# Tokens Manual

BASE24®

**© 2013 by ACI Worldwide, Inc. All rights reserved.**

All information contained in this documentation, as well as the software described in it, is confidential and proprietary to ACI Worldwide, Inc., or one of its subsidiaries, is subject to a license agreement, and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by electronic, mechanical, recording, or any other means, without the prior written permission of ACI Worldwide, Inc., or one of its subsidiaries.

ACI, ACI Worldwide, and the ACI product names used in this documentation are trademarks or registered trademarks of ACI Worldwide, Inc., or one of its subsidiaries.

Other companies' trademarks, service marks, or registered trademarks and service marks are trademarks, service marks, or registered trademarks and service marks of their respective companies.

# Contents


**What's New** . . . . . xv

**Preface** . . . . . xxix

**1: Introduction** . . . . . 1-1

BASE24 Internal Messages . . . . . 1-2

Internal Message Conversions . . . . . 1-2

Tokens in the Internal Message. . . . . 1-4

STM, PSTM, and TSTMH Core Fields . . . . . 1-4

Function-Specific Fields . . . . . 1-5

Tokens in Other Structures . . . . . 1-6

Tokens and BASE24-telebanking . . . . . 1-7

Internal Transaction Data . . . . . 1-7

Token Processing . . . . . 1-7

Token Basics and Examples . . . . . 1-9

Header Token . . . . . 1-9

Token Headers . . . . . 1-10

ASCII and Binary Token Formats . . . . . 1-18

Example of Message Processing with Tokens . . . . . 1-19

**2: System Token Processing** . . . . . 2-1

Internal Message Processing Overview . . . . . 2-2

Storing Message Information in BASE24 Files . . . . . 2-2

Token Length Limits . . . . . 2-3

Processes that Handle Token Data . . . . . 2-4

Device Handler Processes. . . . . 2-4

Authorization Processes . . . . . 2-5

Integrated Authorization Server Processes . . . . . 2-5

ISO Host Interface Processes . . . . . 2-5

Interchange Interface Processes . . . . . 2-6

Migration Process . . . . . 2-8
Super Extract Process . . . . . 2-8
Settlement Initiator Processes . . . . . 2-8
Enscribe Refresh Process . . . . . 2-9
Transaction Log File Perusal Subsystems . . . . . 2-9
Report Programs . . . . . 2-9
Token Buffers . . . . . 2-11
Token Utilities . . . . . 2-12
Determining Which Tokens Are in the Token Buffer . . . . . 2-13
TKN^GET^INFO Utility . . . . . 2-13
TKN^GET^IDS Utility . . . . . 2-18
Adding Tokens to the Token Buffer . . . . . 2-24
TKN^ADD^INFO Parameters . . . . . 2-24
Status Codes Returned by the TKN^ADD^INFO Utility . . . . . 2-28
TKN^ADD^INFO Processing . . . . . 2-28
Updating Existing Tokens . . . . . 2-31
TKN^UPDT^INFO Parameters . . . . . 2-31
Status Codes Returned by the TKN^UPDT^INFO Utility . . . . . 2-34
TKN^UPDT^INFO Processing . . . . . 2-35
Moving Tokens to a Different Token Buffer . . . . . 2-37
TKN^SORT^INFO Parameters . . . . . 2-37
Status Codes Returned by the TKN^SORT^INFO Utility . . . . . 2-44
TKN^SORT^INFO Processing . . . . . 2-45
Logging Tokens to Token Buffer . . . . . 2-49
TKN^LOG^INFO Parameters . . . . . 2-49
Status Codes Returned by the TKN^LOG^INFO Utility . . . . . 2-51
TKN^LOG^INFO Processing . . . . . 2-52
Deleting Tokens . . . . . 2-55
TKN^DEL^INFO Parameters . . . . . 2-55
Status Codes Returned by the TKN^DEL^INFO Utility . . . . . 2-58
TKN^DEL^INFO Processing . . . . . 2-58
Converting Token Data to Another Format . . . . . 2-61
TKN^MAIN^CONVERT Parameters . . . . . 2-62
Status Codes Returned by the TKN^MAIN^CONVERT Utility . . . . . 2-65
TKN^MAIN^CONVERT Processing . . . . . 2-66

**3: User Tokens . . . . . 3-1**
Creating User Tokens . . . . . 3-2

**4: Configuring Internal Message Token Processing . . . . . 4-1**
Token File Screens . . . . . 4-2
Specifying Tokens to be Logged. . . . . 4-3
Key Field Settings. . . . . 4-4
Operator Procedures . . . . . 4-15
Updating TKN Logging Records . . . . . 4-18
Specifying Tokens to be Extracted . . . . . 4-19
Key Field Settings. . . . . 4-19
Operator Procedures . . . . . 4-31
Updating TKN Extract Records . . . . . 4-33
Specifying Tokens to be Sent in the External Message . . . . . 4-35
Key Field Settings. . . . . 4-35
Operator Procedures . . . . . 4-42

**5: BASE24 Base Tokens . . . . . 5-1**
Header Token—Binary Format. . . . . 5-4
Header Token—ASCII Format . . . . . 5-5
Token Header—Binary Format. . . . . 5-6
Token Header—ASCII Format . . . . . 5-7
Token 08 Customer Name Token—Binary Format . . . . . 5-8
Token 08 Customer Name Token—ASCII Format . . . . . 5-9
Token 12 MICR Data Token . . . . . 5-10
Token 13 Credit Line Token—Binary Format . . . . . 5-11
Token 13 Credit Line Token—ASCII Format . . . . . 5-12
Token 18 Account Qualifier Token . . . . . 5-13
Token 23 Track 1 Token. . . . . 5-14
Token 25 Surcharge Data Token—Binary Format. . . . . 5-15
Token 25 Surcharge Data Token—ASCII Format . . . . . 5-18
Token 27 Cardholder Postal Code Token. . . . . 5-19
Token 28 ACI Proactive Risk Manager Token. . . . . 5-20

Token 30 Issuer Fee Rebate Token—Binary Format . . . . . 5-23
Token 30 Issuer Fee Rebate Token—ASCII Format . . . . . 5-25
Token 32 PRM Real Time Token. . . . . 5-26
Token B0 Switch Token (Acquirer) and
Token B1 Switch Token (Issuer)—Binary Format . . . . . 5-28
Interface FIIDs and Version IDs. . . . . 5-28
Alaska Option ISO . . . . . 5-30
American Express CAPN ISO (AXCI) . . . . . 5-32
American Express Global Network (GNS) . . . . . 5-34
BankNet . . . . . 5-35
BIC ANSI . . . . . 5-42
BIC ISO . . . . . 5-42
Cash Station ISO. . . . . 5-43
Deluxe ISO . . . . . 5-45
DIAS . . . . . 5-46
Discover ISO. . . . . 5-47
EPS-Net . . . . . 5-50
FDR ISO . . . . . 5-51
JCB ISO . . . . . 5-51
LINK (LIS5) . . . . . 5-52
MAC MASM (MACI) . . . . . 5-56
MDS Cirrus ISO . . . . . 5-58
Money Station (MONY). . . . . 5-65
MPS . . . . . 5-66
NBGC . . . . . 5-67
NPC ISO . . . . . 5-69
NYCE ISO . . . . . 5-70
Networks ISO . . . . . 5-73
PLUS ISO . . . . . 5-75
Pulse ISO . . . . . 5-78
SPAN2 . . . . . 5-80
Star ISO. . . . . 5-81
SVS . . . . . 5-86
Shazam ISO . . . . . 5-87
ValueLink . . . . . 5-89
Visa Debit Processing Service (DPS). . . . . 5-91

VisaNet . . . . . 5-95
Token B0 Switch Token (Acquirer) and
Token B1 Switch Token (Issuer)—ASCII Format . . . . . 5-105
Interface FIIDs and Version IDs. . . . . 5-105
Alaska Option ISO . . . . . 5-106
American Express CAPN ISO (AXCI) . . . . . 5-106
American Express Global Network (GNS) . . . . . 5-107
BankNet . . . . . 5-107
BIC ANSI . . . . . 5-109
BIC ISO . . . . . 5-109
Cash Station ISO. . . . . 5-110
Deluxe ISO . . . . . 5-110
DIAS . . . . . 5-111
Discover ISO. . . . . 5-111
EPS-Net . . . . . 5-112
FDR ISO . . . . . 5-113
JCB ISO . . . . . 5-113
LINK (LIS5) . . . . . 5-113
MAC MASM (MACI) . . . . . 5-114
MDS Cirrus ISO . . . . . 5-115
Money Station (MONY). . . . . 5-117
MPS . . . . . 5-117
NBGC . . . . . 5-117
NPC ISO . . . . . 5-118
NYCE ISO . . . . . 5-119
Networks ISO . . . . . 5-120
PLUS ISO . . . . . 5-120
Pulse ISO . . . . . 5-121
SPAN2 . . . . . 5-122
Star ISO. . . . . 5-122
SVS . . . . . 5-124
Shazam ISO . . . . . 5-124
ValueLink . . . . . 5-125
Visa Debit Processing Service (DPS). . . . . 5-126
VisaNet . . . . . 5-127
Token B2 EMV Request Data Token—Binary Format . . . . . 5-130

Token B2 EMV Request Data Token—ASCII Format . . . . . 5-172
Token B3 EMV Discretionary Data Token—Binary Format. . . . . 5-174
Token B3 EMV Discretionary Data Token—ASCII Format. . . . . 5-183
Token B4 EMV Status Token—Binary Format. . . . . 5-184
Token B4 EMV Status Token—ASCII Format . . . . . 5-193
Token B5 EMV Response Data Token—Binary Format. . . . . 5-194
Token B5 EMV Response Data Token—ASCII Format . . . . . 5-202
Token B6 EMV Script Data Token—Binary Format. . . . . 5-203
Token B6 EMV Script Data Token—ASCII Format. . . . . 5-204
Token B7 TLF Token—Binary Format . . . . . 5-205
Token B7 TLF Token—ASCII Format . . . . . 5-208
Token B8 Transaction Profile Token . . . . . 5-209
Token B9 Transaction Description Token . . . . . 5-210
Token BA Acquirer Routing Token . . . . . 5-211
Token BB Pre-Pay Generic Receipt Token—Binary Format. . . . . 5-212
Token BB Pre-Pay Generic Receipt Token—ASCII Format. . . . . 5-213
Token BC TSS Index Token . . . . . 5-214
Token BD Multiple Currency Token—Binary Format . . . . . 5-215
Token BD Multiple Currency Token—ASCII Format. . . . . 5-218
Token BE Original Currency Release 6.0 Token—Binary Format . . . . . 5-219
Token BE Original Currency Release 6.0 Token—ASCII Format . . . . . 5-221
Token BF Pre-Pay Receipt Token—Binary Format. . . . . 5-222
Token BF Pre-Pay Receipt Token—ASCII Format. . . . . 5-223
Token BG Track 3 Token. . . . . 5-224
Token BH Reversal Date and Time Token. . . . . 5-225
Token BI Pre-Pay Top-Up Token. . . . . 5-227
Token BJ EMV Issuer Script Results Token . . . . . 5-231
Token BK Multiple Logical Network Token—Binary Format . . . . . 5-233
Token BK Multiple Logical Network Token—ASCII Format . . . . . 5-234
Token BL Virtual Primary Account Number Token . . . . . 5-235
Token BM Transaction Subtype Token . . . . . 5-236

Token BN Data Encryption Key Token . . . . . 5-239
Token BO Encrypted Balance Token—Binary Format . . . . . 5-240
Token BO Encrypted Balance Token—ASCII Format . . . . . 5-241
Token BP Person-to-Person Transaction Token. . . . . 5-242
Token BQ Completion Required Token. . . . . 5-245
Token BR Split Transaction Routing Token . . . . . 5-246
Token BS Pre-Pay Switch Token—Binary Format . . . . . 5-250
Token BS Pre-Pay Switch Token—ASCII Format . . . . . 5-252
Token BT Pre-Pay Response Token—Binary Format . . . . . 5-253
Token BT Pre-Pay Response Token—ASCII Format . . . . . 5-255
Token BU Pre-Pay Selection Token—Binary Format . . . . . 5-256
Token BU Pre-Pay Selection Token—ASCII Format . . . . . 5-259
Token BV Pre-Pay Voucher Receipt Token—Binary Format . . . . . 5-260
Token BV Pre-Pay Voucher Receipt Token—ASCII Format . . . . . 5-262
Token BW Pre-Pay Online Receipt Token—Binary Format . . . . . 5-263
Token BW Pre-Pay Online Receipt Token—ASCII Format . . . . . 5-264
Token BX Pre-Pay Original Data Token . . . . . 5-265
Token BY Switch Common Data Token . . . . . 5-266
Token M1 Migration ATM Data1 Token . . . . . 5-269
Token M2 Migration POS Data1 Token . . . . . 5-270
Token M4 Migration EPS HISO Token . . . . . 5-271
Token M5 Migration Customer Data Token . . . . . 5-272
Token N8 Inventory Voucher Token—Binary Format . . . . . 5-274
Token N8 Inventory Voucher Token—ASCII Format . . . . . 5-276
Token S0 Intra Country Data Token . . . . . 5-277
Token S1 Gateway Info Token. . . . . 5-278
Token S2 Dynamic Currency Conversion Status Token . . . . . 5-281
Token S3 Dynamic Currency Conversion Processing Token—Binary Format . . . . . 5-283
Token S3 Dynamic Currency Conversion Processing Token—ASCII Format . . . . . 5-287
Token S4 EMV Supplementary Data Token—Binary Format . . . . . 5-288
Token S4 EMV Supplementary Data Token—ASCII Format . . . . . 5-289

