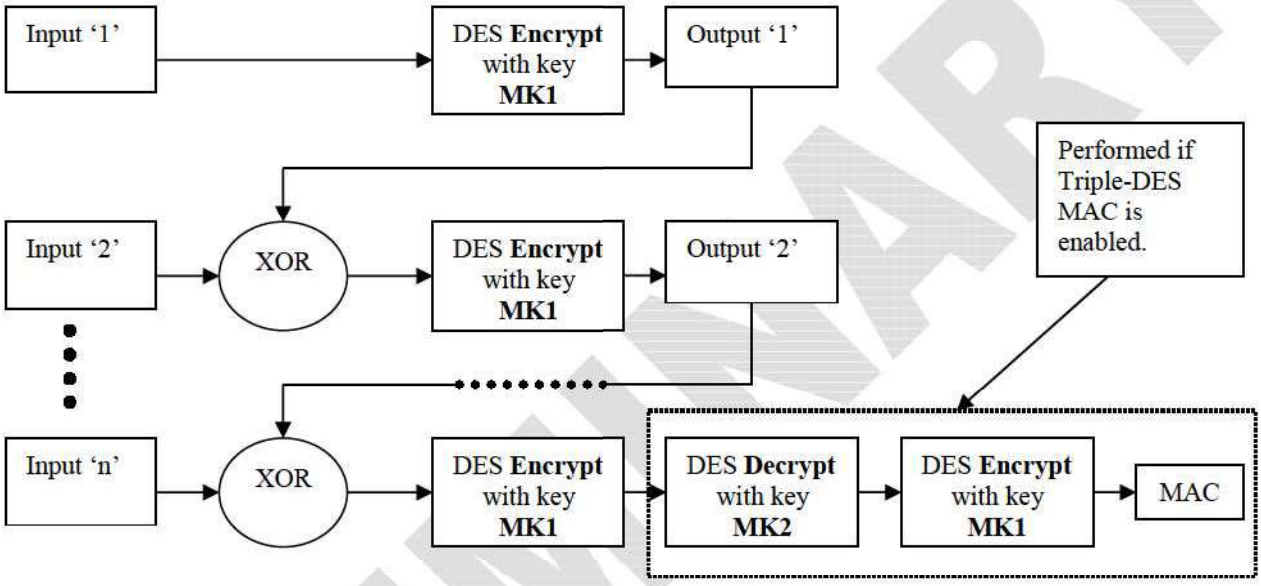
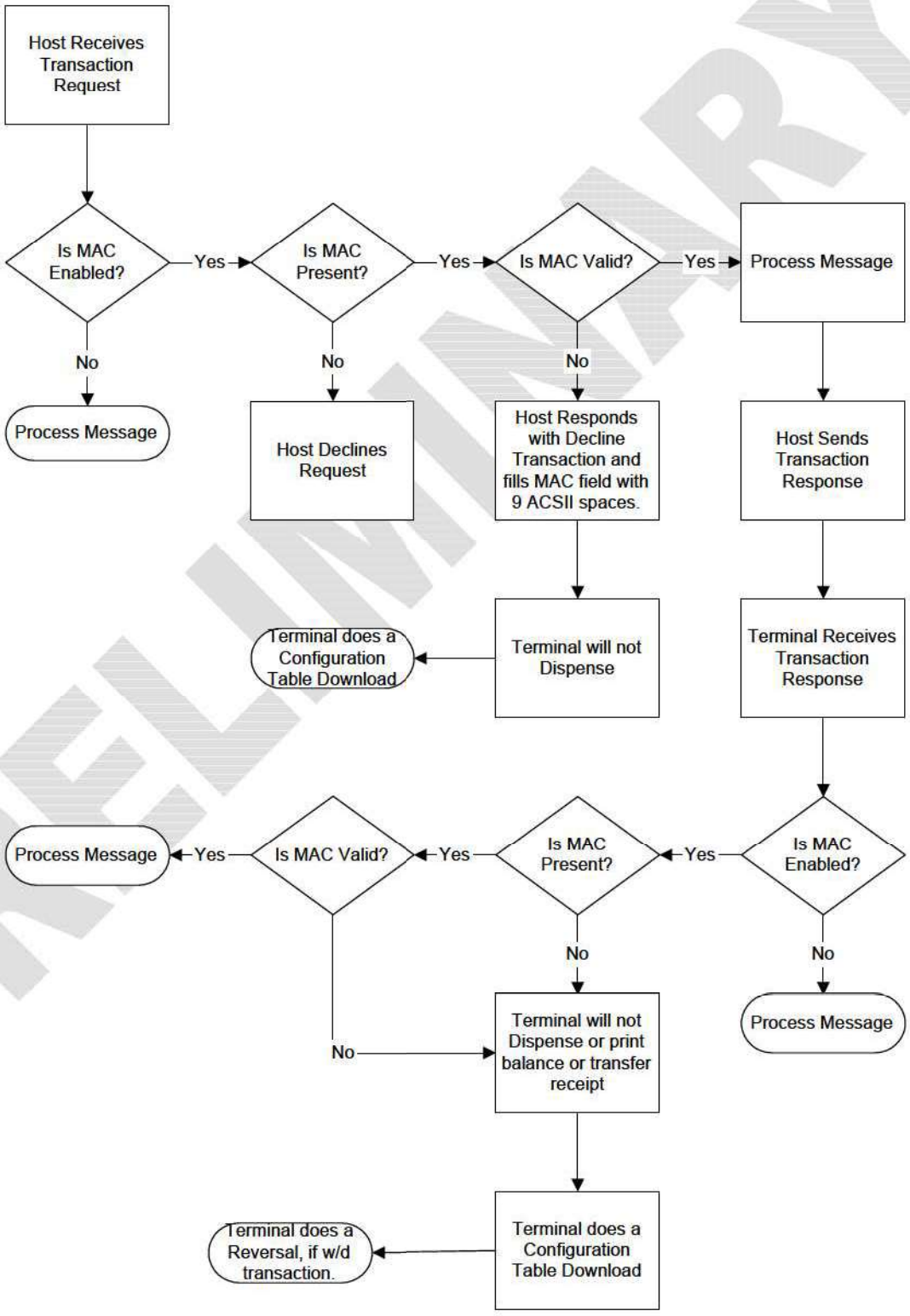


Figure B1 – Triple-DES MAC Algorithm

PRELIMINARY

PRELIMINARY

10.5.1 Message Protocol with MAC Enabled



PRELIMINARY

10.6 Extended Transaction Code Table

CODE	Transaction Type	CODE	Account Type
01	Withdrawal	00	No Account
02	Transfer	01	No Account Selection
03	Balance Inquiry	11	Primary Checking Bank Account
04	Prepaid Telephone	12	Primary Savings Bank Account
05	Bill Payment	15	Primary Credit Card Bank Account
06	Prepaid Electricity	16	Primary Money Market Bank Account
07	Mini Statement	21	Secondary Checking Bank Account
08	Non-Cash Withdrawal	22	Secondary Savings Bank Account
09	Send Money	25	Secondary Credit Card Bank Account
10	Currency Exchange	26	Secondary Money Market Bank Account
11	Crypto Purchase	31	Primary Checking Brokerage Account
		32	Primary Savings Brokerage Account
		35	Primary Credit Card Brokerage Account
		36	Primary Money Market Brokerage Account
		41	Secondary Checking Brokerage Account
		42	Secondary Savings Brokerage Account
		45	Secondary Credit Card Brokerage Account
		46	Secondary Money Market Brokerage Account

Table 113 – Extended Transactions Code Table

To use an extended transaction code string, first place a '99' into the 2-digit transaction code field of the transaction request message. Then construct the extended transaction code string by combining the FID 'l' (ASCII Char - 6Ch) with a transaction type code, a 'from' account type code and a 'to' account type code, consecutively, in the order shown below.

Extended Transaction Code String. – FID 'l' plus Six ASCII numbers. – 'l' + T_1 T_2 T_3 T_4 T_5 T_6

T_1 T_2 - Transaction Type Code

PRELIMINARY

T_3 T_4 - 'From' Account Type Code (Code '00' not allowed)

T_5 T_6 - 'To' Account Type Code

Note: '01' as the from account indicates that the customer did not select an account type. This occurs for South African EMV cards used in South African EMV terminals, because EMV application selection takes the place of account type selection.

The reply transaction codes must match those in the request.

Example:

1.) Transfer from Secondary Money Market Bank Account to Primary Savings Brokerage Account – 'l022632'

† **Preliminary**

10.7 Triple-DES Encryption of the PIN Block

Triple-DES (3DES) encryption/decryption (as specified in ANSI X9.52) is a sequence of DES encryption/decryption operations as shown in diagram 1 and diagram 2 below. 3DES is highly secure and uses two independent 64-bit block keys, *K1* and *K2*. Single-DES is less secure and is achieved when *K1* and *K2* are equal (*K1 = K2)*. *K1* and *K2* are obtained by sending both the Encrypted PIN-Key 1 ('~') and Encrypted PIN-Key 2 ('{') FIDS in a response message. The Encrypted PIN-Key's are decrypted using the Master Key, resulting in *K1* and *K2*. The Clear-Text PIN Block Format and Cipher-Text PIN Format are the same for both Single DES and Triple DES. See 10.4 for a description of the PIN Block Format.

The presence of only the Encrypted PIN-Key 1 ('~') in a response message will cause 3DES capable terminals to store *K2* with the value of *K1*, and set the Encryption Mode Flag (EMF) and the '[' FID to zero (0). The presence of both the Encrypted PIN- Key 1 ('~') and Encrypted PIN-Key 2 ('{') will store *K1* and *K2* with separate values and set the EMF and the '[' FID to two (2), **even for equal Key values**. When the Key values are equal (*K1 = K2)* in a 3DES capable terminal, Single-DES encryption/decryption is achieved by performing the 3DES encryption/decryption scheme. Table 1 summarizes the '[' FID and EMF.

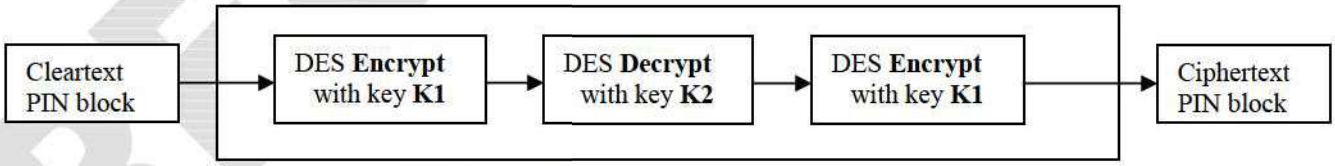


Diagram 1 - 3DES Encryption

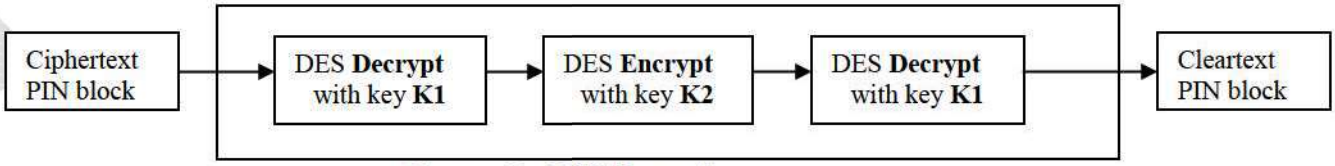


Diagram 2 - 3DES Decryption

PRELIMINARY

	Terminal is 3DES capable		Terminal is not 3DES capable
KEYS received in response message from host	'~' ONLY	'~' & '{'	'~' ONLY
Encryption Mode Flag in Communications header	0	2	0
'[' FID in request message	'[0'	'[2'	NOT SENT

Table 1 – 3DES Capable FID '['

PRELIMINARY

10.8 Encryption Key Sets

For Triton terminals that support multiple host processors, each host processor is assigned one encryption key set. The terminal parameters for each host processor include, but are not limited to, terminal id number, primary and backup host phone numbers, and master and working encryption keys. An encryption key set consists of three pairs of working keys and three pairs of master keys. Each pair of working keys consists of a left and a right block that is used for either Single-DES or Triple-DES encryption of a PIN-Block, MAC message, or DATA message. The PIN, MAC, and DATA master keys are used for encryption and decryption of the associated PIN, MAC, and DATA working keys. The working keys left and right blocks are decrypted as shown in diagram E2 and diagram E3. The Working keys are transmitted to the terminal in encrypted form by using the appropriate FID as a miscellaneous field in a configuration download message. The Master keys are manually entered or electronically injected at the terminal. The FID's are shown in Table E1.