Token S6 Track 2 Token . . . . . 5-290
Token S7 Person-to-Person Transaction 2 Token . . . . . 5-291
Token S8 PAN Mapping Token . . . . . 5-293
Token S9 Additional Authorization Data token. . . . . 5-294
Token SA Generic Data Token. . . . . 5-295

**6: BASE24-atm Tokens . . . . . 6-1**
Token 02 Statement Print Token . . . . . 6-3
Token 03 BASE24-atm Release 5.0 Token . . . . . 6-4
Token 06 PIN Change Token . . . . . 6-7
Token 07 Self-Service Banking Base Token—Binary Format . . . . . 6-9
Token 07 Self-Service Banking Base Token—ASCII Format. . . . . 6-10
Token 14 Self-Service Banking Check Token . . . . . 6-11
Token 15 Self-Service Banking Check Terminal Settlement Token—Binary Format 6-13
Token 15 Self-Service Banking Check Terminal Settlement Token—ASCII Format. 6-14
Token 21 PS2000 ATM Token. . . . . 6-15
Token 22 Additional Hopper Token—Binary Format . . . . . 6-16
Token 22 Additional Hopper Token—ASCII Format . . . . . 6-18
Token 24 ATM Flag 1 Token . . . . . 6-19
Token A5 Non-Currency Dispense Token . . . . . 6-21
Token A6 ATM Interchange Compliance Token . . . . . 6-22
Token A7 Multiple Account Token . . . . . 6-23
Token A8 Bag Deposit Token . . . . . 6-25
Token A9 Money Exchange Token—Binary Format . . . . . 6-26
Token A9 Money Exchange Token—ASCII Format. . . . . 6-28
Token AA Merchant Banking Center Settlement Token—Binary Format . . . . . 6-29
Token AA Merchant Banking Center Settlement—ASCII Format . . . . . 6-32
Token AB ATM Balances Token—Binary Format . . . . . 6-33
Token AB ATM Balances Token—ASCII Format. . . . . 6-36
Token AD Cash Acceptor Terminal Settlement Token . . . . . 6-37
Token AE Bill Payment Payee List Token—Binary Format . . . . . 6-38
Token AE Bill Payment Payee List Token—ASCII Format . . . . . 6-40

Token AF Bill Payment Confirmation Token . . . . . 6-41
Token AG ATM Data 1 Token . . . . . 6-42
Token AH Multiple Account Inquiry Token—Binary Format . . . . . 6-43
Token AH Multiple Account Inquiry Token—ASCII Format . . . . . 6-44
Token AI Interim Statement/Passbook Data Token . . . . . 6-45
Token AJ Interim Statement Processing Token—Binary Format . . . . . 6-48
Token AJ Interim Statement Processing Token—ASCII Format . . . . . 6-49
Token AK Passbook Processing Token—Binary Format . . . . . 6-50
Token AK Passbook Processing Token—ASCII Format . . . . . 6-52
Token AL Hold Token—Binary Format . . . . . 6-53
Token AL Hold Token—ASCII Format . . . . . 6-54
Token AM ATM Preferred Transaction Token—Binary Format . . . . . 6-55
Token AM ATM Preferred Transaction Token—ASCII Format . . . . . 6-57
Token AO Diebold BNA Counts Token—Binary Format . . . . . 6-58
Token AO Diebold BNA Counts Token—ASCII Format . . . . . 6-60
Token AR Custom Response Code Token . . . . . 6-61
Token AS Shared BNA Counts Token—Binary Format . . . . . 6-62
Token AS Shared BNA Counts Token—ASCII Format . . . . . 6-64
Token AT BNA Multiple Currency Token—Binary Format . . . . . 6-65
Token AT BNA Multiple Currency Token—ASCII Format . . . . . 6-67
Token AU Check Bundle Token—Binary Format . . . . . 6-68
Token AU Check Bundle Token—ASCII Format . . . . . 6-69
Token AV Bulk Check MICR Token—Binary Format . . . . . 6-70
Token AV Bulk Check MICR Token—ASCII Format . . . . . 6-71
Token AW Bulk Check Amount Token—Binary Format . . . . . 6-72
Token AW Bulk Check Amount Token—ASCII Format . . . . . 6-73
Token AX Bulk Check SSB Token—Binary Format . . . . . 6-74
Token AX Bulk Check SSB Token—ASCII Format . . . . . 6-76
Token AY Bulk Check Disposition Token—Binary Format . . . . . 6-77
Token AY Bulk Check Disposition Token—ASCII Format . . . . . 6-78

**7: BASE24-pos Tokens . . . . . 7-1**

Token 01 Address Verification Token . . . . . 7-4
Token 04 BASE24-pos Release 5.0 Token . . . . . 7-5
Token 05 Check Guarantee/Verification Token . . . . . 7-9
Token 10 American Express Token—Binary Format . . . . . 7-11
Token 10 American Express Token—ASCII Format . . . . . 7-19
Token 11 Automated Clearing House (ACH) Debit Token . . . . . 7-21
Token 16 Alternate Merchant ID Token . . . . . 7-22
Token 17 Visa Payment Service 2000 Token . . . . . 7-23
Token 19 Visa Payment Service 2000 Offline Token . . . . . 7-26
Token 20 Interchange Compliance Token . . . . . 7-29
Token 29 Check Guarantee/Verification 2 Token . . . . . 7-31
Token 31 Check Callback Token . . . . . 7-33
Token C0 BASE24-pos Release 5.1 Token . . . . . 7-34
Token C1 Station ID Token . . . . . 7-39
Token C2 Purchase Card and Fleet Card Token . . . . . 7-40
Token C3 Certificate Token . . . . . 7-59
Token C4 Point of Service Data Token . . . . . 7-60
Token C5 Increased Optional Data Token . . . . . 7-64
Token C6 Trans Stain XID Token . . . . . 7-65
Token C7 Cardholder Serial Number Token . . . . . 7-66
Token C8 Merchant Serial Number Token . . . . . 7-67
Token C9 MHI Additional Data Token—Binary Format . . . . . 7-68
Token C9 MHI Additional Data Token—ASCII Format . . . . . 7-69
Token CA DUKPT Data Token . . . . . 7-70
Token CB POS Balances Token—Binary Format . . . . . 7-71
Token CB POS Balances Token—ASCII Format . . . . . 7-73
Token CE Authentication Data Token . . . . . 7-74
Token CF International Address Verification Service (IAVS) Data Token . . . . . 7-76
Token CH POS Data1 Token—Binary Format . . . . . 7-77
Token CH POS Data1 Token—ASCII Format . . . . . 7-83

Token CI POS Merchant Token . . . . . 7-84
Token CJ Pre-Pay Merchant Token—Binary Format . . . . . 7-88
Token CJ Pre-Pay Merchant Token—ASCII Format. . . . . 7-90
Token CK Industry Data Token . . . . . 7-91
Token CP Healthcare Token. . . . . 7-97
Token CQ Reward Program Token . . . . . 7-106
Token CR POS Split Transaction Routing Token . . . . . 7-107
Token CS Enhanced Reversal Routing Token . . . . . 7-108
Token CT Transaction Specific Data Token Using Redefines. . . . . 7-109
Token CT Transaction Specific Data Token Using Extended Datasets . . . . . 7-113
Token CU American Express Additional Data Token . . . . . 7-123
Token CV Healthcare/Transit Token—Binary Format . . . . . 7-124
Token CV Healthcare/Transit Token—ASCII Format. . . . . 7-126
Token CW Healthcare Service Token—Binary Format. . . . . 7-127
Token CW Healthcare Service Token—ASCII Format . . . . . 7-129
Token CX American Express Private Use Data Token . . . . . 7-130
Token CY Auto-Substantiation Data Token. . . . . 7-131
Token CZ POS Data 2 Token—Binary Format . . . . . 7-132
Token CZ POS Data 2 Token—ASCII Format . . . . . 7-134
Token F1 E-commerce Additional Data Token . . . . . 7-135
Token F2 Installment Payment Data Token . . . . . 7-136
Token F3 Transit Transaction Token. . . . . 7-137
Token F4 Digital Wallet Token . . . . . 7-139
Token U0 EBT Voucher Token . . . . . 7-142
Token U1 EBT Available Balance Token—Binary Format. . . . . 7-143
Token U1 EBT Available Balance Token—ASCII Format . . . . . 7-145
Token U2 Stored Value Token—Binary Format . . . . . 7-146
Token U2 Stored Value Token—ASCII Format . . . . . 7-149

**8: BASE24-teller Tokens . . . . . 8-1**
Token T0 Financial Token—Binary Format . . . . . 8-2
Token T0 Financial Token—ASCII Format . . . . . 8-5

Token T1 CAF Inquiry Token—Binary Format . . . . . 8-6
Token T1 CAF Inquiry Token—ASCII Format . . . . . 8-11
Token T2 CAF Update Token . . . . . 8-12
Token T3 NBF Token—Binary Format . . . . . 8-14
Token T3 NBF Token—ASCII Format . . . . . 8-18
Token T4 PBF Inquiry Token—Binary Format . . . . . 8-19
Token T4 PBF Inquiry Token—ASCII Format . . . . . 8-24
Token T5 PBF Update Token . . . . . 8-25
Token T6 SPF Inquiry Token—Binary Format . . . . . 8-26
Token T6 SPF Inquiry Token—ASCII Format . . . . . 8-30
Token T7 SPF Update Token—Binary Format . . . . . 8-31
Token T7 SPF Update Token—ASCII Format . . . . . 8-33
Token T8 WHFF Inquiry Token—Binary Format . . . . . 8-34
Token T8 WHFF Inquiry Token—ASCII Format . . . . . 8-39
Token T9 WHFF Update Token—Binary Format . . . . . 8-40
Token T9 WHFF Update Token—ASCII Format . . . . . 8-42
Token TA Administrative Token . . . . . 8-43
Token TB Account Token—Binary Format . . . . . 8-44
Token TB Account Token—ASCII Format . . . . . 8-46
Token TC Override Token—Binary Format . . . . . 8-47
Token TC Override Token—ASCII Format . . . . . 8-50
Token TD PIN Token—Binary Format . . . . . 8-51
Token TD PIN Token—ASCII Format . . . . . 8-53
Token TE Native Message Token . . . . . 8-54

**A: Token Impacts on Log File Disk Space Requirements . . . . . A-1**
Information Needed to Determine Log File Impacts . . . . . A-2
How Logging Tokens Can Affect Disk Space Requirements . . . . . A-3

**Index . . . . . Index-1**

# What's New

The following tables highlight the major changes that have been made to the ***BASE24 Tokens Manual*** since its publication for BASE24 release 6.0 version 10. The first column of each table lists the sections and appendixes in which major changes have been made. The second column of each table describes the major changes for the section or appendix.

## September 2013

| Section/ Appendix | Major Changes |
|---|---|
| 1 | Adds new tokens, M5 and SA, as Base tokens. |
| 5 | Adds one new field for token B0/B1 (Banknet ISO Interface) to hold the gratuity amount. |
| 5 | Adds one new field for token B0/B1 (Discover ISO Interface) to hold point of service data. |
| 5 | Adds six new fields for token B0/B1 (LINK LIS5 Interface) to handle Mobile Payment Transactions (MPT). |
| 5 | Adds one new field for token B0/B1 (MDS Cirrus ISO Interface) to hold the gratuity amount. |
| 5 | Adds four new fields for token B0/B1 (NYCE ISO Interface) to hold the PIN description and indicator. |
| 5 | Adds two new fields for token B0/B1 (Shazam Interface) to hold card acceptor location data. |
| 5 | Adds one new field for token B0/B1 (VisaNet ISO Interface) to hold the spend qualified indicator and corrects one field. |
| 5 | Adds a token, M5 (Migration Customer Data), to hold migration data. |
| 5 | Adds a token, SA (Generic Data), to hold generic interface data. |

| Section/ Appendix | Major Changes |
|---|---|
| 7 | Updates two field descriptions in the C2 token (Purchase): LOCAL-TAX-INCL field should define "0" as Tax not included, and MISC-FUEL-TAX should not mention implied decimal places. |
| 7 | Adds two new values in the ERR-FLG field in token 04 (POS 5.0) for unmatched reversals. |
| 7 | Adds a new value to the PARTIAL-AUTH-OPT field, adds three new values to the RVSL-RSN-IND field, and adds two new fields, AUTH-MSG-IND and TERM-TYP, for token CH (POS Data1). |
| 7 | Updates the BUS-APPL-ID field in the Transaction Specific Data Token (CT) to add additional valid values for Shazam. MDS now uses this token as well. |
| 7 | Adds three new data sets in the Transaction Specific Data Token using Extended Data Sets (CT) to add additional fields for VisaNet ISO Interface. |

# May 2013

| Section/ Appendix | Major Changes |
|---|---|
| 1 | Adds token S9 (Additional Authorization Data) as a new BASE24 Base token. |
| 5 | Adds two new fields to hold the product type and Bilateral Discretionary data for token B0/B1 (LINK Interface).<br><br>Updates comments to the DATASET-ID field in token S4 (EMV Supplementary Data) to include support for value "00" and clarify that a value of "01" is not specific to Visa transactions.<br><br>Adds token S9 (Additional Authorization Data) to support the additional authorization data that may conditionally be included in Visa field 48, usage 2. |
| 6 | Updates token 21 (PS2000 ATM) with new values "0" and "␣" (where ␣ is a blank space) to the SRV-IND field and comments for the TRAN-ID and VALID-CDE fields to reflect current Visa usages. |

| Section/ Appendix | Major Changes |
|---|---|
| | Updates token 24 (ATM Flag 1) with comments for the AUTH-ONLY and SVC-IND fields. |
| 7 | Updates token 17 (Visa Payment Service 2000) with new values and comments for the SRV-IND field, in addition to comments for the TRAN-ID and VALID-CDE fields.<br><br>Updates token C2 (Purchase Card and Fleet Card) with new fields for the VISA-FLEET-DATA redefine of TKN-DATA. Decrements USER-FLD6 by 170 bytes to accommodate the new fields.<br><br>Updates token CT (Transaction Specific Data) with new fields for the MC-MBR-DEF-DATA definition and increments the token Dataset ID to ~7.<br><br>Updates token F3 (Transit Transaction) with new value "07" in the TXN-TYP-IND field.<br><br>Updates token F4 (Digital Wallet) with the new redefine NYCE-MOBILE-ACCESS, in addition to a new value "03" for the WALLET-IND-FLG field. |

# September 2012

| Section/ Appendix | Major Changes |
|---|---|
| 1 | Adds token F4 (Digital Wallet) as a new BASE24-pos token. |
| 5 | Expands the size of token B0/B1 (Pulse ISO Interface) to include the new NATL-PT-SVC-COND-CDE field. |
| 7 | Adds the following new fields to the VISA-FLEET-DATA redefine in the Purchase Card and Fleet Card (C2) token:<br><br>FUEL-BRAND<br>FUEL-TXN-VALID-RSLT<br>FUEL-ACCPT-MDE<br>DRV-ID<br>JOB-NUM<br>FLEET-NUM<br>VEHICLE-RGSTR-NUM |

| Section/ Appendix | Major Changes |
|---|---|
| | Adds the FAILED-CVM-ALWD and DUP-CHK-REQ fields to the POS Data1 (CH) token.<br><br>Adds token F4 (Digital Wallet) as a new BASE24-pos token. |

# March 2012

| Section/ Appendix | Major Changes |
|---|---|
| 1 | Adds token F2 (Installment Payment Data) and F3 (Transit Transaction) as new BASE24-pos tokens. |
| 5 | Expands the size of token B0/B1 (Star ISO Interface) to include new fields:<br><br>MC-RATE-IND<br>STAR-ISSUER-IGI<br>NATL-PT-SVC-COND-CDE<br>INTERLINK-ATTR<br>ICHG-GRP-ID<br>AGGR-IND<br>AUTH-PGM<br>TXN-SUBTYP<br>PROD-ID<br><br>Expands the size of token B0 (LINK Interface) to include new fields SOCL-DEPRIVATN-AREA, OUTSIDE-HOME-TERRITORY-IND, POST-CDE, and two future use fields.<br><br>Updates token B2 (EMV Request Data) with additional values for field CRD-VRFY-RSLTS.<br><br>Updates token BM (Transaction Subtype) with an additional value for field TXN-SUBTYPE.<br><br>Updates token BP (Person-to-Person Transaction) with additional values for field BUS-USE-FLAG.<br><br>Updates token S1 (Gateway Info) to include new field ECOMM-IND. |
| 7 | Adds token F2 (Installment Payment Data) as a new BASE24-pos token. |

| Section/ Appendix | Major Changes |
|---|---|
| | Adds token F3 (Transit Transaction) as a new BASE24-pos token.<br><br>Updates token 17 (Visa Payment Service 2000) with additional values for fields SRV_IND and MKT-SPFC-DATA-ID.<br><br>Updates token 20 (Interchange Compliance) with new values for field LIFE-CYCLE-IND.<br><br>Updates token CH (POS Data1) with corrected values for fields NUM-MM-GRATUITY and RVSL-RSN-IND, and new fields INSTL-PLAN-TYP and INSTL-GRATUITY-PRD.<br><br>Updates token CT (Transaction Specific Data Token Using Extended Datasets) to increment the second byte of the DATASET-ID field from 5 to 6.<br><br>Updates token C4 (Point of Service Data) description for field TERM-INPUT-CAP-IND. |

# November 2011

| Section/ Appendix | Major Changes |
|---|---|
| 2 | Adds description for token usage during BASE24-eps migration process. |
| 5 | Expands the size of token B0/B1 (NYCE ISO Interface) to include new field ISS-CAT.<br><br>Replaces USER-FLD with new field TERM-TYP on token B0/B1 (VisaNet ISO Interface).<br><br>Adds new Base migration token M4. This token is used for message exchanges with BASE24-eps and is not intended to be extracted.<br><br>Adds VISA-GTWY-DSCV-INFO overlay to token S1 (Gateway Info). |
| 7 | Replaces USER-FLD1 with new field RVSL-RSN-IND on token CH (POS Data1).<br><br>Includes new field FUND-SRC in the MONEY-XFER-DATA defintion, and increments the value of the DATASET-ID field to '~5' on token CT (Transaction Specific Data Using Extended Datasets).<br><br>Includes new dataset definition MC-MBR-DEF-DATA on token CT (Transaction Specific Data Using Extended Datasets).<br><br>Increases existing field ASGN-ID from 6 bytes to 10 bytes on token CY (Auto-Substantiation Data), and notes the field can now be used to carry a Visa Merchant Verification Value (MVV). |

# August 2011

| Section/ Appendix | Major Changes |
|---|---|
| 5 | Updates the maximum length for token S1 (Gateway Info Token) from 200 to 202. |
| 7 | Adds ASCII formats for token CV (Healthcare/Transit Token) and token CW (Healthcare Service Token). |

# June 2011

| Section/ Appendix | Major Changes |
|---|---|
| 5 | Updates the binary and ASCII formats in the B0 and B1 tokens (Switch tokens) as follows:<br>• Banknet: adds the PMNT-INITIATION-CHAN field; updates the version ID to 15.<br>• Discover ISO: adds the PROC-CDE, POS-SRVC-DATA, TXN-QUAL, and USER-FLD-ACI fields; updates the version ID to 04.<br>• MDS Cirrus ISO: adds the CHRGBCK-IND, PMNT-INITIATE-CHAN, PRMTN-CDE fields; updates the USER-FLD-ACI field; updates the version ID to 15.<br>• Visa Debit Processing Service: adds the VISA-WATCH-LIST fields; updates the version ID to 07.<br>• VisaNet: adds the ORIG-RESP-CDE, ADDL-AMT, ACCT-ID1 fields; updates the USER-FLD-ACI field; updates the version ID to 19.<br>Updates token B4 (EMV Status Token)—binary format: updates the description of the valid values allowed in the TERM-ENTRY-CAP and ARQC-VRFY fields.<br>Updates token BE (Original Currency Release 6.0 token)—binary format: adds the CONV-IND field and modifies the USER-FLD1 field.<br>Updates token BE (Original Currency Release 6.0 token)—ASCII format: adds the CONV-IND field and modifies the USER-FLD1 field.<br>Adds token S8 (PAN Mapping token) as a new Base token. |
| 7 | Updates the CRD-TYP field in token C2 (Purchase Card and Fleet Card Token). A value of A is used to identify an American Express purchasing card.<br>Updates the TXN-STAT-IND field in token C4 (Point of Service Data Token). Adds descriptions for values 2 and 3.<br>Updates the DEV-TYP field in token F1 (E-commerce Additional Data Token). A value of 0 is used to identify PAN entry using a server as well as a mobile device. |

# November 2010

| Section/ Appendix | Major Changes |
|---|---|
| 5 | Corrects the ASCII format positions in token 30 (Issuer Fee Rebate Token).<br><br>Updates the binary and ASCII formats in the B0 and B1 tokens (Switch tokens) as follows:<br><br>• Updates the Star ISO format: adds the CVC2-VALID-PRTCPT-IND, CHRGBCK-ELIGIBILITY-IND, FRAUD-SCORE-RSN-CDE, BUS-APPL-ID, AVS-RSLT-CDE, and VISA-WATCH-LIST fields; updates the version ID to 11.<br><br>• Adds formats for the following interfaces:<br><br>DIAS Interchange Interface (DIAS)<br>Money Station Interface (MONY)<br>NBGC Interchange Interface (NBGC)<br>SPAN2 Interchange Interface (SPAN)<br><br>• Notes that the BIC ANSI and JCBI interfaces do not use the B0/B1 tokens. |
| 7 | Corrects the binary and ASCII format positions in token CH (POS Data1 Token. |

# August 2010

| Section/ Appendix | Major Changes |
|---|---|
| 5 | Updates the binary and ASCII formats in the B0 and B1 tokens (Switch tokens) as follows:<br>• BankNet ISO format: adds FRAUD-DATA fields; updates the version ID to 14.<br>• MDS Cirrus ISO format: adds the FRAUD-DATA, ICHG-RATE-IND, and CVC2-PRG-IND fields; modifies the length of the USER-FLD-ACI field; updates the version ID to 14.<br>• VisaNet ISO format: adds VISA-MONEY-XFER-DATA fields; updates the version ID to 18.<br>• Replaces the duplicate FIID/Version ID table under the ASCII format with a link to the same table under the binary descriptions.<br>Updates the following tokens:<br>• B4 (EMV Status): updates the list of values provided for the TERM-ENTRY-CAP field, position 4.<br>• B5 (EMV Response Data): updates the M/Chip 4 (MCHIP4-ADDL-DATA) definition to correct the update counter values in byte 2, bit positions 2-1.<br>• BE (Original Currency Release 6.0 Token): clarifies the description.<br>• S1 (Gateway Info): adds the ADVC-DAT-TIM field. |
| 7 | Updates the following tokens:<br>• CT (Transaction Specific Data Token): adds a new DATASET-ID value of 5F for Visa Money Transfer Data; adds a new MONEY-XFER-DATA definition to be used to reference data for dataset 5F.<br>• F1 (E-commerce Additional Data Token): adds a new DEV-TYP (e-commerce device type) field. |

# May 2010

| Section/ Appendix | Major Changes |
|---|---|
| 5 | Adds new tokens M1 and M2 for migrating information between BASE24 and BASE24-eps.<br><br>Updates the binary and ASCII formats in the Switch token (acquirer) and Switch token (issuer), tokens B0 and B1, as follows:<br><br>• Adds a table that presents the FIID and version for each interface DDL contained in the token.<br>• Adds the Alaska Option ISO format<br>• Updates the Banknet format to add the following fields for mandates: PIN-CAP-CDE, AUTH-SYS-ADV-DAT-TIM, FRAUD-SCORE, and USER-FLD-ACI<br>• Adds the Cash Station ISO format<br>• Adds the Deluxe ISO format<br>• Adds the Discover ISO format<br>• Adds the FDR ISO format<br>• Adds the MPS format<br>• Adds the NPC ISO format<br>• Adds the NYCE ISO format<br>• Adds the Networks ISO format<br>• Adds the Pulse ISO format<br>• Adds the Star ISO format<br>• Adds the SVS format<br>• Adds the Shazam ISO format<br>• Adds the ValueLink format |
| 7 | Adds new value to the list of valid values in the CRD-TYP field in the Purchase Card and Fleet Card token, token C2.<br><br>Updates the list of valid values for the CRD-VRFY-FLG2 field in the POS Data1 token, token CH to remove the value S. This value indicated that CVV2 should be on the card but the merchant could not find it. |

# January 2010

| Section/ Appendix | Major Changes |
|---|---|
| 5 | Updates the ASCII formats for AXCI ISO, Banknet, MDS, and VisaNet in the Switch token (acquirer) and Switch token (issuer), tokens B0 and B1, to adjust for lengths of non-binary fields. |

# December 2009

| Section/ Appendix | Major Changes |
|---|---|
| 1 | Adds new tokens to the token table list:<br>• Token AV, Bulk Check MICR token<br>• Token AW, Bulk Check Amount token<br>• Token AX, Bulk Check SSB token<br>• Token AY, Bulk Check Disposition token<br>• Token F1, E-commerce Additional Data token |

| Section/ Appendix | Major Changes |
|---|---|
| 5 | Updates description for Surcharge Data token, token 25.<br><br>Corrects value descriptions of Byte 2, position 2-1 for the ARPC-RESP-CDE field in the EMV Response Data token, token B5.<br><br>Adds subtype values to the list of valid values for the TXN-SUBTYP field in the Transaction Subtype token, token BM.<br><br>Updates the TXN-SUBTYP field in the Transaction Subtype token, token BM, to add new subtype ABC0.<br><br>Adds new and redefined values to the CRD-LVL-PROD-ID-VAL field of the Switch Common Data token, token ID BY.<br><br>Updates the binary format for AXCI ISO in the Switch token (acquirer) and Switch token (issuer), tokens B0 and B1 because there is no difference between the AXCI ISO fields in the binary and ASCII tokens.<br><br>Updates the binary and ASCII formats in the Switch token (acquirer) and Switch token (issuer), tokens B0 and B1 for the following interfaces.<br><br>• Adds fields to Banknet.<br>• Adds fields to MDS.<br>• Adds fields to VisaNet. |
| 6 | Adds Bulk Check MICR token, token AV.<br><br>Adds Bulk Check Amount token, token AW.<br><br>Adds Bulk Check SSB token, token AX.<br><br>Adds Bulk Check Disposition token, token AY. |

| Section/ Appendix | Major Changes |
|---|---|
| 7 | Updates S and T values for the E-COM-FLG field in the BASE24-pos Release 5.1 token, token C0.<br><br>Adds new value to the list of valid values for the AUTHN-COLL-IND field in the BASE24-pos Release 5.1 token, token C0.<br><br>Updates the FUEL-TYP field in the VISA-FLEET-DATA redefine of the Purchase Card and Fleet Card token, token C2 to include alphanumeric characters.<br><br>Adds new value to the list of valid values for the PARTIAL-AUTH-OPT field in the POS Data1 token, token CH.<br><br>Adds new values to the list of valid values in the BUS-APPL-ID field for token CT Transaction Specific Data Token Using Extended Datasets.<br><br>Adds new value to the list of valid values for the IIAS-IND field in the Auto-Substantiation Data token, token CY.<br><br>Adds new token E-commerce Additional Data token, token F1. |

*ACI Worldwide, Inc.*

# Preface

This manual contains specifications for BASE24 tokens and token processing. Tokens allow BASE24 processes to easily handle dynamic message formats.