Configuration parameter	FID
PIN Working Key Left Block	'~'
PIN Working Key Right Block	'{'
MAC Working Key Left Block	'm'
MAC Working Key Right Block	'}'
DATA Working Key Left Block	'ue'
DATA Working Key Right Block	'uf'
PIN Master Key Left Block	Manually entered or electronically injected at the terminal.
PIN Master Key Right Block	
MAC Master Key Left Block	
MAC Master Key Right Block	
DATA Master Key Left Block	
DATA Master Key Right Block	

Table 114 E1 – Service parameters.

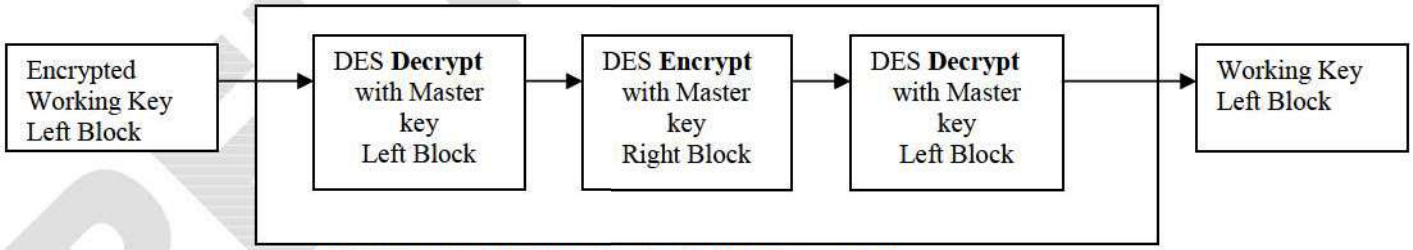
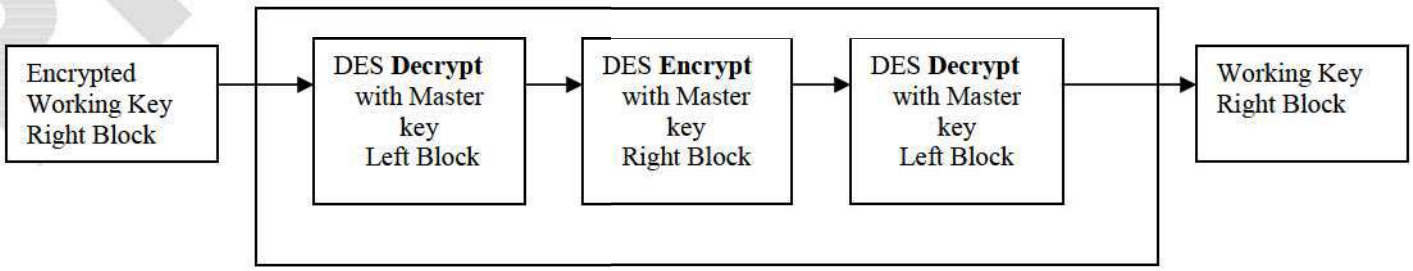


Diagram E2 – Working Key Left Block decryption.



Highly Confidential: Property of Triton Systems – Do Not Disclose

Diagram E3 – Working Key Right Block decryption.

PRELIMINARY

10.9 **ANSI 16-bit CRC**

This section describes the implementation of the CRC-16 (ANSI) routine.

1. Insert a FID 'ub' into a miscellaneous field in the message and initialize the 4 ASCII Characters to ASCII zero (0x30) characters.
2. Calculate the CRC on the message excluding the STX, ETX, and LRC (in dial-up).
3. Convert the 2-byte CRC into 4 ASCII characters and insert these in place of the 4 ASCII zeros (0x30) of the message in the order of MSB to LSB.
4. Upon the receipt of the message, the FID 'ub' will be searched for and the 4 ASCII characters representing the CRC-16 value will be extracted and replaced by 4 ASCII zeros (0x30).
5. Recalculate the CRC on the message excluding the STX, ETX, and LRC (in dial-up).
6. Compare calculated CRC with the extracted CRC from the original message.

PRELIMINARY

Highly Confidential: Property of Triton Systems – Do Not Disclose

10.9.1 ANSI 16-bit CRC

The following is a C code example of the byte by byte method using a lookup table that considers the data and the final CRC being reflected.

```
const unsigned int crc_table[256] =
{
0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
};

/* This function will take the pointer to a message (string) and the length
of the message and will return the computed CRC using the crc_table[]. */

unsigned int crc16 (unsigned char *string, int length )
{
    unsigned int crc;
    crc = 0x0000;    /* initial CRC value 0x0000 */
    while (length--)
    {
        crc = ( crc >> 8 ) ^ crc_table[(crc ^ (int) *string++) & 0xff];
```


Highly Confidential: Property of Triton Systems – Do Not Disclose

```
      /* This performs the modulo-2 division by using the look up table. */
    }
return(crc); /* return the calculated CRC */}
```

PRELIMINARY

10.10 EMV

Mapping of EMV Data Elements to Existing Fields or FIDS

Some of the data elements required by an issuer to process an EMV transaction map to existing fields or FIDS in the TDL legacy communication protocol. All other fields will be encoded as data in the 'ud' FID as Tag-Length-Value elements or data in the 'uh' FID as TSEMVID-Length-Value elements. However, these EMV data elements that have functionally equivalent fields or FIDS will instead continue to be transported as such, instead of being data in the 'ud' or 'uh' FIDs. The following table shows which EMV data elements can be found in which existing fields or FIDS.

10.10.1 EMV-1

EMV Field	Tag	Existing Field or FID
Amount Authorized	9F02	Standard Amount 1 field or 'e' FID data
Amount Other	9F03	No field or FID. Always implied to be 0.
Enciphered PIN	99	PIN Block
Track 2 Equiv. Data	57	Track 2 field
Transaction Seq. Counter	9F41	Sequence Number field (only if not present in the 'ud' FID)†Sequence Number field
Terminal ID	9F1C	Terminal ID field

Table 115 EMV-1.1.1

† Please note that if the EMV Transaction Sequence Counter tag (9F41) appears in the 'ud' FID, the value found in that FID is the definitive tag 9F41. In such a case, the sequence number found in the TDL Legacy Sequence Number message field is not the EMV Transaction Sequence Counter tag (9F41). Instead, the EMV Transaction Sequence Counter holds a separate, EMV-compliant sequence number, incremented only for EMV transactions, while the TDL Legacy Sequence Number field holds the usual sequence number, incremented for every transaction. These two sequence counters will almost always have different values.

Please note that if EMV data is sent from the terminal in a transaction request message in the 'ud' FID, the transaction response message must contain a 'ud' FID with, at minimum, tag 8A for the transaction to be approved.

Balance Inquiry and PIN Change Transactions

Balance Inquiry and PIN Change transactions are considered non-EMV transactions which use EMV functionality. EMV best practices recommend that these transactions be completed with an AAC, which for non-EMV transactions simply means completion and not a decline. Triton relies on a special authorization code from the host for this. The host should return authorization code '085' (see 10.2.18) to inform the terminal that the transaction was approved but the terminal should complete the chip transaction with an AAC. If a standard approval code '000' is sent, the terminal will rely on the chip for the final approval, which could be declined.

Encoding of Other EMV Data Elements

Other EMV data elements present in a message will be encoded into the 'ud' or 'uh' FIDs. Data elements specified in the EMV standard that are assigned unique tags will be packed into the 'ud' FID (Encoded EMV data block). Data elements specified in the EMV standard but not assigned a tag will be packed into the 'uh' FID (Encoded EMV untagged-data block).

Encoded EMV data block ('ud' FID)

PRELIMINARY

The data that follows the 'ud' FID is of variable length and contains zero or more data elements encoded according to Basic Encoding Rules—Tag-Length-Value (BER-TLV). See ISO/IEC 8825:1990, ISO/IEC 7816-4:1995 Annex D, and EMV 2000 Book 3 Annex B for a more complete and official description of this format. Information presented here is a summary of important points from these documents and is included for convenience only.

Encoded EMV untagged-data block ('uh' FID)

The data that follows the 'uh' FID is of variable length and contains zero or more data elements encoded according to a scheme identical to the Basic Encoding Rules—Tag-Length-Value (BER-TLV) scheme. However, the identifiers (TSEMVIDs) used in this scheme (instead of tags) are proprietary to Triton's communication specification. **Note**: It is important for any host processing this block to extract these data elements before sending them anywhere. The TSEMVIDs are not specified in the EMV standard and are not meaningful anywhere outside of the TDL legacy communication scheme. See table EMV-1.1.2 for a listing of these TSEMVIDs. TSEMVIDs are encoded using the same rules that cover the encoding of the tag field in BER-TLV.

10.10.2 EMV-2

Assigned TSEMVID	*Data Element in EMV Standard*
DF01	*Certification Authority Public Key Check Sum*
DF02	*Certification Authority Public Key Exponent*
DF03	*Certification Authority Public Key Modulus*
DF04	*Default Dynamic Data Authentication Data Object List (DDOL)*
DF05	*Issuer Script Results*
DF06	*Last Online ATC Register*
DF07	*Maximum Target Percentage to be used for Biased Random Selection*
DF08	*Merchant Name and Location*
DF09	*Message Type*
DF0A	*Personal Identification Number (PIN) Pad Secret Key*
DF0B	*Target Percentage to be used for Random Selection*
DF0C	*Terminal Action Code – Default*
DF0D	*Terminal Action Code – Denial*
DF0E	*Terminal Action Code – Online*
DF0F	*Threshold Value for Biased Random Selection*
DF10	*Transaction Reference Currency Code*
DF11	*Default Transaction Certificate Data Object List (TDOL)*

Table 116 EMV-1.1.2

Basic Format

BER-TLV data elements are each represented with three fields, a tag, a length, and a value. The tag field uniquely identifies the data element. The length field specifies the length in bytes of the value field. The value field holds the actual data element bytes. The following description of these fields apply *before* encoding as 7-bit ASCII hex digits for packing into a Triton communication message.

Tag

For the purposes of this specification, tag fields will always be one or two bytes, although they are not restricted to such by ISO/IEC 8825:1990. Values 0x00 and 0xFF are not considered valid tags, as per ISO/IEC 7816-4. Tags code a Class, Type, and Number; see table B-1 in Annex B of EMV 2000 Book 3 for the coding scheme of a tag. As shown in the table, if the least significant 5 bits of the first byte of the tag are 1, the tag is a two-byte tag. Otherwise, it is a one-byte tag.

Defined tags also carry with them an implied format of the encapsulated data element. See EMV 2000 Book 3 Annex A for a table of defined tags and associated data formats defined by EMV, as

well as documentation for individual payment systems that may define their own, issuer-specific tags.

Length
For the purposes of this specification, length fields will always be one, two, or three bytes, although they are not restricted to such by ISO/IEC 8825:1990. The length field specifies the length in bytes of the following value field. Lengths of 0x00 are allowed and indicate the absence of a value field.

If the most significant bit of the most significant byte of the length field is 0, the length field is one byte and the remaining seven bits encode the length value (0-127). If the most significant bit of the most significant byte is 1, the length field is more than one byte and the remaining seven bits encode the number of bytes that follow the most significant in the length field. The subsequent bytes together represent an integer that specifies the length in bytes of the value field.