With a fixed message format, any time that additional data is added to a message all processes that receive the message must be modified to recognize the new message format. This requires at a minimum recompiling every process that receives the message, including those that do not use the new data during processing.

With the dynamic message format allowed by tokens, when additional data is required in a message, it is added to the message in the form of a token. Each BASE24 process that receives an internal message has the ability to receive a variable number of tokens with the message. Thus, except to the processes that use the additional data during their processing, no change is required to implement a new token.

This manual provides an introduction to tokens and how they are processed by a BASE24 system, detailed information about the token utilities that are used in token processing, and a description of each token structure.

## Audience

This manual is a reference source for managers, systems analysts, and systems programmers responsible for testing and maintaining a BASE24 system.

# Additional Documentation

The BASE24 documentation set is arranged so that each BASE24 manual presents a topic or group of related topics in detail. When one BASE24 manual presents a topic that has already been covered in detail in another BASE24 manual, the topic is summarized and the reader is directed to the other manual for additional information. Information has been arranged in this manner to be more efficient for readers who do not need the additional detail and at the same time provide the source for readers who require the additional information. This manual contains references to the following BASE24 publications:

- The ***BASE24 External Message Manual*** documents the ISO external message format as used by the ISO Host Interface process, and the handling of token data in that format.
- The ***BASE24 BIC ISO Standards Manual*** documents the ISO external message format as used by the BIC ISO Interface process, and the handling of token data in that format.
- The ***BASE24 Base Files Maintenance Manual*** documents BASE24 file maintenance screen fields, including those for the Token File (TKN).
- The ***BASE24 Refresh and Extract Operators Manual*** discusses fixed-length and variable-length extracts of the transaction log files.

This manual contains references to the following International Organization for Standardization (ISO) publications:

- The ISO 4217 standard, ***Codes for the Representation of Currencies and Funds.***
- The ISO 3166 standard, ***Codes for the Representation of Names of Countries***
- The ISO 8583:1993 standard, ***Bank Card Originated Messages—Interchange Message Specifications—Content for Financial Transactions.***

This manual also contains references to the ***ACI Standard POS Device Message Specifications Manual***.

This manual also contains references to the ***NCR NDC+ CAM2 Functional Specification.***

This manual also contains references to the MasterCard M/Chip and the Visa Smart Debit Credit (VSDC) documentation sets and the EMVCo specification.

## Prerequisites

Tokens can play a substantial role in BASE24 processing. This manual assumes that the reader is familiar with BASE24 products, their basic functions, and their terminology. The product-specific introduction manuals and transaction processing manuals provide this type of background information.

## Software

This manual documents standard processing as of its publication date. Software that is not current and custom software modifications (CSMs) may result in processing that differs from the material presented in this manual. The customer is responsible for identifying and noting these changes.

## Manual Summary

The following is a summary of the contents of this manual.

"Conventions Used in this Manual" follows this preface and describes notation and documentation conventions necessary to understand the information in the manual.

**Section 1, "Introduction,"** introduces the reader to the BASE24 internal message. It includes an overview of message components and structure. It introduces the concept of tokens, and provides an overview of how tokens are used.

**Section 2, "System Token Processing,"** describes the utilities used to build, process, and log tokens.

**Section 3, "User Tokens,"** explains how users can create their own tokens.

**Section 4, "Configuring Internal Message Token Processing,"** describes how to configure the tokens that are logged to various BASE24 transaction log files, that are extracted from those files by the Super Extract process, and that are sent to the host in the external message.

**Section 5, "BASE24 Tokens,"** describes the BASE24 Base message tokens. This section also describes the Header token and the token header.

**Section 6, "BASE24-atm Tokens,"** describes the BASE24-atm message tokens.

**Section 7, "BASE24-pos Tokens,"** describes the BASE24-pos message tokens.

**Section 8, "BASE24-teller Tokens,"** describes the BASE24-teller message tokens.

**Appendix A, "Token Impacts on Log File Disk Space Requirements,"** describes how users can assess the disk space impacts of logging additional token data, and provides examples of how adding token data can affect disk space requirements.

## Publication Identification

Three entries appearing at the bottom of each page uniquely identify this BASE24 publication. The publication number (for example, BA-AE000-12 for the ***BASE24 Tokens Manual***) appears on every page to assist readers in identifying the manual from which a page of information was printed. The publication date (for example, Fall-2013 for Fall, 2013) indicates the issue of the manual. The software release information (for example, R6.0v11 for release 6.0, version 11) specifies the software that the manual describes. This information matches the document information on the copyright page of the manual.

# 1: Introduction

This section provides an introduction to BASE24 tokens. It includes the following topics:

- A discussion of BASE24 internal messages, and how tokens affect the structure of the internal message
- An overview of where tokens are used in BASE24 products
- An introduction to BASE24 token concepts

# BASE24 Internal Messages

When transactions enter a BASE24 system, they are converted to a product-specific standard internal message format. The standard internal message format is recognizable to all BASE24 processes for that product. This standard format contains all of the data required to authorize the transaction, along with a number of fields that are used to route the transaction and indicate its status.

Using the standard internal message, BASE24 processes can handle transactions the same way, regardless of the transaction type or originating source. It also serves to insulate BASE24's internal transaction handling from differences in EFT devices, host transaction handling, and interchange transaction handling.

## Internal Message Conversions

The internal message is converted to and from other message formats by Device Handler processes, Host Interface processes, and Interchange Interface processes.

### Device Handler Processes

When a transaction originates at a device, the respective BASE24 Device Handler process receives the transaction and translates it into the appropriate BASE24 internal message format. Likewise, when a transaction is to be returned to the device, the Device Handler process translates the internal message into a format recognizable to the device. Device Handler processes allow different types of devices to be supported without impacting BASE24 internal transaction processing.

### Host Interface Processes

When a transaction originates from an acquirer host, or is sent to an issuer host for authorization, the Host Interface process handles the translation to and from the standard internal message format. BASE24 can communicate with host processors using an International Organization for Standardization (ISO) based message. For more information on the ISO-based external message, refer to the ***BASE24 External Message Manual***.

## Interchange Interface Processes

When transactions originate from, or must be sent to, an interchange, the Interchange Interface process handles the translation to and from the standard internal message format. The Interchange Interface process allows for communication with an interchange according to interchange requirements and protocols, without affecting BASE24 internal transaction handling.

# Tokens in the Internal Message

In earlier BASE24 releases, the internal message had a static message structure. The structure of the message could not change as it was passed from process to process. Each process received and processed the same structure.

When changes were required (for example, to support new functionality) to this static structure, every process that handled the message had to be modified to recognize the new message structure. Not only did the processes that used the new fields have to be changed so that they could process the additional information, any other process that received the message also had to be changed. At a minimum, every process that received the message needed to be recompiled with the new message structure.

Beginning with BASE24 release 5.0, the internal message structure is dynamic. The message can change as it is passed from process to process, and as it is logged to transaction log files. With the dynamic message format, when additional data is required in a message, it is added to the message in the form of a token. Each BASE24 process that receives an internal message has the ability to receive a variable number of tokens with the message. Thus, except to the processes that use the additional data during their processing, no change is required to implement a new token.

The dynamic BASE24 standard internal messages are composed of a core set of fields included in every message, and a variable number of function-specific sets of fields.

## STM, PSTM, and TSTMH Core Fields

When a transaction enters the BASE24 system, the BASE24 Device Handler process, Host Interface process, or Interchange Interface process creates the core set of fields. These fields contain routing and transaction status information. For the BASE24-atm and BASE24-pos products, the core fields also include many of the fields required to authorize the transaction. This core set of fields is known as the BASE24-atm Standard Internal Message (STM), the BASE24-pos Standard Internal Message (PSTM), or the BASE24-teller Standard Internal Message Header (TSTMH).

## Function-Specific Fields

As the transaction is processed and additional information is required, BASE24 processes add function-specific fields to the message. These function-specific fields are known as tokens. A *token* is a collection of related data required to perform a specific function. A token can be made up of one field or of a series of related fields. Tokens are only added to the message if they are required to process the specific transaction.

# Tokens in Other Structures

In addition to being carried in the internal message, tokens appear in a number of other structures in BASE24 products. These structures are described below.

- ISO-based external messages. The ISO Host Interface process and BIC ISO Interface process can move tokens received in the internal message to the external message to be sent to a host or co-network. The tokens sent in the external message are configurable using the Token File (TKN).
- Processing files. Transaction information is stored in a number of files while transactions are being processed. When information from the internal message is written to these files for approved transactions, token information is also stored in the files.
- Transaction log file records. Transaction log files provide information about each transaction processed by the BASE24 system. The processes that create log file records can include tokens in the information logged to the files. The tokens logged to transaction log files are configurable using the TKN.
- Transaction log file extract records. The Super Extract process can extract token information from the transaction log files. The tokens extracted from transaction log files are configurable using the TKN.

Refer to section 2 for more information about how processes move information from the internal message into these structures. Refer to section 4 for instructions on configuring token use using the TKN.

# Tokens and BASE24-telebanking

The BASE24-telebanking product accepts tokens in the messages it sends and receives, and includes tokens in messages it logs to the ITS Transaction Log File (ITLF). However, unlike the BASE24-atm, BASE24-pos, and BASE24-teller products, the BASE24-telebanking product does not create tokens or use them to authorize transactions.

## Internal Transaction Data

The Internal Transaction Data (ITD), not the BASE24-telebanking standard internal message (BSTM), contains all of the information needed for a transaction while it is being processed by the BASE24-telebanking Integrated Authorization Server process. The Integrated Authorization Server process creates a BSTM from data in the ITD only when it is necessary to use the Host Interface process, when sending data to a host, and when sending a transaction to a Billpay server process. The Integrated Authorization Server process also creates or updates an ITD from information it receives in a BSTM.

The BASE24-telebanking product does not use tokens because, unlike the STM, PSTM, and TSTMH, a field or group of fields can be added or changed in the ITD with minimal effort. As a result, changes are made to the ITD instead of creating or changing tokens when new or different information is needed for BASE24-telebanking processing.

## Token Processing

Token processing by the BASE24-telebanking product is limited to accepting them in messages it sends and receives, and including them in messages it logs to the ITLF. The BASE24-telebanking Integrated Authorization Server process moves any tokens it receives in the BSTM to one of two ITD fields reserved for tokens. The Token field in the ITD contains tokens that will be logged to the ITLF and the Token—Not Logged field in the ITD contains tokens that will not be logged to the ITLF.

The Integrated Authorization Server process uses configuration information in the TKN to determine which ITD field should receive each token. The ISO Host Interface process determines which tokens are placed in messages being sent to the host. The Super Extract process uses configuration information in the TKN to

determine which tokens are extracted from the Token field in the ITLF. Tokens in the Token—Not Logged field cannot be extracted because this field is not in the ITLF.

# Token Basics and Examples

As described previously, the internal message consists of a series of core fields—known as the STM, PSTM, or TSTMH—followed by some number of function-specific tokens. Tokens are only added to the message as they are needed, so it is possible for an internal message to have no tokens associated with it. When the first token is required to process the transaction, the BASE24 system adds two tokens to the message. The first token is the Header token. The second token is whatever token needed to be added to the message. When subsequent tokens are needed, they are added to the message individually. The general layout of an internal message with message tokens is illustrated below.

**Standard Internal Message with Tokens**

| | | | | | |
|---|---|---|---|---|---|
| STM/PSTM/TSTM | Header Token | Token | Token | Token | … |

## Header Token

The Header token contains a count of the number of tokens associated with the message and the overall length of all token data. The Header token is added to the message when the first token is added, and is updated each time a subsequent token is added. The Header token is illustrated below.