Value
The value field consists of zero or more data bytes as specified in number by the length field. These bytes represent the actual data of the data element.

PRELIMINARY

Packing of TLV Encoded Data Elements in the 'ud' FID

TLV encoded data elements will be packed into the data of the 'ud' FID with no separation. All bytes of TLV encoded data elements, when packed into the data of the 'ud' FID, will be represented as ASCII encoded hexadecimal digits, requiring two digits to represent a byte, effectively doubling the size of the data. This is necessary to keep message formatting characters, such as the field separator (0x1C) and end of text (0x03) from being included in the 'ud' FID data. The length field of the TLV encoded data elements will reflect the size of the value field before it is encoded into ASCII hex digits so care must be taken in parsing the elements. Several examples follow for clarity.

Packing of Untagged Data Elements in the 'uh' FID

Untagged data elements defined in the EMV standard, encoded using a TSEMVID, length, and value, will be packed into the data of the 'uh' FID with no separation. All bytes of the encoded data elements, when packed into the data of the 'uh' FID, will be represented as ASCII encoded hexadecimal digits, requiring two digits to represent a byte, effectively doubling the size of the data. This is necessary to keep message formatting characters, such as the field separator (0x1C) and end of text (0x03) from being included in the 'uh' FID data. The length field of the encoded data elements will reflect the size of the value field before it is encoded into ASCII hex digits so care must be taken in parsing the elements.

EMV Example 1

A basic example with one-byte lengths is presented here.

Tag	Length	Data Bytes										
82	2	0x1C	0x00									
9F26	8	0xF4	0x7A	0x21	0xE3	0x9C	0xAF	0x5D	0x07			

Table 117 EMV-2.1.1

Resulting Bytes in the 'ud' FID Data								
0x38 ('8')	0x32 ('2')	0x30 ('0')	0x32 ('2')	0x31 ('1')	0x67 ('C')	0x30 ('0')	0x30 ('0')	0x39 ('9')
0x46 ('F')	0x32 ('2')	0x36 ('6')	0x30 ('0')	0x38 ('8')	0x46 ('F')	0x34 ('4')	0x37 ('7')	0x41 ('A')
0x32 ('2')	0x31 ('1')	0x45 ('E')	0x33 ('3')	0x39 ('9')	0x43 ('C')	0x41 ('A')	0x46 ('F')	0x35 ('5')
0x44 ('D')	0x30 ('0')	0x37 ('7')						

Table 118 EMV-2.1.2

PRELIMINARY

EMV Example 2

A more complicated example with a three-byte length is shown here. Note the encoding of the length 289 as 0x820121, forming three bytes 0x82, 0x01, and 0x21, yielding the eight hex digits 0x38, 0x32, 0x30, 0x31, 0x32, and 0x31. The first and most significant byte, in binary, is 10000010_b, meaning that this length is a multi-byte length field (most significant bit is 1) and there are two bytes following in the length field (remaining bits code the number 2). The remaining bytes, 0x01, and 0x21 code the number of bytes in the value field by forming the number 0x121 in hex, or 289 in decimal.

Tag	Length	Data Bytes									
91	10	0x84	0xB6	0xDE	0xAD	0xF0	0x0F	0x0A	0x11	0x9C	0x00
8E	289	Data bytes not shown…									

Table 119 EMV-2.2.1

Resulting Bytes in the 'ud' FID Data								
0x39 ('9')	0x31 ('1')	0x31 ('0')	0x30 ('A')	0x38 ('8')	0x34 ('4')	0x42 ('B')	0x36 ('6')	0x44 ('D')
0x45 ('E')	0x41 ('A')	0x44 ('D')	0x46 ('F')	0x30 ('0')	0x30 ('0')	0x46 ('F')	0x30 ('0')	0x41 ('A')
0x31 ('1')	0x31 ('1')	0x39 ('9')	0x67 ('C')	0x30 ('0')	0x30 ('0')	0x38 ('8')	0x45 ('E')	0x38 ('8')
0x32 ('2')	0x30 ('0')	0x31 ('1')	0x32 ('2')	0x31 ('1')	Data bytes not shown…			

Table 120EMV-2.2.2

PRELIMINARY

EMV Elements to be Included in Messages

While, theoretically, any EMV data element may be sent or received to process an EMV transaction, hosts should typically expect certain common fields to be included in transaction requests, transaction responses, and transaction reversals. These common fields are listed here. This is by no means an exhaustive list of all possible EMV data elements that can appear in messages, so hosts should be prepared for any defined EMV tag. Any tag not available to the terminal may not be sent, so hosts should be prepared to handle missing data.

10.10.3 EMV-3

Field Description	Tag	TSEMVID	Existing Field or FID	Notes
Amount Authorized	-	-	Amount 1 or 'e' FID	Use Amount 1 field if not using extended amounts, 'e' FID otherwise.
Application ID	9F06 or 4F	-	-	
App Interchange Profile	82	-	-	
App Transaction Counter	9F36	-	-	
Cryptogram	9F26	-	-	
Cryptogram Info Data	9F27	-	-	
Enciphered PIN	-	-	PIN Block	
Issuer App Data	9F10	-	-	
Issuer Script Identifier	9F18	-	-	Only if previous trans. had a script.
Issuer Script Results	-	DF05	-	Only if previous trans. had a script.
Terminal Verification Results	95	-	-	
Unpredictable #	9F37	-	-	
App PAN Seq. #	5F34	-	-	
Terminal ID	-	-	Term. ID Field	
Terminal Country Code	9F1A	-	-	
Track 2 Equiv. Data	-	-	Track 2 Field	
Transaction Currency Code	5F2A	-	-	
Transaction Date	9A	-	-	
Transaction Type	9C	-	-	
Transaction Seq. Counter	9F41	-	or Seq. Number Field	If the 9F41 tag appears in the 'ud' FID, it contains a separate, EMV-compliant sequence number incremented only for EMV transactions. In this case, the TDL Legacy Sequence Number field contains the usual sequence number, incremented for every transaction.
IFD Serial Number	9F1E	-	-	
Terminal Capabilities Profile	9F33	-	-	
Terminal Type	9F35	-	-	
CVM List	8E	-	-	
CVM Results	9F34	-	-	
POS Entry Mode	9F39	-	-	
Transaction Status Information	9B	-	-	
Transaction Category Code	9F53	-	-	This tag is sent if the terminal is processing a MasterCard Application

Table 121 EMV-3

PRELIMINARY

10.10.4 **EMV-4**

Field Description	Tag	TSEMVID	Existing Field or FID	Notes
Authorization Response Code	8A	-	-	This tag is required for a successful transaction. A value of '00' indicates a request for an approval from the ICC, while any other value indicates a request for a decline from the ICC.
Authorization Code	89	-	-	
Issuer Authentication Data	91	-	-	
Issuer Script	71	-	-	
Issuer Script	72	-	-	
Issuer Script Identifier	9F18	-	-	

Table 122 EMV-4

10.10.5 **EMV-5**

Field Description	Tag	TSEMVID	Existing Field or FID	Notes
Application ID	9F06 or 4F	-	-	
App Interchange Profile	82	-	-	
App Transaction Counter	9F36	-	-	
Issuer App Data	9F10	-	-	
Terminal Verification Results	95	-	-	
Amount Authorized	-	-	Amount 1 or 'e' FID	Use Amount 1 field if not using extended amounts, 'e' FID otherwise.
App PAN Seq. #	5F34	-	-	
Terminal ID	-	-	Term. ID Field	
Track 2 Equiv. Data	-	-	Track 2 Field	
Transaction Type	9C	-	-	
Transaction Seq. Counter	9F41	-	or Seq. Number Field	If the 9F41 tag appears in the 'ud' FID, it contains a separate, EMV-compliant sequence number incremented only for EMV transactions. In this case, the TDL Legacy Sequence Number field contains the usual sequence number, incremented for every transaction.
Issuer Script Identifier	9F18	-	-	
Issuer Script Results	-	DF05	-	
Transaction Status Information	9B	-	-	

Table 123 EMV-5

PRELIMINARY

Issuer Script Results

The results of Issuer Scripts, after execution, will be sent at one of two possible times. If the transaction that contained the issuer scripts requires a reversal, during that reversal, the results will be sent (TSEMVID DF05). However, if that transaction did not require a reversal, the script results will be sent during the next transaction request message.

Issuer script results are encoded as detailed in the EMV specification, book 4, Annex A5.

PRELIMINARY

POS Entry Mode Flag

The POS entry mode flag varies based upon each published specification. The following table EMV-6 describes the current Triton Terminal implementation verses several specifications. It is up to the acquirer to translate the values sent by the terminal into the appropriate value for each issuer or network. Additionally, the latest ISO standards specify that the POS entry mode flag be three bytes instead of two. This additional byte indicates if a terminal supports PIN entry. This byte is implied to be a value of 1 for all Triton Terminals.

10.10.6 EMV-6

Specification	ICC	Magstripe Only	Fallback 1*	Fallback 2*
Current Triton Terminals	05	N/A	92	92
VISA	Not specified	90	91†	92†
LINK	05	02/06‡	90	90
MasterCard	Not specified	Not specified	80x	80x

Table 124 EMV-6

* Fallback 1: Code indicates transaction is a fallback. Last transaction was successful chip or magstripe only transaction.

* Fallback 2: Code indicates transaction is a fallback. Last transaction was an unsuccessful chip transaction.

†91/92: VISA says to use these values, but their specification goes on to say that the acquirer must not ever send the value 0x92 or 0x91 to the issuer, indicating that the acquirer must translate it to 0x90. (To quote—"The POS Entry Mode Codes of '91' '92' are not valid in messages from the acquirer to VisaNet and, if used from the terminal to the acquirer, shall be converted, by the acquirer, to '02' '90' as appropriate.)

‡02/06: LINK LIS5 specifies the code for magstripe read as 0x02 for non-VISA transactions, 0x06 for VISA transactions, contradicting how VISA says the acquirer should translate this code.

PIN Management Services Transactions (PMS)

PIN management Services Transaction include both PIN Change and PIN Unlock (sometimes called PIN Unblock) transactions. The Triton Terminal implementation uses the PIN Change transaction code for all PMS transactions. PIN Unlock transactions will be differianciated from PIN change transaction by the lack of the @ fid containing a new PIN.

PRELIMINARY

10.11 Top Up

E*Trade, Triton Waves, United Kingdom, and South Africa versions of terminal software support mobile phone top-up transactions. Each version requires a different mix of miscellaneous FIDs for proper operation. Table H1 shows the proper FIDs to use for each version. See the Valid Miscellaneous Field ID Table for the proper message type to place the FID into.

10.11.1 Top Up miscellaneous FIDs to use for certain version of terminal software

FID and description	E*Trade Transaction Code 44		Triton Waves Transaction Code 44		United Kingdom Transaction Code 44		South Africa Transaction Code 99 Use Extended Transaction Code String. – FID '1'	
	Request	Response	Request	Response	Request	Response	Request	Response
xa Telco-product Serial Number						X		
xb mobile phone number	X		X	X activation phone number	X		X	
xd general taxes		X	X	X				
xe top-up transaction totals and amounts	X	X	X	X				
xf contact phone number		X		X		X		X
xg service provider name		X	X	X	X	X	X	
xh Expiration date		X						
xn Actual amount debited from the customer's account								X
xp voucher PIN number				X				X
xq service provider name and fast cash amounts				X		X and service provider ID		
xt general tax rate				X				
xu service provider ID					X			
xv service provider transaction totals and amounts					X	X		

PRELIMINARY

Table 125 – Top Up Miscellaneous FIDs to use for certain version of terminal software

PRELIMINARY

11 Remote Key Transport Extensions

11.1 Introduction

This document is a supplemental document for implementing remote master key transport within the TDL legacy message format.

11.2 Message Format Summary

11.2.1 New FIDs

new 'r' FID subset will be used for all RKT elements.

'r' ASCII Char – 75h.	Two character FIDs. The first character after the FID 'r' is another character that defines which data element is to be transmitted. The table below describes each data element. A Field Separator must follow each field.	
	'a'	This FID is sent in a normal transaction response message to tell the terminal to go temporarily out of service after the call and initiate a RKT sequence.
	'b'	Sent with no parameters in a RKT request message to request the host's certificate. When this FID is in a RKT response message, the data will be the host's certificate. The certificate will be base 64 encoded as specified in RFC 3548.
	'c'	Thumbprint of host certificate, 20 binary bytes. Generated by the EPP and sent in a RKT request message. This field will be base 64 encoded as specified in RFC 3548.
	'd'	EPP Certificate – sent with a RKT request message. The certificate will be base 64 encoded as specified in RFC 3548. There are two EPP certificates, but only one certificate will be sent at a time due to size restrictions.
	'e'	Random number, 8 binary bytes. Generated by the EPP and sent in a RKT request message. This number will be base 64 encoded as specified in RFC 3548.
	'f'	Encrypted key block. Generated and signed by the host. Sent in a RKT response message. This block will be base 64 encoded as specified in RFC 3548
	'g'	Verification message. Generated and signed by the EPP. Sent in a RKT request message. This field will be base 64 encoded as specified in RFC 3548.
	'h'	Error code. Optional error code can follow.
	'i'	This FID indicates to the terminal that RKT is enabled on this particular terminal. It will be included in a configuration download request message if RKT is enabled.

Table 126 – New FIDs

11.2.2 New Timeout

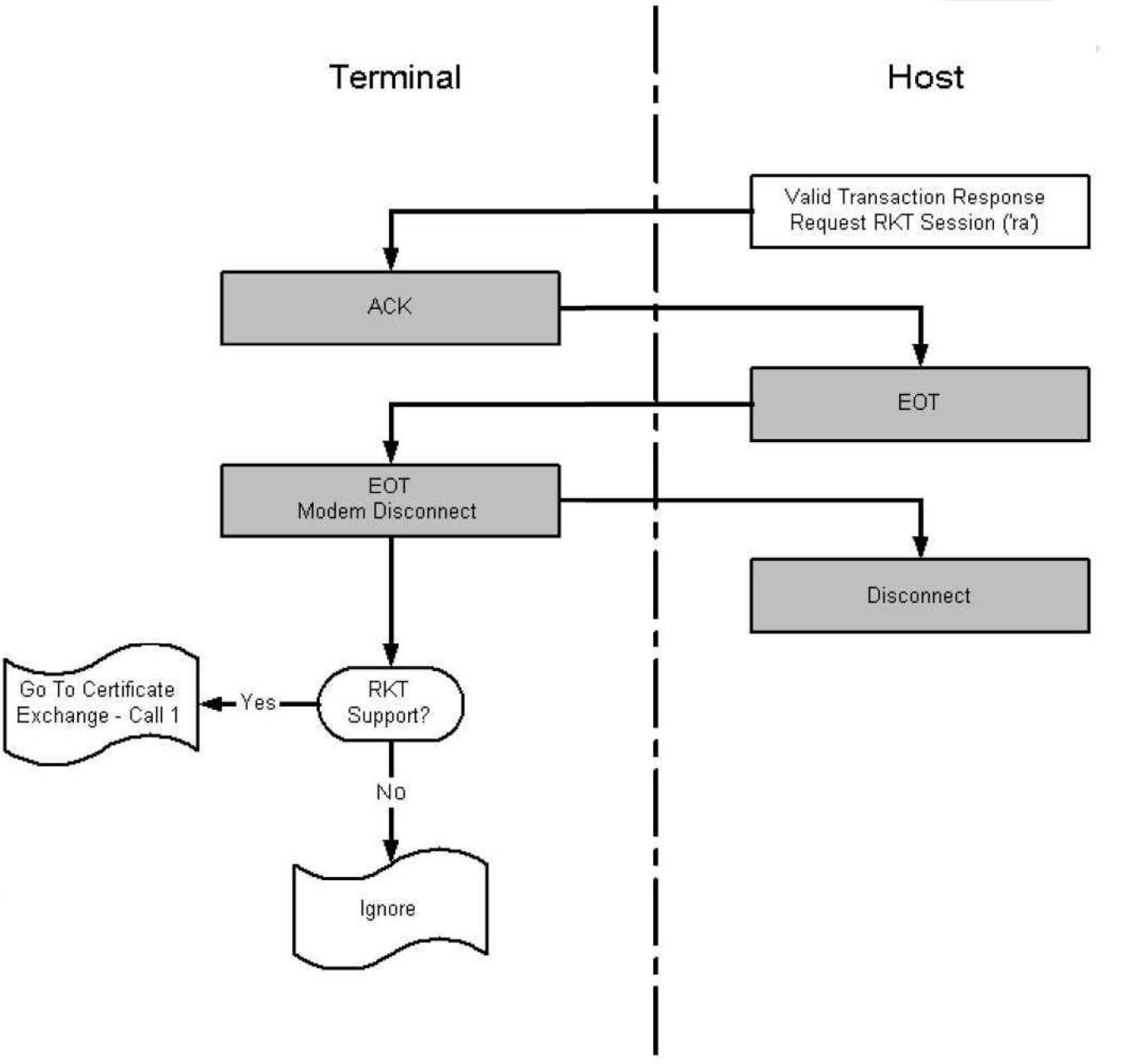
The timeouts described in the section "Terminal Does Not Acknowledge Response From Host", see 10.3.7, will be 20 seconds instead of 3 for the RKT process.

PRELIMINARY

11.3 RKT Message Flows

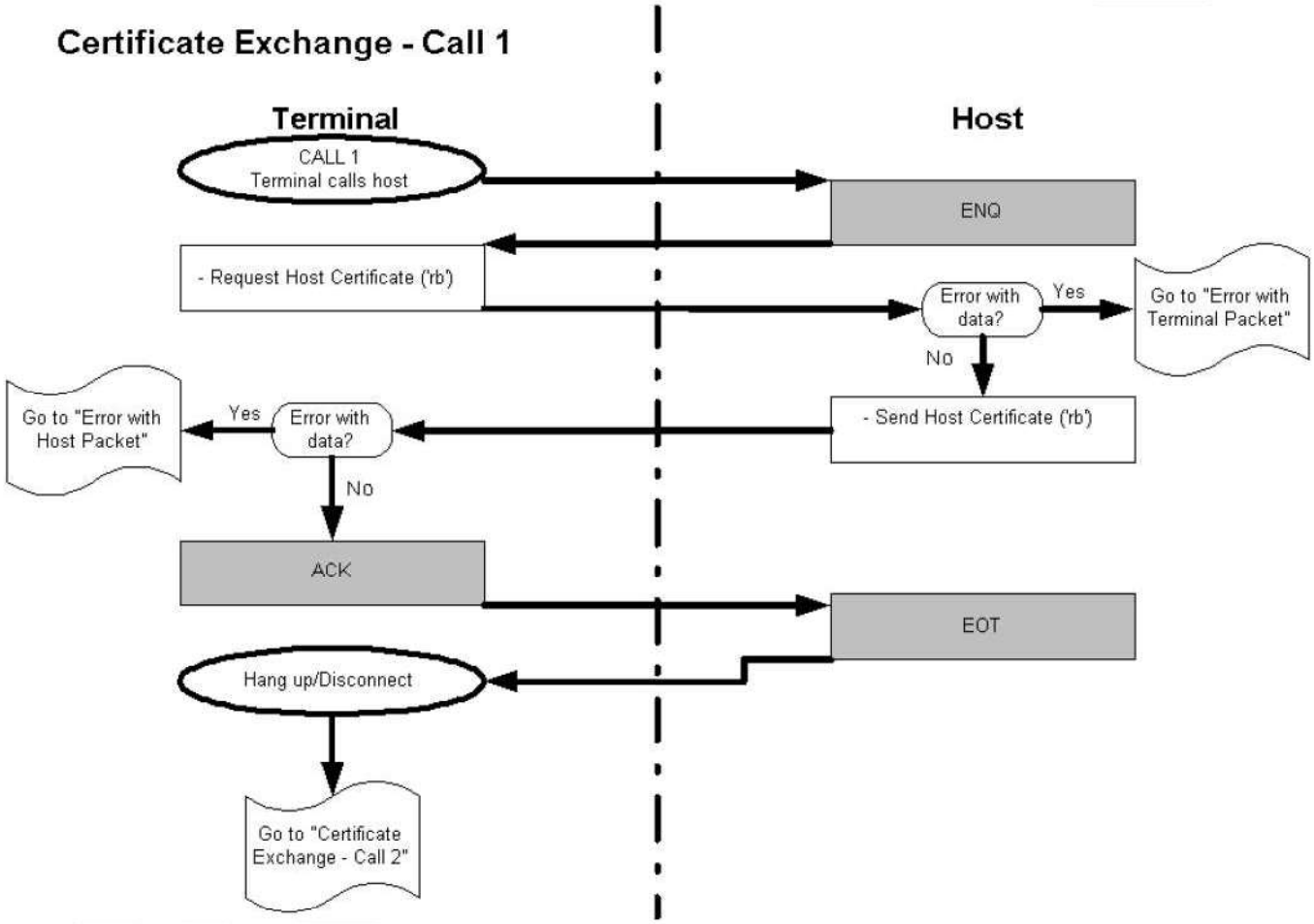
Note: In these flowcharts, the states in grey can be skipped during TCP/IP sessions, but will remain for dial-up sessions.