**Header Token**

| Eye Catcher | Count | Length |
|---|---|---|
| & | 02 | 30 |

The first field in the Header token contains an eye catcher. The eye catcher makes it easy to locate token information when viewing internal messages. The eye catcher in the Header token is an ampersand (&).

The second field contains the token count. In the example, the token count field contains the value 2. This indicates that there are two tokens in the internal message—the Header token plus one additional token.

The final field contains the overall length of token data. The length includes the total length of the Header token, plus the length of each individual token added to the message.

## Token Headers

Each token that is added to the message has its own token header. Unlike the Header token, which contains information about all tokens in the message, the token header contains information about one specific token. The token header identifies the individual token and contains the binary length of the individual token. The token header is followed by the token data. Together, the token header and the token data form a single token. The general format of a token is illustrated below.

**Data Token**

| Eye Catcher | Token ID | Token Length | Token Data |
|---|---|---|---|
| ! | 13 | 30 | 11101361109261209… |

The first field in the data token is another eye catcher. The eye catcher separates each token in the message from the previous token. The eye catcher in data tokens is always an exclamation point (!).

The second field contains the token ID. Each token is assigned a unique token ID, ranging from 00 through ZZ. This allows for up to 1296 different token IDs. Token IDs in the range P0 through RZ are reserved for user-defined tokens. Token IDs in the ranges 00 through OZ and S0 through WZ are reserved for use by BASE24 products. Token IDs in the range X0 through ZZ are reserved for use by distributors. In the example, the token ID field contains the value 13, which identifies the Credit Line token.

The third field contains the length of the data that follows. The length does not include the length of the token header. Each token must have an even length.

The final portion of the data token is the actual data. The data can be a single field, or a collection of fields. The tokens that are currently defined and the products that use them are listed below.

| ID | Token Name | Product |
|---|---|---|
| N/A | Header token | Base |
| N/A | Token header | Base |
| 01 | Address Verification token | BASE24-pos |
| 02 | Statement Print token | BASE24-atm |

| ID | Token Name | Product |
|---|---|---|
| 03 | BASE24-atm Release 5.0 token | BASE24-atm |
| 04 | BASE24-pos Release 5.0 token | BASE24-pos |
| 05 | Check Guarantee/Verification token | BASE24-pos |
| 06 | PIN Change token | BASE24-atm |
| 07 | Self-Service Banking Base token | BASE24-atm |
| 08 | Customer Name token | Base * |
| 10 | American Express token | BASE24-pos |
| 11 | Automated Clearing House (ACH) Debit token | BASE24-pos |
| 12 | MICR Data token | Base † |
| 13 | Credit Line token | Base * |
| 14 | Self-Service Banking Check token | BASE24-atm |
| 15 | Self-Service Banking Check Terminal Settlement token | BASE24-atm |
| 16 | Alternate Merchant ID token | BASE24-pos |
| 17 | Visa Payment Service 2000 token | BASE24-pos |
| 18 | Account Qualifier token | Base |
| 19 | Visa Payment Service 2000 Offline token | BASE24-pos |
| 20 | Interchange Compliance token | BASE24-pos |
| 21 | PS2000 ATM token | BASE24-atm |
| 22 | Additional Hopper token | BASE24-atm |
| 23 | Track 1 token | Base |
| 24 | ATM Flag 1 token | BASE24-atm |
| 25 | Surcharge Data token | Base |
| 27 | Cardholder Postal Code token | Base |

| ID | Token Name | Product |
|---|---|---|
| 28 | ACI Proactive Risk Manager token | Base |
| 29 | Check Guarantee/Verification 2 token | BASE24-pos |
| 30 | Issuer Fee Rebate token | Base |
| 31 | Check Callback token | BASE24-pos |
| 32 | PRM Real Time token | Base |
| A5 | Non-Currency Dispense token | BASE24-atm |
| A6 | ATM Interchange Compliance token | BASE24-atm |
| A7 | Multiple Account token | BASE24-atm |
| A8 | Bag Deposit token | BASE24-atm |
| A9 | Money Exchange token | BASE24-atm |
| AA | Merchant Banking Center Settlement token | BASE24-atm |
| AB | ATM Balances token | BASE24-atm |
| AD | Cash Acceptor Terminal Settlement token | BASE24-atm |
| AE | Bill Payment Payee List token | BASE24-atm |
| AF | Bill Payment Confirmation token | BASE24-atm |
| AG | ATM Data 1 token | BASE24-atm |
| AH | Multiple Account Inquiry token | BASE24-atm |
| AI | Interim Statement/Passbook Data token | BASE24-atm |
| AJ | Interim Statement Processing token | BASE24-atm |
| AK | Passbook Processing token | BASE24-atm |
| AL | Hold token | BASE24-atm |
| AM | ATM Preferred Transaction token | BASE24-atm |
| AO | Diebold BNA Counts token | BASE24-atm |

| ID | Token Name | Product |
|---|---|---|
| AR | Custom Response Code token | BASE24-atm |
| AS | Shared BNA Counts token | BASE24-atm |
| AT | BNA Multiple Currency token | BASE24-atm |
| AU | Check Bundle token | BASE24-atm |
| AV | Bulk Check MICR token | BASE24-atm |
| AW | Bulk Check Amount token | BASE24-atm |
| AX | Bulk Check SSB token | BASE24-atm |
| AY | Bulk Check Disposition token | BASE24-atm |
| B0 | Switch token (Acquirer) | Base |
| B1 | Switch token (Issuer) | Base |
| B2 | EMV Request Data token | Base |
| B3 | EMV Discretionary Data token | Base |
| B4 | EMV Status token | Base |
| B5 | EMV Response Data token | Base |
| B6 | EMV Script Data token | Base |
| B7 | TLF token | Base |
| B8 | Transaction Profile token | Base |
| B9 | Transaction Description token | Base |
| BA | Acquirer Routing token | Base |
| BB | Pre-Pay Generic Receipt token | Base |
| BC | TSS Index token | Base |
| BD | Multiple Currency token | Base |
| BE | Original Currency 6.0 token | Base |

| ID | Token Name | Product |
|---|---|---|
| BF | Pre-Pay Receipt token | Base |
| BG | Track 3 token | Base |
| BH | Reversal Date and Time token | Base |
| BI | Pre-Pay Top-Up token | Base |
| BJ | EMV Issuer Script Results token | Base |
| BK | Multiple Logical Network token | Base |
| BL | Virtual Primary Account Number token | Base |
| BM | Transaction Subtype token | Base |
| BN | Data Encryption Key token | Base |
| BO | Encrypted Data token | Base |
| BP | Person-to-Person Transaction token | Base |
| BQ | Completion Required token | Base |
| BR | Split Transaction Routing token | Base |
| BS | Pre-Pay Switch token | Base |
| BT | Pre-Pay Response token | Base |
| BU | Pre-Pay Selection token | Base |
| BV | Pre-Pay Voucher Receipt token | Base |
| BW | Pre-Pay Online Receipt token | Base |
| BX | Pre-Pay Original Data token | Base |
| BY | Switch Common Data token | Base |
| C0 | BASE24-pos Release 5.1 token | BASE24-pos |
| C1 | Station ID token | BASE24-pos |
| C2 | Purchase Card and Fleet Card token | BASE24-pos |

| ID | Token Name | Product |
|---|---|---|
| C3 | Certificate token | BASE24-pos |
| C4 | Point of Service Data token | BASE24-pos |
| C5 | Increased Optional Data token | BASE24-pos |
| C6 | Trans Stain XID token | BASE24-pos |
| C7 | Cardholder Serial Number token | BASE24-pos |
| C8 | Merchant Serial Number token | BASE24-pos |
| C9 | MHI Additional Data token | BASE24-pos |
| CA | DUKPT Data token | BASE24-pos |
| CB | POS Balances token | BASE24-pos |
| CE | Authentication Data token | BASE24-pos |
| CF | International Address Verification Service (IAVS) token | BASE24-pos |
| CH | POS Data1 token | BASE24-pos |
| CI | POS Merchant token | BASE24-pos |
| CJ | Pre-Pay Merchant token | BASE24-pos |
| CK | Industry Data token | BASE24-pos |
| CP | Healthcare token | BASE24-pos |
| CQ | Reward Program token | BASE24-pos |
| CR | POS Split Transaction Routing token | BASE24-pos |
| CS | Enhanced Reversal Routing token | BASE24-pos |
| CT | Transaction Specific Data token | BASE24-pos |
| CU | American Express Additional Data token | BASE24-pos |
| CV | Healthcare/Transit token | BASE24-pos |
| CW | Healthcare Service token | BASE24-pos |

| ID | Token Name | Product |
|---|---|---|
| CX | American Express Privte Use Data token | BASE24-pos |
| CY | Auto-Substantiation Data token | BASE24-pos |
| CZ | POS Data 2 token | BASE24-pos |
| F1 | E-commerce Additional Data token | BASE24-pos |
| F2 | Installment Payment Data token | BASE24-pos |
| F3 | Transit Transaction Token | BASE24-pos |
| F4 | Digital Wallet | BASE24-pos |
| M1 | Migration ATM Data1 token | Base |
| M2 | Migration POS Data1 token | Base |
| M4 | Migration EPS HISO token | Base |
| M5 | Migration Customer Data token | Base |
| N8 | Inventory Voucher token | Base |
| S0 | Intra Country Data token | Base |
| S1 | Gateway Info token | Base |
| S2 | Dynamic Currency Conversion (DCC) Status token | Base |
| S3 | Dynamic Currency Conversion (DCC) Processing token | Base |
| S4 | EMV Supplementary Data token | Base |
| S6 | Track2 token | Base |
| S7 | Person-to-Person Transaction 2 token | Base |
| S8 | PAN Mapping token | Base |
| S9 | Additional Authorization Data token | Base |
| SA | Generic Data token | Base |

| ID | Token Name | Product |
|---|---|---|
| T0 | Financial token | BASE24-teller |
| T1 | CAF Inquiry token | BASE24-teller |
| T2 | CAF Update token | BASE24-teller |
| T3 | NBF token | BASE24-teller |
| T4 | PBF Inquiry token | BASE24-teller |
| T5 | PBF Update token | BASE24-teller |
| T6 | SPF Inquiry token | BASE24-teller |
| T7 | SPF Update token | BASE24-teller |
| T8 | WHFF Inquiry token | BASE24-teller |
| T9 | WHFF Update token | BASE24-teller |
| TA | Administrative token | BASE24-teller |
| TB | Account token | BASE24-teller |
| TC | Override token | BASE24-teller |
| TD | PIN token | BASE24-teller |
| TE | Native Message token | BASE24-teller |
| U0 | EBT Voucher token | BASE24-pos |
| U1 | EBT Available Balance token | BASE24-pos |
| U2 | POS Stored Value token | BASE24-pos |

* This token is currently used by BASE24-teller only.

† This token is currently used by BASE24-atm only.

The Base tokens, including the Header token and the token header, are described in section 5. The BASE24-atm tokens are described in section 6. The BASE24-pos tokens are described in section 7. The BASE24-teller tokens are described in section 8.

# ASCII and Binary Token Formats

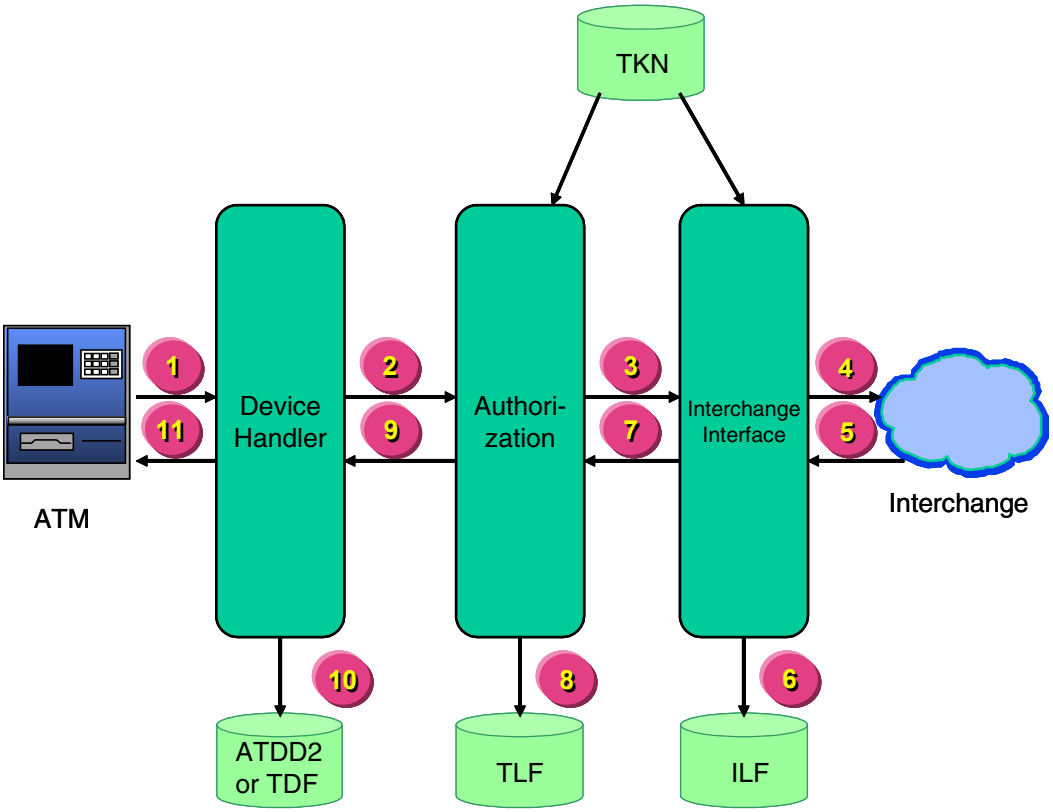
For many tokens, two data structures are defined—an ASCII format and a binary format. The binary format is used in the internal message and when tokens are written to BASE24 files. The ASCII format is used for tokens in the external message. Token data in extract records can be in either binary or ASCII format, depending on the setting for the NUMERIC FLD FORMAT field on screen 1 of the Extract Configuration File (ECF).