11.3.1 Initiate RKT Session from Host

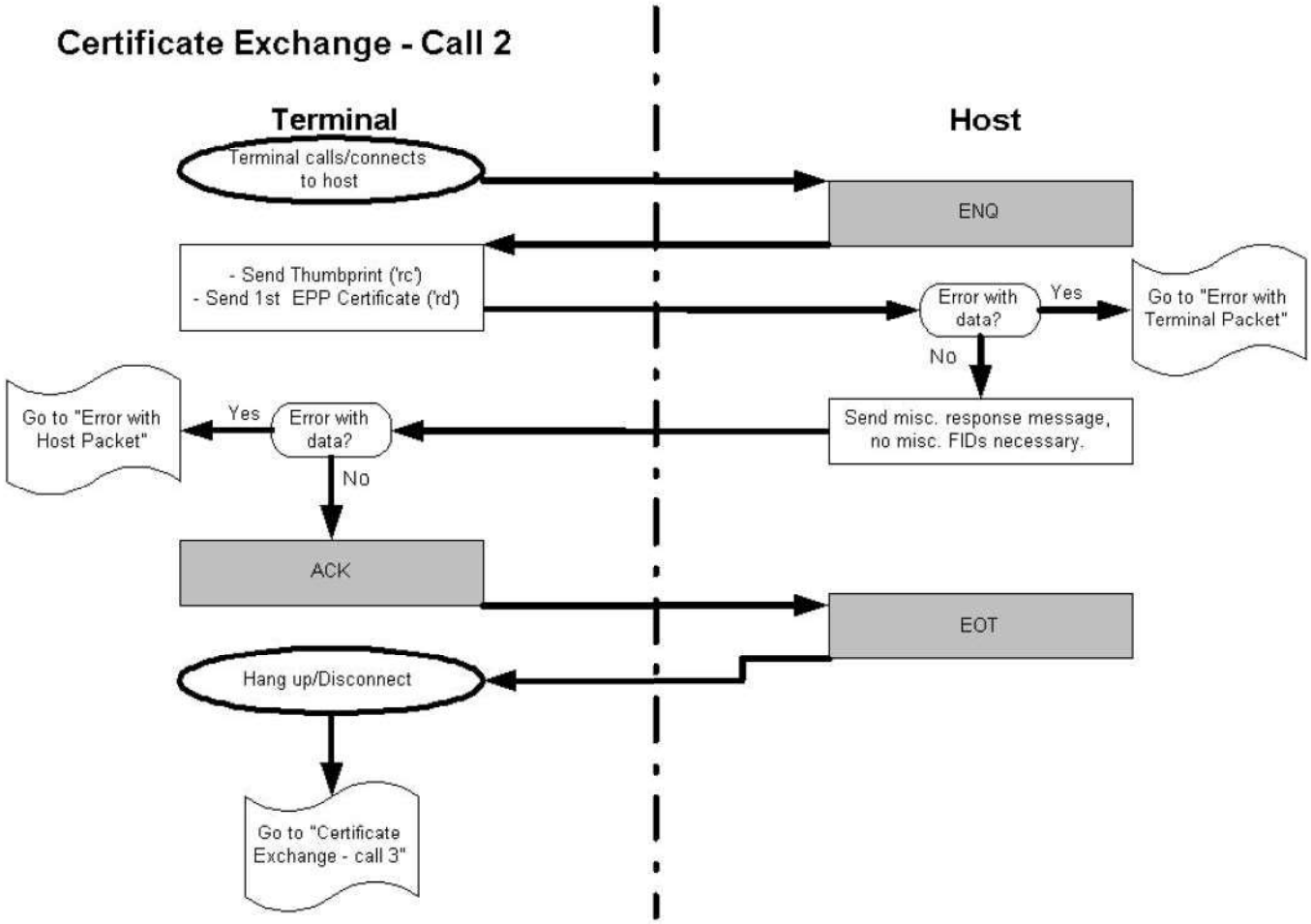


Highly Confidential: Property of Triton Systems – Do Not Disclose

11.3.2 Certificate Exchange Flow



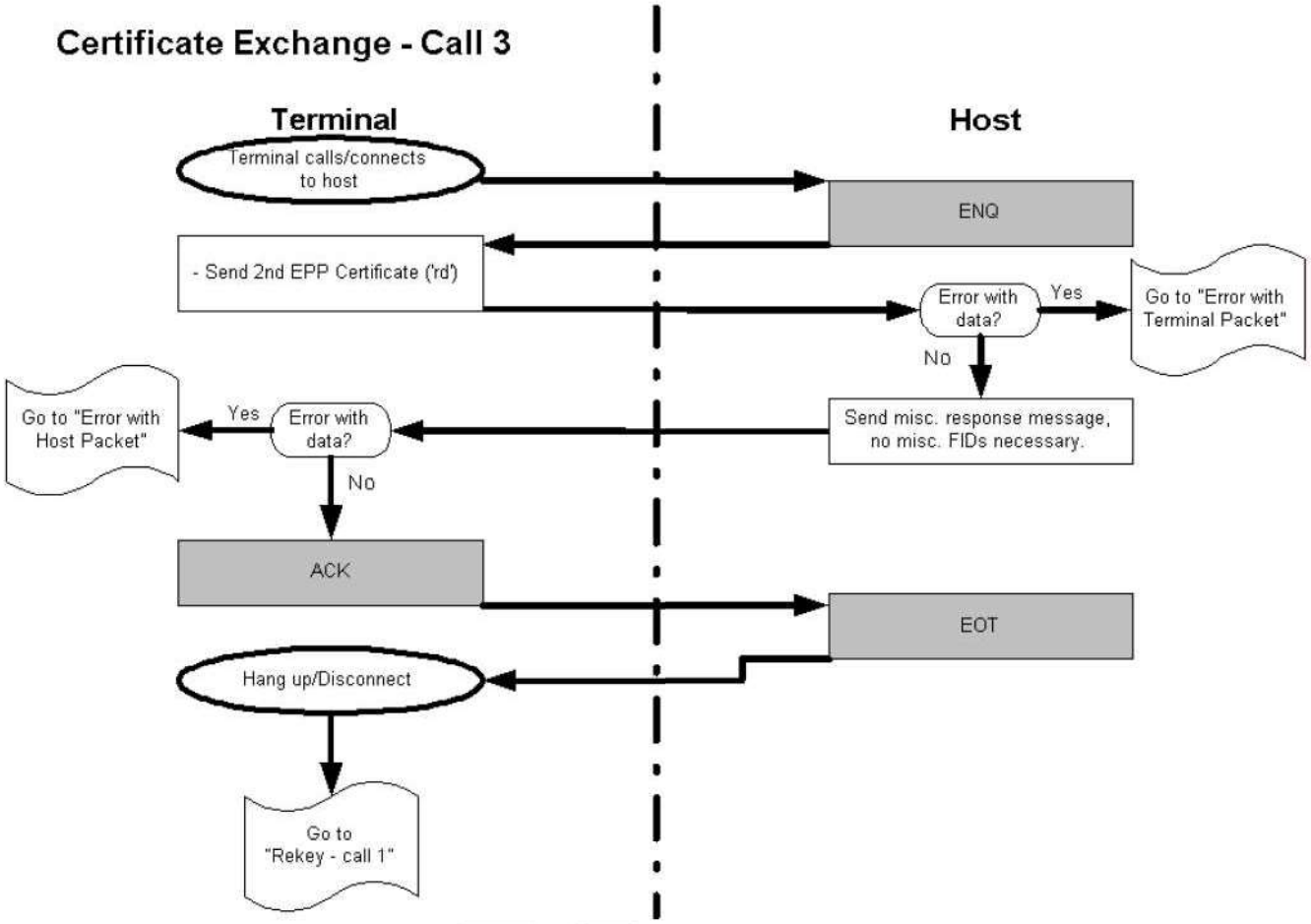
Highly Confidential: Property of Triton Systems – Do Not Disclose



PRELI

Highly Confidential: Property of Triton Systems – Do Not Disclose

Certificate Exchange - Call 3

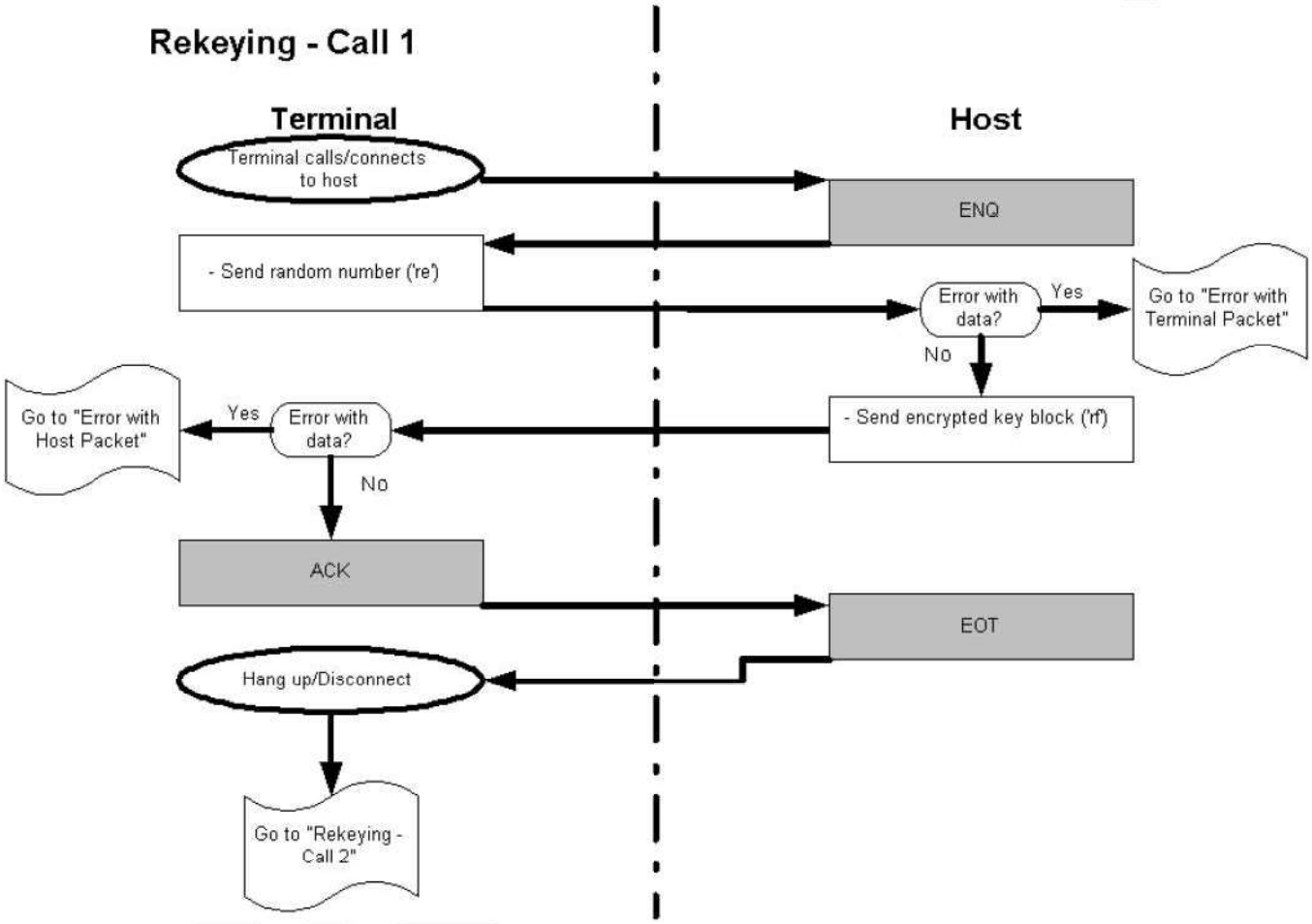


PRELI

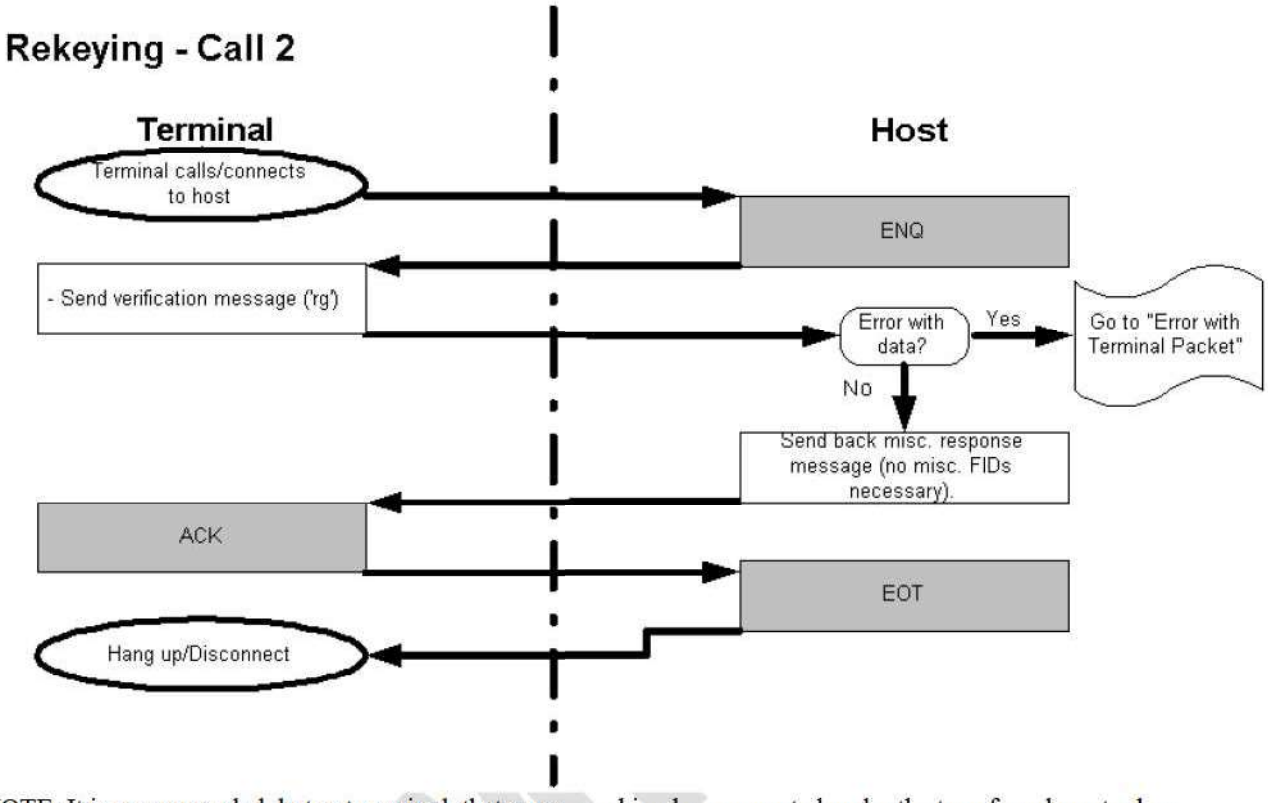
Highly Confidential: Property of Triton Systems – Do Not Disclose

11.3.3 Re-keying Flow

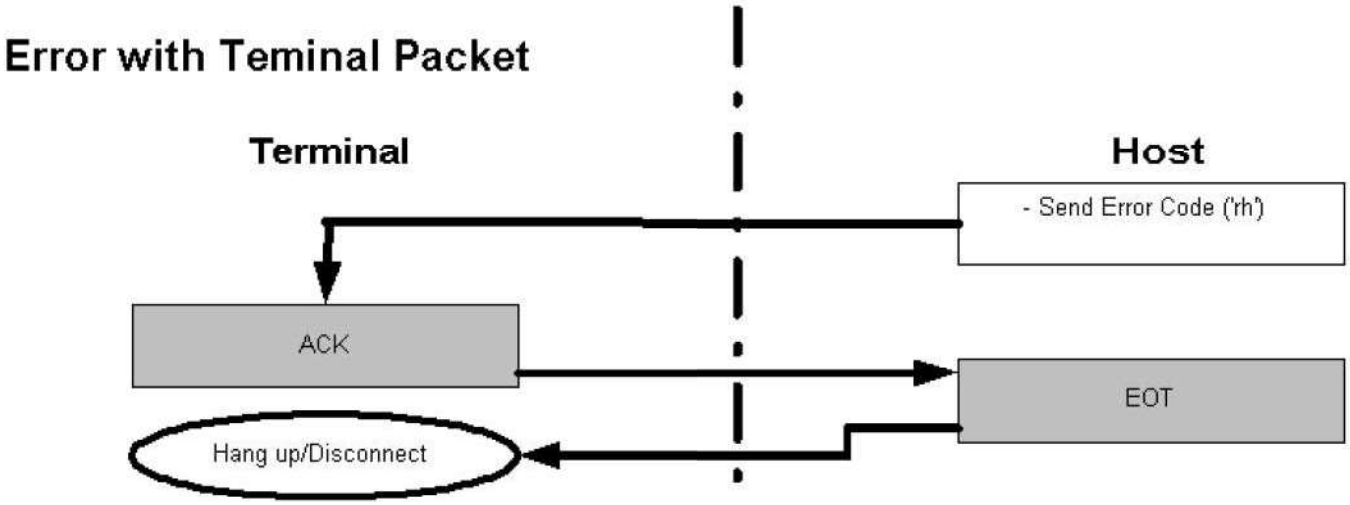
The master key will be activated as soon as the encrypted key block is received by the terminal. If for some reason communications are interrupted, before the verification message is sent, the new master key will still be activated.



PRE

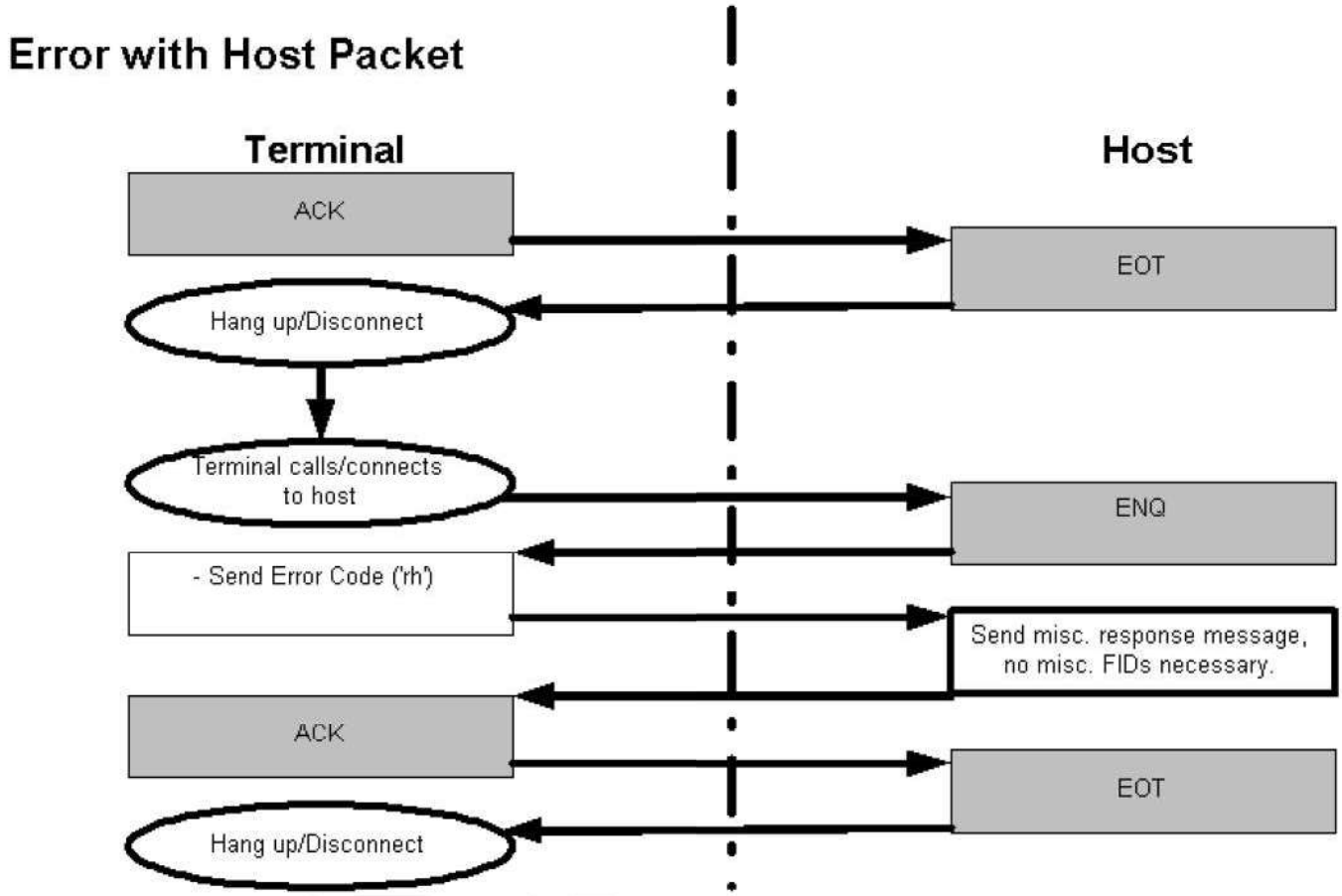


NOTE: It is recommended, but not required, that a new working key encrypted under the transferred master key is sent in the response to the verification message.



Highly Confidential: Property of Triton Systems – Do Not Disclose

11.3.4 Error with RKT Data Packets



PRELIMINARY

Highly Confidential: Property of Triton Systems – Do Not Disclose

PRELIMINARY

12 Triton Terminal Currency Exchange Specification

12.1 Introduction

This document is a supplemental document for implementing currency exchange within TDL legacy message format. This is based on the following requirements:

1. Exchange rate(s) will be sent by the host in a transaction message response or configuration download response.
2. The terminal should handle all currency conversion locally and send the converted amount and current exchange rate to the host for each transaction. Note that these values may not be accurate depending on when the terminal last communicated to the host.
3. The terminal should be able to handle up to 4 different types of currency dispensing. The base or local currency type must be the same for all.
4. The communications between the host and ATM should use standard ISO 4217 currency codes to indicate currency types (see http://www.iso.ch/iso/en/prods-services/popstds/currencycodeslist.html).
5. The terminal should send all currency exchange transaction information to the host in the host totals download request.
6. The terminal should be able to send appropriate transaction reversal information for the case where the entire amount could not be dispensed.

PRELIMINARY

12.2 Message Format Summary

12.2.1 Transaction Codes

Currency exchange transactions will use transaction code 99 to indicate use of an extended transaction code in 10.6. The following extended transaction code will be added:

- 10 – Currency Exchange Transaction

12.2.2 Authorization Codes

The following authorization code will be added:

- 037 – Invalid Exchange Rate. This indicates that the host did not approve the exchange rate sent by the terminal in a transaction request. The host should send a valid exchange rate in the transaction response.

12.2.3 Miscellaneous FIDs

New 'a' FID subset: New set of FIDs to handle all currency exchange information. Each FID will be sent as 2 character FID beginning with 'a'.

The term "local" refers to the currency used as the **base** currency for the terminal and host. This designates the type of currency for amounts specified in a standard transaction (i.e., withdrawal) request.

The term "foreign" refers to the currency used as the **requested** currency type for a currency exchange transaction.

PRELIMINARY

12.3 Currency Exchange FID Overview

FID	Name	Format	Description
aa	Exchange Rate	VAR ASCII Char	3 digit numeric local currency code (ISO 4217) 1 digit currency code count (0-4) Repeat for currency code count: - 3 digit foreign currency code (ISO 4217) - 11 digit exchange rate (embedded decimal, padded with zeros to left or right) Example for an exchange rate of 1 US dollar to 1.23456789 pounds: 84018261.234567890 This FID is sent from terminal to host for a currency exchange transaction to indicate the exchange rate applied. This FID is sent from terminal to host in a configuration download request to specify which currency conversions the terminal can support (based on cassette configuration). If the terminal has an existing exchange rate for that currency, it will send that rate. If the terminal has not received a rate for a given currency, zeros will be sent for the exchange rate. This FID is sent from host to terminal in a transaction response or configuration download response for each currency exchange type supported. The local currency code must match the configuration for the terminal. Exchange rates sent with a different local currency type will be ignored. To disable all currency exchange transactions, the host would send an 'aa' FID with a currency code count of 0. Note that receiving an 'aa' FID from the host in a transaction or configuration download response will **override** any previous 'aa' FID sent. Only the last set will be used.
ab	Requested Amount (Foreign)	13 ASCII Char	Requested amount with embedded decimal and zero filled to the left in foreign currency for the specified exchange rate (GBP for the above example): 1234567890.12 The local amount will be specified in amount1 or 'e' FID of the transaction request.
ac	Currency Exchange	39 ASCII Char	Totals for the currency exchange transactions since

PRELIMINARY

	Close Totals		last close. Sent from the terminal (if send terminal totals is enabled) and from the host. Totals are zero filled to the left. Format: 3 digit numeric foreign currency code 4 digit number of transactions 16 digit settlement total (local/converted amount) Example: 826002012345678901 23.45 If send terminal totals are enabled, the terminal will send 'ac' FIDs for each currency type active since the last close.
ad	Dispensed Amount (Foreign)	13 ASCII Char	Actual dispensed amount (for reversals) with embedded decimal and zero filled to the left in foreign currency for the specified exchange rate (GBP for the above example): 1234567890.12
ae	Commission Amount	13 ASCII Char	The amount of the commission charged for the transaction to be presented to the customer for verification prior to dispensing currency. The value will contain an embedded decimal place and zero filled to the left. This value is in local currency. This FID is sent from host to terminal in a transaction response or configuration download to be used on all subsequent transactions. This FID is sent from terminal to host in a transaction request message and a reversal request message to indicate the exchange rate that was presented to the customer.
af	Status Monitoring	12 ASCII Char	3 digit currency code (numeric) for each cassette in this order: • Casscttc A • Cassette B • Cassette C • Cassette D Space filled if cassette not configured. Example of cassette A containing US dollars and cassette B containing Euro notes: "840978 " This FID will be sent from terminal to the host if status monitoring is enabled.