The binary and ASCII formats for a token contain the same fields. The only difference is that binary fields are converted to numeric fields in the ASCII format.

All tokens passed in an external message to the BASE24 ISO Host Interface process or to the BASE24 BIC Interchange Interface process are sent in ASCII format. Some of the fields may be converted to binary when the tokens are processed by the interface. If this is the case, any unused fields or parts of fields must contain ASCII zeroes rather than spaces in order to allow the conversion to succeed.

# Example of Message Processing with Tokens

The following is an example of how a message with tokens is processed by the system. This example is for the BASE24-atm product supporting Diebold 10XX/478X or NCR 5XXX terminals.




1. The Device Handler process receives a PIN change rom a device. The transaction requires the PIN Change token. The Device Handler process creates the STM, and then calls the token utilities to add the PIN Change token to the internal message. Since this is the first token to be added to the message, the Header token is added along with the token. The internal message that is sent to the Authorization process has the structure illustrated below.

| | Header Token | | | Token 1 | | | |
|---|---|---|---|---|---|---|---|
| STM | & | 02 | 62 | ! | 06 | 50 | Data |

2. The Authorization process receives the message from the Device Handler process and performs its initial processing of the message. If additional tokens are required at this time, the Authorization process calls the token utilities to add the tokens. In this example, the Authorization process is not

adding any additional tokens. The structure of the message sent to the Interchange Interface process is the same as the message received by the Authorization process.

3. The Interchange Interface process receives the message from the Authorization process and performs its initial processing. If additional tokens are required at this time, the Interchange Interface process calls the token utilities to add the tokens. In this example, the Interchange Interface process is not adding any additional tokens. The Interchange Interface process creates the external message from the internal message, and then suspends the internal message with tokens in memory.
4. The external message is sent to the interchange for processing.
5. The interchange processes the message and returns a response to the Interchange Interface process.
6. When the Interchange Interface process receives the response, it retrieves the suspended internal message with tokens from its extended memory. The Interchange Interface process creates an Interchange Log File (ILF) record for the transaction. To create the ILF record, the Interchange Interface process must determine what tokens, if any, should be logged. This information is specified in the Token File (TKN) records that the Interchange Interface process read at initialization. The log file record contains a record header, selected fields from the internal message, a copy of the external message, and the tokens that were in the message and were configured to be logged. The structure of the ILF record is illustrated below.

| Header | STM | SEM | Header Token | Token Data |
|---|---|---|---|---|

7. The Interchange Interface process sends the internal response message to the Authorization process.
8. The Authorization process receives the response message and creates a Transaction Log File (TLF) record for the transaction. To create the TLF record, the Authorization process must determine what tokens, if any, should be logged. This information is specified in the TKN records that the Authorization process read at initialization. The log file record contains a record header, information from the STM, and the tokens that were in the message and were configured to be logged. The structure of the TLF record is illustrated below.

| Header | STM Fields | Header Token | Token Data |
|---|---|---|---|

9. The Authorization process sends the internal response message to the Device Handler process.

10. The Device Handler process receives the response message from the Authorization process. If the transaction was approved, the Device Handler process saves token data for the message in the BASE24-atm Terminal Data Dynamic File—scratch pad (ATDD2) or the TLF along with certain STM fields in the BASE24-atm Terminal Data Dynamic File—general data (ATDD1). This data is used to create a reversal message, if the transaction does not complete as authorized. If the transaction was denied, the Device Handler process saves only the STM information in the ATDD1. Token data is not saved for denied transactions.

    **Note:** Data entered on BASE24-atm Terminal Data files (ATD) screens is stored in different files, depending on the type of terminal being defined. For Diebold 10XX/478X and NCR 5XXX Device Handler terminals, data is stored in the BASE24-atm Terminal Data Dynamic File—general data (ATDD1), the BASE24-atm Terminal Data Dynamic File—scratch pad (ATDD2), and the BASE24-atm Terminal Data Static File—general data (ATDS1) files. Dynamic data is stored in the ATDD1 file, token data is optionally stored in the ATDD2 file, and static data is stored in the ATDS1 file. For all other types of ATM terminals, all data, including tokens, is stored in the Terminal Data File (TDF).

11. The Device Handler process creates a native-mode message from the internal message and sends it to the device.

*ACI Worldwide, Inc.*

# 2: System Token Processing

This section provides an overview of internal message processing, and provides detailed descriptions of the utilities used to process BASE24 tokens. It includes information on the following topics:

- Determining what tokens are contained in a token buffer
- Adding tokens to a token buffer
- Updating token fields
- Moving tokens into an output buffer
- Sorting tokens in the output buffer
- Deleting tokens from a token buffer
- Converting tokens to another format

# Internal Message Processing Overview

Transactions that route to or through a BASE24 system for authorization do so in a series of transaction messages. Within a BASE24 system, these messages are in the standard internal message format.

As the internal message is routed between processes, each process can update fields in the message to indicate the results of the processing it performed. Some processes store copies of the message in a file for further processing or to retain transaction context. Other processes log a copy of the message to a log file, to record the outcome of message processing. These log records can then be extracted from the file for processing by a host.

When tokens are appended to the internal message, they represent part of the processing that has been performed on the message. No record of the message is complete without taking the token information into consideration. Therefore, any file in which the internal message is stored must also be able to contain token information.

## Storing Message Information in BASE24 Files

Internal message information is stored in the following files:

| Log Files | Processing Files |
|---|---|
| Transaction Log File (TLF) | BASE24-atm Terminal Data files |
| POS Transaction Log File (PTLF) | BASE24-pos Terminal Data files |
| Teller Transaction Log File (TTLF) | Teller Terminal Data File (TTDF) |
| Interchange Log File (ILF) | Administrative Queue File (AQF) |
| ITS Transaction Log File (ITLF) | Terminal Queue File (TQF) |
| | POS Referral File (PRF) |

*Log files* contain historical records of completed transactions. The information in the file is, in general, not used for further authorization processing. You can configure which tokens should be logged to the log files. By logging tokens to the log file, the token information is available for perusal, for reporting, and, using extract, for host processing. The tokens written to the log files are specified using the Token File (TKN).

*Processing files* contain transaction information that is used for further authorization processing of the transaction. The records are used to recreate the internal message, if it becomes necessary (for example, to create a reversal message if the transaction fails to complete as authorized). For processing files, all token information carried in the message for an approved transaction is stored in the file (token data is not stored for denied transactions). By storing all token information in the file, all token information is available when further processing is performed on the message.

## Token Length Limits

Theoretically, a large number of tokens can be carried in the internal message, with no limits on the length of an individual token or on the overall length of all token data. In practice, however, there are limits on the amount of token data that can be added to a message or stored in a file. Limiting factors include HP NonStop operating system record size limitations and SKELB message size limitations.

Users who create their own tokens to be carried in the internal message should be aware that limitations on the amount of token data that can be stored and processed do exist. Currently, the processing files have the tightest limits on the amount of token data that can be handled. If more token data is carried in the message for an approved transaction than can be stored in the appropriate processing file, the Device Handler process stores as much token data as it can and continues processing. For such a transaction, the possibility exists that a complete reversal could not be processed, due to the missing token data.

If a BASE24 process exceeds the amount of space available for token information, as many complete tokens as can be contained in the token buffer are kept in the token buffer. Any token which does not fit in the token buffer is simply discarded. The process logs a message indicating the situation, and processing continues as normal, if possible.

**Note:** Additional disk space may be required to store transaction log files (the BASE24-atm Transaction Log File (TLF), POS Transaction Log File (PTLF), Teller Transaction Log File (TTLF), Interchange Log Files (ILFs), and ITS Transaction Log File (ITLF)), depending on the current record length and the amount of token data you choose to log to the file. For more information on determining the impact of logging token data to a transaction log file, refer to appendix A.

# Processes that Handle Token Data

A number of processes in the BASE24 system handle token data in some manner. The pages that follow identify the processes that use token data, and indicate how token data is used by each process.

## Device Handler Processes

Device Handler processes can send, receive, and process token data in the internal message. In some cases, Device Handler processes add tokens to the message. Device Handler processes also write token data to the terminal data files (BASE24-atm Terminal Data files, BASE24-pos Terminal Data files, or Teller Terminal Data File (TTDF)).

Although BASE24-atm and BASE24-pos Device Handler processes create transaction log file records (that is, TLF or PTLF records) as a result of cutting over a terminal, the Device Handler process does not actually write the record to the file. Instead, the Device Handler process sends the record to the Authorization process, which logs the record. Therefore, the Device Handler process is not involved in logging tokens to the transaction log files for cutover records (also refer to the topic "BASE24-teller Device Handler Processes" below).

### BASE24-pos CRT Authorization and CRT Administration Device Handler Processes

In addition to the token processing described above, the BASE24-pos CRT Authorization Device Handler process writes token data to the Terminal Queue File (TQF), reads token data from the TQF, and reads token data from the POS Referral File (PRF). The BASE24-pos CRT Administration Device Handler process writes token data to the Administrative Queue File (AQF) and reads token data from the AQF.

### BASE24-teller Device Handler Processes

In addition to the token processing described above, BASE24-teller Device Handler processes log token data to the TTLF for log-only transactions. When BASE24-teller Device Handler processes create TTLF records, the Device Handler process must determine which tokens in the internal message should be logged with the record. The Device Handler process retrieves this information from the TKN.

## Authorization Processes

Authorization processes can send, receive, and process token data in the internal message. Authorization processes can also add tokens to the message.

When the Authorization process creates transaction log file records (TLF, PTLF, or TTLF), the Authorization process must determine which tokens in the internal message should be logged with the record. The Authorization process retrieves token logging information from the TKN.

## Integrated Authorization Server Processes

Integrated Authorization Server processes can send and receive token data in the internal message. The Integrated Authorization Server processes move token data between the internal message and fields in the Internal Transaction Data (ITD). However, Integrated Authorization Server processes do not add, change, or delete tokens.

Integrated Authorization Server processes use two ITD fields to control which tokens are written to ITS Transaction Log File (ITLF) records. Token information in one ITD field gets written to the ITLF and token information in the other ITD field does not. The Integrated Authorization Server processes move token information from the internal message to different ITD fields depending on logging information from the TKN.

## ISO Host Interface Processes

ISO Host Interface processes can send, receive, and process token data in the internal or external message. ISO Host Interface processes can also add tokens to the message.

The ISO Host Interface process stores copies of the internal message in extended memory while the external message is being processed by the host. The copies of the internal message include any tokens that are appended to the message.

When the ISO Host Interface process creates an external message, it must determine which tokens in the internal message should be sent to the host and the order in which the tokens should be added to the external message. The ISO Host Interface process retrieves this information from the TKN. The ISO Host Interface process then sorts the tokens into the specified order and adds the tokens to the external message.

When the ISO Host Interface process receives an external message, it must determine which tokens in the external message are not currently present in the internal message. For an incoming request message, the ISO Host Interface process moves all tokens in the external message to the internal message. For an incoming response message, the ISO Host Interface process updates tokens that already exist in the internal message with the new token data from the external message. If a token is present in the external message that was not present in the internal request message, the token is moved to the internal message. For the BASE24-teller product, not all tokens are carried in the token bits of the external message.

**Note:** The ISO Host Interface process performs special processing for the Address Verification and Statement Print tokens (tokens 01 and 02). Within the BASE24 system, the information for these tokens is carried in the BASE24-pos and BASE24-atm standard internal messages (PSTM and STM), respectively—rather than in tokens. When the ISO Host Interface process prepares an external message for sending to a host, it moves the information from the PSTM or STM into the appropriate Address Verification or Statement Print token structure and adds the token to the external message. Conversely, when the ISO Host Interface process receives an external message from the host carrying one of these tokens, it moves the token data into the appropriate PSTM or STM fields and deletes the token.

## Interchange Interface Processes

Interchange Interface processes can send, receive, and process token data in the internal message. Interchange Interface processes can also add tokens to the message.

The Interchange Interface process stores copies of the internal message in extended memory while the external message is being processed by the interchange. The copies of the internal message include any tokens that are appended to the message.

When the Interchange Interface process creates Interchange Log File (ILF) records, the Interchange Interface process must determine which tokens in the internal message should be logged with the record. The Interchange Interface process retrieves this information from the TKN.