Table 127 – Currency Exchange FID Overview

PRELIMINARY

PRELIMINARY

13 Triton Terminal Prosa Fee Extensions

13.1 Introduction

This document is a supplemental document for implementing Prosa Fee Extensions within TDL legacy message format. This is based on the following requirements:

7. Prosa fee, bank name, and IVA tax information will be sent by the host in a transaction message response or configuration download response.
8. The terminal should handle all Prosa fee FID exchanges making the appropriate notifications and journal entries as needed for making additional costs assessed to the customer clear and easy to track.
9. The terminal should be able to include any such fees in reversal requests sent by the terminal due to inability to dispense or by reason of user cancellation.

13.2 Message Format Summary

13.2.1 Miscellaneous FIDs

New 2 character FID: first character is lower case 'q' (0x71) followed by a second character, lower case 'b' (0x62). This new FID will be defined to encode all the information necessary to describe any fees, taxes and bank name information required for the transaction.

PRELIMINARY

13.3 Prosa Fee FID Overview

FID	Name	Format	Description
qb	Prosa Fee	VAR ASCII Char	The Prosa Fee FID is constructed as a single FID capable of encoding a variety of data elements associated with Prosa-specific transaction fee data. There may be more than one Prosa Fee FID built into a single transaction request/response message at any given time. Each of these must be separated by the FS (0x1C) byte, represented in these examples as "<FS>", as required described in TDL this legacy Specification section of the document. Subfield IDs are specified using the first 2 byte locations after the FID characters 'qb'. Subfield IDs that require only a single ASCII character to define must pad to the right with spaces (0x20), denoted below as "<SP>". The following subfields are currently defined: a. '"A<SP>' – Loyalty Fee b. '"B<SP>' – Line Fee c. '"C<SP>' – IVA Tax d. '"D<SP>' – Bank Name Following the subfield ID is the subfield information sent as ASCII printable data. For subfields A – C, the information is sent as a variable length amount in printable ASCII and must include an explicit decimal point placed anywhere in the string, but must have at least one ASCII decimal digit on each side of the decimal point (e.g. "1.50"). The value is expressed in terms of the smallest unit for the local currency used by the terminal. For subfield D, the information is sent as a printable ASCII string that can be translated directly as the bank name. An example group of 'qb' FIDs defining a loyalty fee (5.00 dollars), an IVA tax (1.00 dollar) assessed by host along with the bank name is as follows: "qbA<SP>5.00<FS>qbC<SP>1.00<FS>qbD<SP>Bank of the Redwoods<FS>" Prosa fee information sent by the terminal to the host for reversal transactions or other purposes is sent using the same 'qb' FID construct.

Table 128 – Prosa Fee FID Overview

PRELIMINARY

13.3.1 Transaction Example

This section includes an example balance transaction. In this example, the host sends back 'qb' FIDs for the loyalty fee, line fee, bank name and IVA tax. The TCP/IP length bytes at the beginning of the request/response messages are omitted from the example, and field separators are shown as "FS".

Terminal Request:

"TERMID12
FS31FS0098FS4320139800002728=110312000000401FS00000000FS00000000FSFB27F720C81BD7FBFSFS
FSub1943FS"

Host Response:

"0FSTERMID12
FS31FS0098FS000FS02380098FS021510FS112138FS021510FS00102714_00000000FSFSFSt112138FSua02151
0FSFSqbA 12.000FSqbB 1.75FSqbC 5.75FSqbD Bank of AmericaFSub4D1AFS"

PRELIMINARY

14 Triton Terminal TCP/IP Protocol Specification

14.1 Introduction

This specification outlines the protocol and configuration for the TCP/IP host communications for all Triton Systems software.

14.2 Configuration

The TCP/IP connection will be configured as specified in this section.

14.2.1 Software Configuration Parameters

Configuration parameters includes:

- Local TCP/IP address
- Local TCP/IP port number
- Gateway TCP/IP address
- Subnet Mask value
- Remote TCP/IP address (host and/or Triton Connect)
- Remote TCP/IP port number (host and/or Triton Connect)
- Response timeout

*Note * All IP addresses and Port numbers are to be statically assigned*

14.2.2 Hardware Configuration 97XX

On the docking board, jumper JP9 will be set to 5 volts. The serial DB-9 connector to RJ-45 connector for the external TCP/IP device will be plugged into Port 2 on the docking board. The 10BaseT connection on the external TCP/IP device will be connected to the local network device.

14.2.3 Hardware Configuration 96XX

On the quad serial port board, jumper JP2 will be set to 5 volts. The serial DB-9 connector to RJ-45 connector for the external TCP/IP device will be plugged into Port 2 on the docking board. The 10BaseT connection on the external TCP/IP device will be connected to the local network device.

PRELIMINARY

14.2.4 Hardware Configuration 98XX

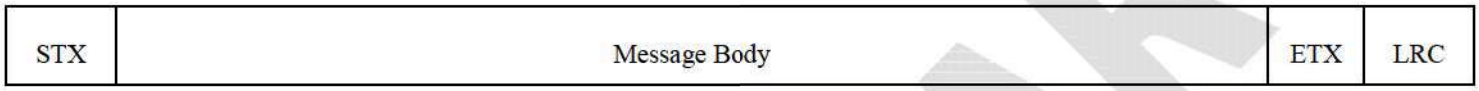
Ethernet port and cables.

PRELIMINARY

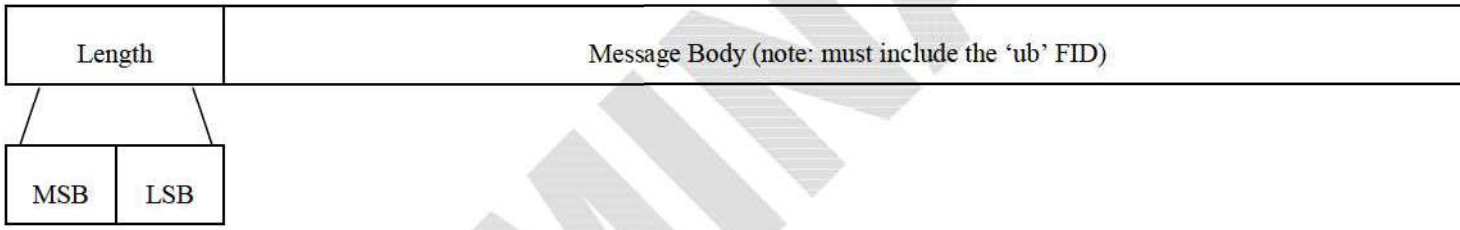
14.3 **Message Formats**

For TCP/IP, Triton software will use the specified message format defined below. The message format will follow the requests/responses outlined in the "Triton Terminal Communications Protocol and Message Format Specification" version TSCD3.5 or later. All message requests and responses listed in that specification will be supported with the following modifications.

(For reference) The specification of a message in a dial-up environment:



The specification of a message in a TCP/IP environment:



Length – a 2-byte hexadecimal value that indicates the total number of bytes in the message (not including the 2 bytes for the length). This 2-byte hexadecimal value is in the format of MSB to LSB. The maximum length of all bytes in message must not exceed the limit specified in the "Triton Terminal Communications Protocol and Message Format Specification" version TSCD3.5 or later. **The length MUST NOT exceed 16,384 byte (4000_H)**.
Note: The length field requires any serial links through which the data passes must be configured for 8-bit data.

Notes for the TCP/IP environment:
1. The STX, ETX, and LRC characters are never included.
2. The length is included with all request messages and it is expected on all response messages.
3. All request messages and responses messages must include the 'ub' FID.
4. If MAC is enabled, MAC field must be calculated in accordance with section 7.

'ubcccc'	'ub' - Two-byte FID indicating a 16-bit CRC. 'cccc' - 4 ASCII characters that represent the CRC-16 two byte checksum. Each character represents a 4-bit hexadecimal numeral (0-9, A-F). Uses the ANSI CRC-16 error checking protocol standard. See Section 6 for a description of the CRC-16 calculation.

PRELIMINARY

14.4 Protocol Definition

This section outlines the specifics for the request and response message protocol between the terminal and the host processor.

14.4.1 Response Message Timeout

The terminal response message timeout will be a configurable parameter. This value should always be greater than the maximum time it will take the host to send a valid response. The default will be 120 seconds.

14.4.2 Normal Request Message and Normal Response Message

The terminal will send a request to open a socket with the host. When this request is granted, the terminal will send the host a request message and wait up to response message timeout for the response message from the host. Once the response message is received, the terminal will release the socket with the host and process the response message.

14.4.3 Error Handling

For any of the errors described in the table below, the terminal will consider the transaction failed and no cash will be dispensed and the customer will be so advised on the receipt, if any.

Error	Action
Failure connecting to host address	Transaction fails. Terminal will go out of service. Terminal will poll the host with a configuration table download request message every 60 seconds, and if the configuration table download response is received, the terminal will go back into service.
Timeout waiting for a response after request message has been sent	Transaction fails. If the request was any financial transaction other than a balance inquiry the terminal will attempt to connect to the host and send a reversal.
Invalid length (greater than 16,384 bytes)	
Incorrect response message format	Transaction fails. If the request was any financial transaction other than a balance inquiry the terminal will attempt to connect to the host and send a reversal.
CRC error on the Configuration Table Download request message	Transaction fails. The host will not send a response message and will terminate the connection with the terminal.
CRC error on the Host Totals Download request message	
CRC error on the Transaction request message	The host will decline the request message and send an authorization code of "027" (CRC Failure) in the response message.
CRC error on the Reversal request message	
CRC error on the response message	The terminal will abort the transaction with a CRC failure. If the request was any financial transaction other than a balance inquiry the terminal will attempt to connect to the host

Highly Confidential: Property of Triton Systems – Do Not Disclose

CRC FID "ub" not present in response message	and send a reversal.

Table 129 – Error Handling

PRELIMINARY

14.5 CRC-16 (ANSI) Calculation

This section describes the implementation of the CRC-16 (ANSI) routine.

14.5.1 Including the CRC into a message

7. If no 'ub' Field present in the message, insert a FID 'ub' into one of the miscellaneous fields in the message and initialize the 4 ASCII Characters to ASCII zero (0x30) characters.
8. Calculate the length of the message excluding the 2 bytes for the length field. Insert the result into the length field in the format described in section 4.
9. Calculate the CRC on the entire message excluding the STX, ETX, and LRC (in dial-up). Starting with the MSB of the length field (in TCP/IP). An example of the code used to perform the calculation is provided in section 6.3.
10. Convert the 2-byte CRC using 4 bits at a time to produce 4 ASCII characters. Insert these characters in place of the 4 ASCII zeros (0x30) of the 'ub' field in the order of MSB to LSB.
11. Transmit the message.

14.5.2 Validating the CRC on a message

1. Wait for all characters to be received.
2. Extract the 4 ASCII characters following the FID 'ub' and replace the characters with 4 ASCII zeros (0x30). Convert the 4 ASCII characters in the order of MSB to LSB into 2 hexadecimal characters using one ASCII character at a time. Save the 2 byte hexadecimal character as the extracted CRC.
3. Recalculate the CRC on the message excluding the STX, ETX, and LRC (in dial-up). Starting with the MSB of the length field (in TCP/IP). An example of the code used to perform the calculation is provided in section 6.3.
4. Compare calculated CRC with the saved CRC from step 2.

PRELIMINARY

14.5.3 C code example of a CRC-16 (ANSI) calculation.

```
const unsigned int crc_table[256] =
{
0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
};

/* This function will take the pointer to a message (string) and the length
of the message and will return the computed CRC using the crc_table[]. */

unsigned int crc16 (unsigned char *string, int length )
{
    unsigned int crc;
    crc = 0x0000;    /* initial CRC value 0x0000 */
    while (length--)
    {
        crc = ( crc >> 8 ) ^ crc_table[(crc ^ (int) *string++) & 0xff];
      /* This performs the modulo-2 division by using the look up table. */
    }
  return(crc); /* return the calculated CRC */
}
```

PRELIMINARY

15 MAC Calculation

This section describes the implementation of the MAC field in the message.