Whether the Interchange Interface process supports token data in the external message depends on the specific interchange. For information on how the BIC ISO Interface process supports token data in the external message, refer to the

following paragraphs. For information on how other specific interchanges handle token information in the external message, refer to the informal documentation available on the subvolume where the code for the interface is found.

## BIC ISO Interface Process

In addition to the token processing described above, BIC ISO Interface processes can send, receive, and process token data in the external message.

The BIC ISO Interface process stores copies of the internal message in extended memory while the external message is being processed by the co-network. The copies of the internal message include any tokens that are appended to the message.

When the BIC ISO Interface process creates an external message, it must determine which tokens in the internal message should be sent to the co-network and the order in which the tokens should be placed in the external message. The BIC ISO Interface process retrieves this information from the TKN. The BIC ISO Interface process then sorts the tokens into the specified order and adds the tokens to the external message.

When the BIC ISO Interface process receives an external message, it must determine which tokens in the external message are not currently present in the internal message. For an incoming request message, the BIC ISO Interface process moves all tokens in the external message to the internal message. For an incoming response message, the BIC ISO Interface process updates tokens that already exist in the internal message with the new token data from the external message. If a token is present in the external message that was not present in the internal request message, the token is moved to the internal message.

**Note:** The BIC ISO Interface process performs special processing for the Address Verification and Statement Print tokens (tokens 01 and 02). Within a BASE24 system, the information for these tokens is carried in the BASE24-pos and BASE24-atm standard internal messages (PSTM and STM), respectively—rather than in tokens. When the BIC ISO Interface process prepares an external message for sending to a co-network, it moves the information from the PSTM or STM into the appropriate Address Verification or Statement Print token structure and adds the token to the external message. Conversely, when the BIC ISO Interface process receives an external message from a co-network carrying one of these tokens, it moves the token data into the appropriate PSTM or STM fields and deletes the token.

## Migration Process

New token records should be created for use by the BASE24 HISO Bridge processes so that tokens can be added or removed from messages to and from BASE24-eps without impacting similar messages between BASE24 HISO processes and hosts. This is particularly true for the M1/M2/M4 tokens, which are needed for message exchanges with BASE24-eps but not with hosts.

You may also want to create a specific token group for the BASE24 HISO Bridge processes. In this case, the token group would be defined in the TKN, and then specified in the Host Configuration File (HCF) for the different BASE24 HISO Bridge processes. Processing settings for the TKN are described in the ***BASE24 Base Files Maintenance Manual.***

**Note:** The M1/M2/M4 tokens are used for message exchanges with BASE24-eps; they are not intended to be extracted.

## Super Extract Process

The Super Extract process extracts and sorts token data from the transaction log files (TLF, PTLF, TTLF, ILF, or ITLF). When the Super Extract process creates an extract record, it must determine which tokens in the transaction log file record should be placed in the extract record, and in what order the tokens should be placed. The Super Extract process retrieves this information from the TKN.

## Settlement Initiator Processes

The BASE24-atm Settlement Initiator process reads the TKN to determine which tokens should be logged to the TLF for settlement records (TLF.HEAD.REC-TYP equals 04).

BASE24-pos and BASE24-teller Settlement currently are not required to handle tokens.

## Enscribe Refresh Process

The Enscribe Refresh reads and processes BASE24-teller standard tokens during impacting of the Positive Balance File (PBF), Stop Payment File (SPF), Warning/Hold/Float File (WHFF), and No Book File (NBF). The Enscribe Refresh process also reads and processes the PIN Change token during impacting of the Cardholder Authorization File (CAF).

## Transaction Log File Perusal Subsystems

The transaction log file perusal subsystems (i.e., the requester and server processes used in perusing the TLF, PTLF, and TTLF) retrieve token information from the transaction log files for display on the transaction detail screens, as described below.

The BASE24-atm TLF perusal subsystem retrieves ERR-FLG field information from the BASE24-atm Release 5.0 token.

The BASE24-pos PTLF perusal subsystem retrieves ERR-FLG field information from the BASE24-pos Release 5.0 token and American Express information from the American Express token. For check guarantee and check verification transactions, the PTLF perusal subsystem retrieves information from the Check Guarantee/Verification token and the Alternate Merchant ID token. For electronic benefit transfer transactions, the PTLF perusal subsystem retrieves information from the Voucher Number token.

The BASE24-teller TTLF perusal subsystem retrieves information from the Account token, CAF Inquiry token, CAF Update token, Credit Line token, Financial token, NBF token, PBF Inquiry token, PBF Update token, Override token, SPF Inquiry token, SPF Update token, WHFF Inquiry token, and WHFF Update token.

**Note:** The ITLF perusal subsystem does not retrieve token information from the ITLF for display on the transaction detail screens because the BASE24-telebanking product does not use tokens in transaction processing.

## Report Programs

The BASE24-atm Report programs retrieve information from the Self-Service Banking Base token, MICR Data token, and Self-Service Banking Check token if the BASE24-atm self-service banking (SSB) add-on product is being used.

The BASE24-teller Report programs retrieve information from the CAF Update token, Credit Line token, Financial token, NBF token, Override token, PBF Update token, SPF Update token, and WHFF Update token.

# Token Buffers

Although typically tokens are carried in the internal message and are moved to or from an external message, or logged to a transaction log file from the internal message, in some cases the internal message is not involved in token handling. Tokens can be created and added to other data structures, such as transaction log file records. For the remainder of this section, the term *token buffer* is used to identify any data structure that can include tokens, regardless of whether there are tokens currently in the data structure.

# Token Utilities

BASE24 processes use a number of utilities when handling token data in internal and external messages, and when storing tokens in files. These utilities are listed below:

- TKN^GET^INFO and TKN^GET^INFOX
- TKN^GET^IDS and TKN^GET^IDSX
- TKN^ADD^INFO and TKN^ADD^INFOX
- TKN^UPDT^INFO and TKN^UPDT^INFOX
- TKN^SORT^INFO and TKN^SORT^INFOX
- TKN^LOG^INFO and TKN^LOG^INFOX
- TKN^DEL^INFO and TKN^DEL^INFOX

Each set of utilities performs the same function. The only difference between the utilities in a set is that the utilities with names ending in X use extended pointers instead of integer pointers. Therefore, the utilities are described only once, for instance TKN^GET^INFO and TKN^GET^INFOX are described in the same topic.

In addition to the utilities named above, some processes also use a utility named TKN^MAIN^CONVERT. The TKN^MAIN^CONVERT utility is used to convert tokens between the following formats:

- Converting binary fields in a token from binary to an ASCII display format
- Converting ASCII display format fields to binary
- Converting ASCII (character) fields to EBCDIC
- Creating an initialized buffer for a specific token
- Moving token data from an input buffer to an output buffer

The remainder of this section describes these token utilities in detail.

# Determining Which Tokens Are in the Token Buffer

As processes handle token buffers, they need to determine what tokens are in the token buffer. Processes need this information in order to perform the following functions:

- Store a copy of the token buffer in a file
- Add a token to the token buffer
- Retrieve information from a token to use during processing
- Update information in a token as a result of processing
- Move token information from one place to another (for example, from an internal message to an external message)

Depending on the information needed, processes can use one of two utilities to retrieve token information. These utilities are described below.

## TKN^GET^INFO Utility

The TKN^GET^INFO utility returns the location and length of all token data or of a specific token in the token buffer. Some possible uses of the TKN^GET^INFO utility are described below:

- To determine whether the token buffer includes tokens, and the position at which token data begins in the token buffer.
- To update information in a token. In this case, the process requires the location of the specific token in which fields are being updated. The presence or location of other tokens in the token buffer is not required, because those tokens are not affected by the processing.
- To retrieve information from a token. In this case, the process requires the location of the specific token from which information is being retrieved. The presence or location of other tokens in the token buffer is not required, because those tokens are not affected by the processing.
- To add a specific token to the token buffer. Before a token is added, the process can use this utility to determine whether the token already exists. In this case, the process requires information about whether the specific token is part of the token buffer. The presence or location of other tokens in the token buffer is not required, because those tokens are not affected by the processing.

## TKN^GET^INFO Parameters

The parameters in the call to the TKN^GET^INFO utility are described, in call order, in the table below:

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-PTR | Input | An integer pointer to the buffer that contains the input data. This buffer can contain an internal message, a transaction log file record, or a token data string. |
| IN-BUF-OFST | Input | An integer variable containing the number of bytes the utility should offset into the input buffer to skip over non-token data. This value can be the length of the product-specific standard internal message (STM, PSTM, or TSTMH), the length of the transaction log file record, or the value 0.<br><br>**Note**: For a BASE24-pos standard internal message or BASE24-pos file record, this value should not include the USER-DATA length. The utility calculates the USER-DATA length. |
| IN-BUF-DATA-LGTH | Input | An integer variable containing the current length of data in the input buffer. |
| TKN-ID | Input | The two-byte token ID of the token requested, in ASCII format. This parameter is optional.<br><br>If this parameter is passed, the utility retrieves information about the specific token identified in this parameter.<br><br>If this parameter is not passed, the utility retrieves information for all token data that is present in the input buffer. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-BUF-PTR | Output | An integer pointer to the token data requested.<br><br>If the TKN-ID parameter is included in the call, this parameter points to the first byte of the specific token identified by the value in the TKN-ID parameter.<br><br>If the TKN-ID parameter is not included in the call, this parameter points to the first byte of the Header token. |
| TKN-BUF-LGTH | Output | An integer variable containing the length of the token data.<br><br>If the TKN-ID parameter is included in the call, this parameter contains the length of the specific token identified by the value in the TKN-ID parameter.<br><br>If the TKN-ID parameter is not included in the call, this parameter contains the length of all the token data present in the input buffer. |
| TKN-DSPY-FRMT-FLG | Input | An integer flag indicating whether the token data is in ASCII display format. This parameter is optional.<br><br>If this parameter is included in the call and is set to true, the utility converts the length and count fields in the Header token into integer values to use in its processing.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the length and count fields in the Header token as they are. When the token data is not in ASCII display format, these fields already contain integer values. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| POS-USER-DATA-FLG | Input | An integer flag indicating whether the token buffer contains BASE24-pos USER-DATA. This parameter is optional. It should not be included in the call unless a BASE24-pos standard internal message or a BASE24-pos transaction log file record is passed in the input buffer.<br><br>If this parameter is included in the call and is set to true, the utility calculates a new offset into the input buffer to skip over the USER-DATA.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the value passed in the IN-BUF-OFST parameter as the offset into the input buffer. |

## Status Codes Returned by the TKN^GET^INFO Utility

The TKN^GET^INFO utility returns a status code indicating the outcome of the call to the procedure. Valid status codes are as follows:

0 = Call successful. The utility completed without any errors.
11 = No token data exists in the token buffer.
22 = A parameter or buffer address contained invalid data.
29 = A parameter was missing on the call to the utility.

## TKN^GET^INFO Processing

To retrieve information about all tokens or about a specific token, the TKN^GET^ INFO utility performs the following steps:

1. Verifies that all required parameters are present in the call. If a parameter is missing, the utility sets the status code to 29 and returns to the calling procedure.

2. Skips over any non-token data contained in the token buffer (for example, over the STM).

3. Determines whether there is any token data in the buffer. To determine whether there is token data, the utility performs the following steps:

    a. If the token buffer contains the PSTM or a PTLF record, and the PSTM or PTLF record includes USER-DATA, the utility skips over the USER-DATA.

    b. If the length of the data in the token buffer is less than the number of bytes that the utility is currently offset into the token buffer, a parameter passed to the utility contained invalid data. The utility sets the status code to 22 and returns to the calling procedure.

    c. If the length of the data in the token buffer is equal to the number of bytes that the utility is currently offset into the token buffer, the token buffer does not include any token data. The utility sets the status code to 11 and returns to the calling procedure.

    d. If the length of the data in the token buffer is greater than the number of bytes that the utility is currently offset into the token buffer, the utility checks for the Header token eye catcher (&) at the current offset. If the Header token eye catcher is found at the current offset, the utility continues with the next step (step 4). If the eye catcher is not found, no token data exists. The utility sets the status code to 11 and returns to the calling procedure.

4. If a TKN-ID parameter was not passed to the utility, the utility returns information about all tokens in the token buffer. The utility sets the TKN-BUF-PTR to point to the first byte of the Header token, sets the TKN-BUF-LGTH parameter to the length in the Header token, and returns to the calling procedure with a status of 0, indicating the call was successful.

5. If a TKN-ID parameter was passed to the utility, the utility must return information about the specific token. To return information about the specific token, the utility performs the following steps:

    a. Searches through the token data until the utility finds the token or reaches the end of the token data. The utility compares the token ID for each token in the input buffer to the token ID in the TKN-ID parameter. If the token IDs do not match, the utility skips to the next token. If the token IDs do match, the utility has found the token, and processing continues with the next step (step 5b). If the utility reaches the end of token data without finding the token, the utility sets the status code to 11 and returns to the calling procedure. The utility determines that it has reached the end of the token data by comparing the total number of tokens skipped and the total token length to the token count and length in

the Header token. If there is a discrepancy between these two indicators (for example, the token count indicates there is another token, but the total length indicates the end of the token data has been reached), the utility sets the status code to 22 and returns to the calling procedure.

b. Sets the TKN-BUF-PTR parameter to point to the first byte of the token data, sets the TKN-BUF-LGTH parameter to the length in the token header, and returns to the calling procedure with a status of 0, indicating the call was successful.