The MAC fields implemented in this specification follow the ANSI Standard X9.9 format.

15.1 Constructing a message with a MAC

1. Build a message following the guidelines in section 4.
2. Insert a Field ID '^' into one of the miscellaneous fields in the formatted message. Fill this field with 9 ASCII zero characters (0x30).
3. Insert a FID 'ub' into one of the miscellaneous fields in the message. Fill this field 4 ASCII zero characters (0x30).
4. Calculate the length of the message excluding the 2 bytes for the length field. Insert the result into the length field in the format described in section 4.
5. Calculate the MAC following the guidelines in section 7.3.
6. Insert this 9 character string in place of the 9 ASCII zeroes in the Field ID '^' created in step 2.
7. Calculate the CRC following the guidelines in section 6.1.
8. Transmit the message.

15.2 Validating a message with a MAC

1. Wait for all characters to be received.
2. Calculate the CRC following the guidelines in section 6.2. Once the CRC calculation is completed, the 4 ASCII zeros in the 'ub' field will remain in the message.
3. Extract and save the 9 character ASCII string from the '^' Field ID. Replace the 9 ASCII characters with 9 ASCII zeros (0x30).
4. Calculate the MAC following the guidelines in section 7.3.
5. Compare the Calculated 9 Character ASCII string from step 4 with the saved 9 character ASCII string from step 3.

15.3 MAC field calculation

1. Create an 8-byte buffer in RAM called 'INPUT' and fill this block with hex zeroes.
2. Create an 8-byte buffer in RAM called 'OUTPUT' and fill this block with hex zeroes.
3. Begin at the MSB of the length in the formatted message, take 8 ASCII characters and transfer them to the INPUT buffer.
4. Exclusive-OR the OUTPUT buffer with the INPUT buffer and store in the INPUT buffer.
5. Encrypt the INPUT buffer using DES with the MAC key and store in the OUTPUT buffer.

PRELIMINARY

Highly Confidential: Property of Triton Systems – Do Not Disclose

6. Take the next 8 ASCII characters from the formatted message and transfer them to the INPUT buffer. Use all characters up to and including the last byte of the message. If less than 8 characters are available, pad the remaining position(s) with Binary zeroes.
7. Repeat steps 4 through 6 until the end of the message is reached.
8. Discard the right-most 32 bits of the OUTPUT buffer. Convert the remaining 32 bits into ASCII characters, 4 bits at a time to produce 8 ASCII characters.
9. Insert an ASCII space character (20h) after the fourth character.
10. The format of the string should be as follows: 'XXXX<sp>XXXX', where X is a hexadecimal ASCII character (0-F).

MAC Field Use

When enabled, the terminal will send the MAC field with all message requests except Configuration Table Download request, and will expect them in all message responses except Configuration Table Download response. If the terminal receives an invalid MAC field in a response message, it will immediately perform a Configuration Table Download on a separate call to update the MAC and/or COMMS key. If the failed MAC was contained in a Transaction Response Message and if the request was any financial transaction other than a balance inquiry, a reversal will follow the Configuration Table Download. Also if a withdrawal was performed, no cash will be dispensed and the customer will be so advised on the receipt, if any.

16 Radio PAD Protocol

This section describes the Radio PAD (**P**acket **A**ssembler/**D**isassembler) protocol elements implemented in Triton Systems terminals. Specifically the RS232 side of the RadioPAD link is covered in this section, including any message level protocol conventions used by the Triton application layer. The BAUD rate between the terminal and device is always 4800 bps.

16.1 Control Mode

The RadioPAD device is needs to be in control mode for attempts to connect to the network. Control mode is RadioPAD device specific, but allows the terminal to send setup information and connect to the provisioned NUA (**N**etwork **U**ser **A**ddress). RS232 communications data exchanged with the device in control mode are always 7/E/1 (7 data bits, Even Parity, and 1 stop bit).

16.2 Data Mode

Once connected to the network, the terminal transitions to data mode. In data mode the device transmits/receives data to/from the network as 8/N/1 (8 data bits, No parity, 1 stop bit) data typically. For some locations, data is conditioned by the terminal to add even parity, emulating 7/E/1 settings, see Figure 6. The maximum size transmit/receive buffer is 65535 (0xFFFF) bytes for platforms other than Z180. These older model terminals still limit the buffer size as specified in section 10.3.

PRELIMINARY

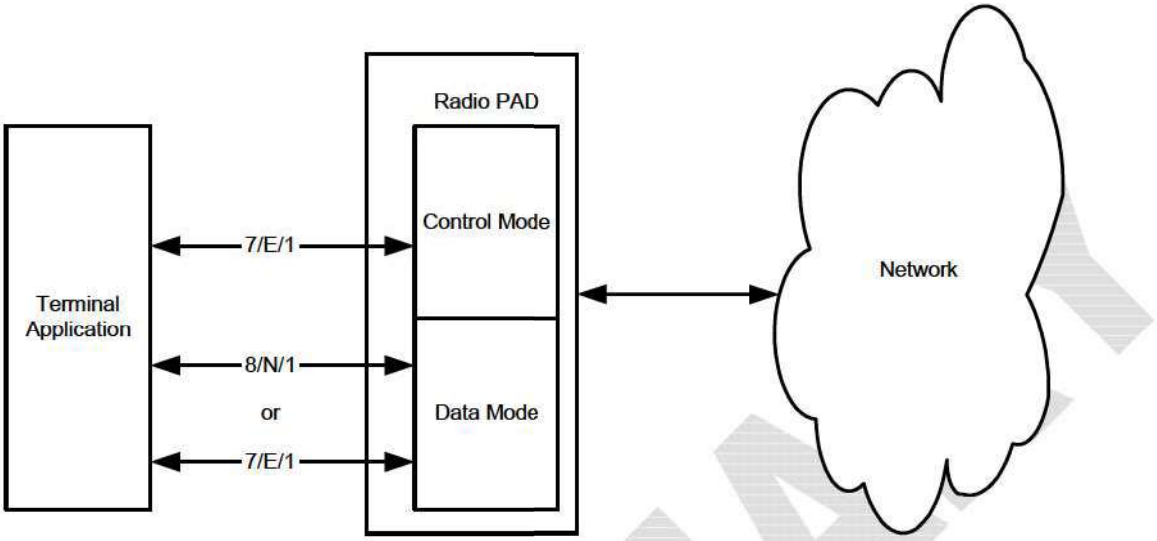


Figure 6 Terminal RP Network Access

16.3 Application Layer Protocol

Application data is framed in a similar fashion as Visa II standards (see section 10.3) minus the ENQ byte exchange. The basic message structure is diagrammed in Figure 7.

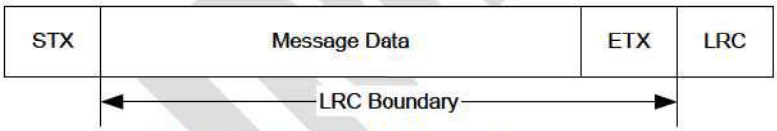


Figure 7 Radio PAD Message Structure

The sender starts with an STX byte indicating the start of the message block. The end of the message data is the ETX followed by the LRC byte calculated by the sender. The LRC is a typical longitudinal redundancy check calculated as an exclusive OR over the message contents from the first message data byte through and including the ETX. Assuming a good LRC is calculated, the terminal sends an ACK back to the host, otherwise a NAK is sent and the message is retried. After an ACK is sent, the terminal waits for an EOT. Failure to receive an EOT is an error condition. Upon any error condition, the terminal hangs up and performs the appropriate communications exception handling. If the EOT is received by the terminal, an EOT is sent to the host from the terminal and the call is terminated. Figure 8 diagrams the Write/Read process in communicating with the host over Radio PAD. Upon successful transmission and response processing the message data is then passed up to the application for transaction processing.

PRELIMINARY

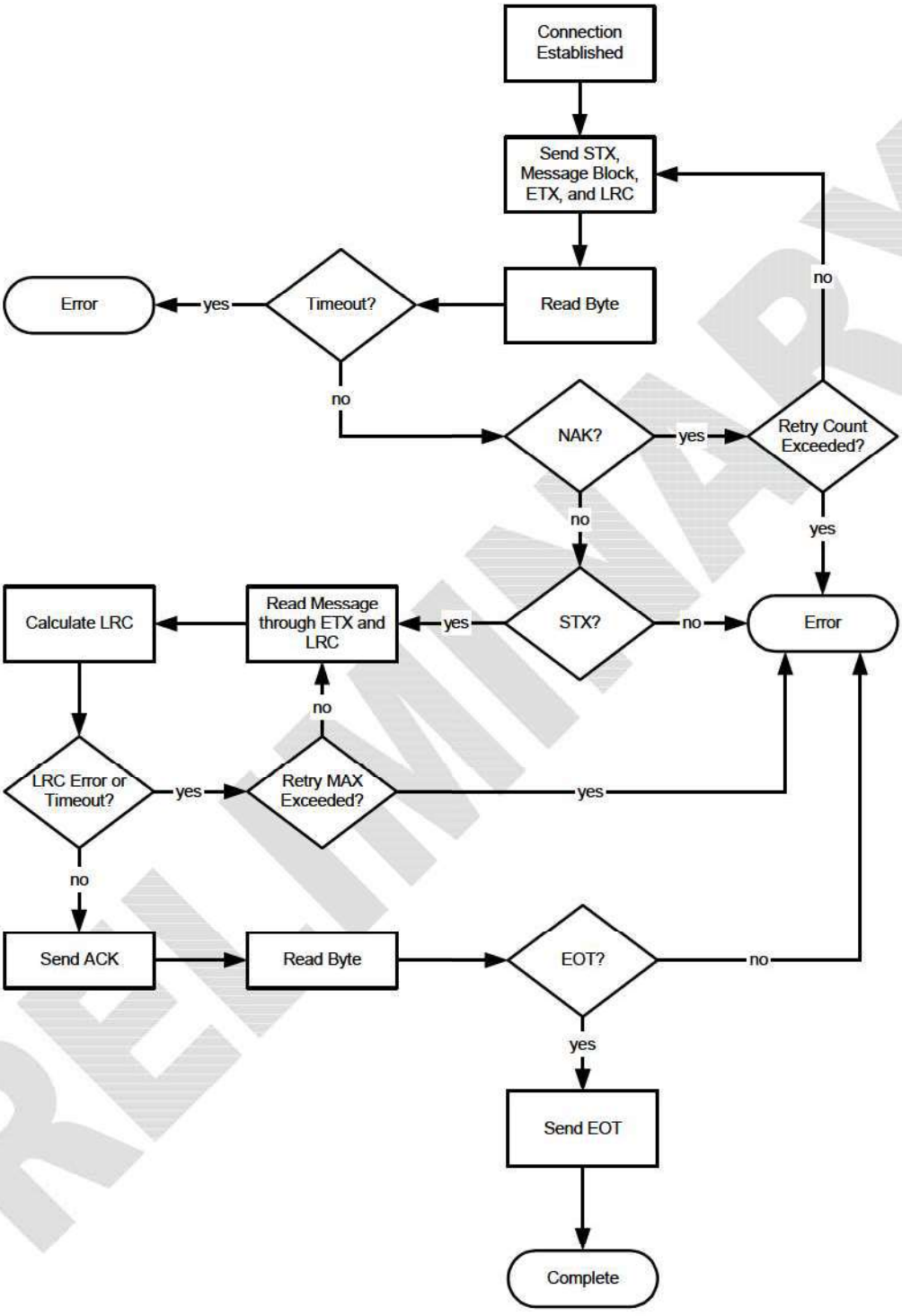


Figure 8 Application Protocol Over Radio PAD

PRELIMINARY