# TKN^GET^IDS Utility

The TKN^GET^IDS utility returns the token ID, location, and length of each token in the token buffer. The TKN^GET^IDS utility can be used in the following situations:

- To determine what tokens are in an internal message, so that an external message can be built.
- To determine what tokens are in an external message, so that an internal message can be built.
- To determine what tokens are in a transaction log file record, so that an extract record can be built.

## TKN^GET^IDS Parameters

The parameters in the call to the TKN^GET^IDS utility are described, in call order, in the table below:

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-PTR | Input | An integer pointer to the buffer that contains the input data. This buffer can contain an internal message, a transaction log file record, or a token data string. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-OFST | Input | An integer variable containing the number of bytes the utility should offset into the input buffer to skip over non-token data. This value can be the length of the product-specific standard internal message (STM, PSTM, or TSTMH), the length of the transaction log file record, or the value 0.<br><br>**Note**: For a BASE24-pos standard internal message or BASE24-pos file record, this value should not include the USER-DATA length. The utility calculates the USER-DATA length. |
| IN-BUF-DATA-LGTH | Input | An integer variable containing the current length of data in the input buffer. |
| TKN-ID-ARRAY-PTR | Output | An integer pointer to an array of token information. Each entry in the array represents a single token and contains three fields: a token ID field, which contains the two-byte ASCII representation of the token ID; a token pointer field, which contains an integer pointer to the token data; and a length field, which contains the length of the token data. The calling application is responsible for allocating space for this array. |
| TKN-ID-ARRAY-CNT | Output | An integer variable containing the number of entries in the token array. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| MAX-TKN-ID-ARRAY-ENTRIES | Input | An integer variable containing the maximum number of entries which can be placed in the token array. This parameter allows the calling application to limit the amount of information returned in the array according to the amount of memory that the calling application allocated for the array. |
| TKN-DSPY-FRMT-FLG | Input | An integer flag indicating whether the token data is in ASCII display format. This parameter is optional.<br><br>If this parameter is included in the call and is set to true, the utility converts the length and count fields in the Header token into integer values to use in its processing.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the length and count fields in the Header token as they are. When the token data is not in ASCII display format, these fields already contain integer values. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| POS-USER-DATA-FLG | Input | An integer flag indicating whether the token buffer contains BASE24-pos USER-DATA. This parameter is optional. It should not be included in the call unless a BASE24-pos standard internal message or a BASE24-pos transaction log file record is passed in the input buffer.<br><br>If this parameter is included in the call and is set to true, the utility calculates a new offset into the input buffer to skip over the USER-DATA.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the value passed in the IN-BUF-OFST parameter as the offset into the input buffer. |

## Status Codes Returned by the TKN^GET^IDS Utility

The TKN^GET^IDS utility returns a status code indicating the outcome of the call to the procedure. Valid status codes are as follows:

0 = Call successful. The utility completed without any errors.
11 = No token data exists in the token buffer.
21 = The token caused the token buffer to exceed the maximum allowable length.
22 = A parameter or buffer address contained invalid data.
29 = A parameter was missing on the call to the utility.

## TKN^GET^IDS Processing

To retrieve the token ID, location, and length of each token in the token buffer, the TKN^GET^IDS utility performs the following steps:

1. Verifies that all required parameters are present in the call. If a parameter is missing, the utility sets the status code to 29 and returns to the calling procedure.
2. Skips over any non-token data contained in the token buffer (for example, over the STM).
3. Determines whether there is any token data in the buffer. To determine whether there is token data, the utility performs the following steps:
    a. If the token buffer contains the PSTM or a PTLF record, and the PSTM or PTLF record includes USER-DATA, the utility skips over the USER-DATA.
    b. If the length of the data in the token buffer is less than the number of bytes that the utility is currently offset into the token buffer, a parameter passed to the utility contained invalid data. The utility sets the status code to 22 and returns to the calling procedure.
    c. If the length of the data in the token buffer is equal to the number of bytes that the utility is currently offset into the token buffer, the token buffer does not include any token data. The utility sets the status code to 11 and returns to the calling procedure.
    d. If the length of the data in the token buffer is greater than the number of bytes that the utility is currently offset into the token buffer, the utility checks for the Header token eye catcher (&) at the current offset. If the Header token eye catcher is found at the current offset, the utility continues with the next step (step 4). If the eye catcher is not found, no token data exists. The utility sets the status code to 11 and returns to the calling procedure.
4. Searches through the token data until the utility reaches the end of the token data. The utility determines that it has reached the end of the token data by comparing the total number of tokens processed and the total token length to the token count and length in the Header token. If there is a discrepancy between these two indicators (for example, the token count indicates there is another token, but the total length indicates the end of the token data has been

reached), the utility sets the status code to 22 and returns to the calling procedure. For each token the utility finds, the utility performs the following steps:

a. Determines whether there is room in the output token array to add the token ID, location, and length. If there is room in the array, the utility continues with the next step (step 4b). If there is no room in the array, the utility sets the status code to 21 and returns to the calling procedure.

b. Updates the output token array and count. To update the token array and count, the utility moves the token ID, location, and length to the array. The utility also adds 1 to the array count.

5. Returns to the calling procedure with a status of 0, indicating the call was successful.

# Adding Tokens to the Token Buffer

To add a token to the token buffer, the process must first build the token. The process creates a data buffer that contains the token data. Each field in the buffer is assigned a value—either the default value for the field, or the appropriate value resulting from the processing performed by the process. Once the data buffer is built, the process adds the token using a call to the TKN^ADD^INFO utility.

## TKN^ADD^INFO Parameters

The parameters in the call to the TKN^ADD^INFO utility are described, in call order, in the table below:

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-PTR | Input | An integer pointer to the buffer that contains the input data. This buffer can contain an internal message, a transaction log file record, or a token data string. |
| IN-BUF-OFST | Input | An integer variable containing the number of bytes the utility should offset into the input buffer to skip over non-token data. This value can be the length of the product-specific standard internal message (STM, PSTM, or TSTMH), the length of the transaction log file record, or the value 0.<br>**Note**: For a BASE24-pos standard internal message or BASE24-pos file record, this value should not include the USER-DATA length. The utility calculates the USER-DATA length. |
| IN-BUF-DATA-LGTH | Input | An integer variable containing the current length of data in the input buffer. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-ID | Input | The two-byte token ID of the token being added, in ASCII format. |
| TKN-BUF-PTR | Input | An integer pointer to the token data being added. |
| TKN-BUF-LGTH | Input | An integer variable containing the length of the token data being added. The value in this parameter must be an even number. |
| MAX-TKN-BUF-LGTH | Input | An integer variable containing the maximum length, in bytes, for the buffer pointed to by the IN-BUF-PTR parameter. The MAX-TKN-BUF-LGTH parameter is optional.<br>If this parameter is not included in the call, a default value of 3988 bytes (that is, the maximum number of bytes of data that can be placed in an XPNET process message) is used when length checking is performed. |
| TTL-TKN-DATA-LGTH | Output | An integer variable containing the total length of the token data contained in the input buffer. The length value includes the length of the token that was added, if the token was added successfully. This parameter is optional.<br>If the calling procedure does not require the token data length after adding a token to an existing buffer, this parameter does not need to be included in the call. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-DSPY-FRMT-FLG | Input | An integer flag indicating whether the token data is in ASCII display format. This parameter is optional.<br><br>If this parameter is included in the call and is set to true, the utility converts the length and count fields in the Header token into integer values to use in its processing.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the length and count fields in the Header token as they are. When the token data is not in ASCII display format, these fields already contain integer values. |
| TKN-EBCDIC-FLG | Input | An integer flag indicating whether the token character data will be translated to an EBCDIC format in the output buffer. This parameter is optional.<br><br>If this parameter is present and is set to true, the utility converts the character fields in the token header (and in the Header token, if the token being added is the first token to be added to the token buffer) to EBCDIC.<br><br>If this parameter is not present, or is present and is set to false, the utility does not convert the character fields in the token header (or Header token) to EBCDIC. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-EBCDIC-FLG *continued* | Input | **Note**: This parameter has no affect on the token data in the input buffer. In the output buffer, this parameter only affects the format of character fields in the Header token and token header. It does not affect character fields within the token itself. The character data in each token is translated to EBCDIC using a call to the TKN^MAIN^ CONVERT utility; this call must be made separately. |
| POS-USER-DATA-FLG | Input | An integer flag indicating whether the token buffer contains BASE24-pos USER-DATA. This parameter is optional. It should not be included in the call unless a BASE24-pos standard internal message or a BASE24-pos transaction log file record is passed in the input buffer.<br><br>If this parameter is included in the call and is set to true, the utility calculates a new offset into the input buffer to skip over the USER-DATA.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the value passed in the IN-BUF-OFST parameter as the offset into the input buffer. |
| TKN-DATA-PTR | Output | An integer pointer to the token data after it is added to the token buffer. This parameter is optional. |

## Status Codes Returned by the TKN^ADD^INFO Utility

The TKN^ADD^INFO utility returns a status code indicating the outcome of the call to the procedure. Valid status codes are as follows:

0 = Call successful. The utility completed without any errors.
10 = The specified token already exists.
21 = The token caused the token buffer to exceed the maximum allowable length.
22 = A parameter or buffer address contained invalid data.
29 = A parameter was missing on the call to the utility.

## TKN^ADD^INFO Processing

To add the token, the TKN^ADD^INFO utility performs the following steps:

1. Verifies that all required parameters are present in the call. If a parameter is missing, the utility sets the status code to 29 and returns to the calling procedure.
2. Verifies that the token to be added has an even, non-zero length. If the token has an odd length or a length of zero, the utility sets the status code to 22 and returns to the calling procedure.
3. Skips over any non-token data contained in the token buffer (for example, over the STM).
4. Determines whether there is any token data in the buffer. To determine whether there is token data, the utility performs the following steps:
    a. If the token buffer contains the PSTM or a PTLF record, and the PSTM or PTLF record includes USER-DATA, the utility skips over the USER-DATA.
    b. If the length of the data in the token buffer is less than the number of bytes that the utility is currently offset into the token buffer, a parameter passed to the utility contained invalid data. The utility sets the status code to 22 and returns to the calling procedure.
    c. If the length of the data in the token buffer is equal to the number of bytes that the utility is currently offset into the token buffer, the token buffer does not include any token data. In this case, the utility skips to step 5.
    d. If the length of the data in the token buffer is greater than the number of bytes that the utility is currently offset into the token buffer, the utility checks for the Header token eye catcher (&) at the current offset.

If the Header token eye catcher is found at the current offset, the utility continues with step 11.

If the Header token eye catcher is not found at the current offset, no token data exists. Processing continues with the next step (step 5).

5. Determines whether the token and the Header token can be added to the token buffer without exceeding the maximum token buffer length. If adding the token and the Header token to the token buffer exceeds the maximum length, the utility sets the status code to 21 and returns to the calling procedure.
6. Adds the Header token to the token buffer. The utility sets the CNT field to 2 and the LGTH field to the sum of the Header token length, the token data to be added, and the token header length.
7. Adds the token to the token buffer. The utility creates a token header that contains the token ID and token length and places it in the token buffer following the Header token. The utility then moves the token data to the token buffer.
8. If the TTL-TKN-DATA-LGTH parameter was included in the call to the utility, the utility sets the parameter to the length in the Header token.
9. If the TKN-DATA-PTR parameter was included in the call to the utility, the utility sets the parameter to point to the first byte of the token data for the token just added to the token buffer.
10. Sets the status code to 0 and returns to the calling procedure.
11. Determines if and where the token data can be added. To determine if and where the token can be added, the utility performs the following steps:
    a. Checks each token ID in the token buffer to determine whether the token being added is already present in the token buffer. If the token is already in the token buffer, the utility sets the status code to 10 and returns to the calling procedure.
    b. Points to the end of the data in the token buffer. The utility skips over each token using the token length from the token header. As the utility skips over each token, the utility adds one to the token count and calculates the sum of all the token lengths. The utility determines that it has reached the end of the token data by comparing the total number of tokens skipped and the total token length to the token count and length in the Header token. If there is a discrepancy between these two indicators (for example, the token count indicates there is another token, but the total length indicates the end of the token data has been reached), the utility sets the status code to 22 and returns to the calling procedure.

c. Determines whether the token can be added to the token buffer without exceeding the maximum token buffer length. If adding the token to the token buffer exceeds the maximum length, the utility sets the status code to 21 and returns to the calling procedure.

d. Adds the token to the token buffer. The utility creates a token header that contains the token ID and token length and places it in the token buffer following the last token in the buffer. The utility then moves the token data to the token buffer.

e. Updates the Header token by adding 1 to the count field and the token length (token header and token data) to the total token length field.

f. If the TTL-TKN-DATA-LGTH parameter was included in the call to the utility, the utility sets the parameter to the length in the Header token.

g. If the TKN-DATA-PTR parameter was included in the call to the utility, the utility sets the parameter to point to the first byte of the token data for the token just added to the token buffer.

h. Sets the status code to 0 and returns to the calling procedure.

# Updating Existing Tokens

Tokens are updated as processing requires. Depending on the impact of the update to the token length, tokens can either be updated in place, or using a call to the TKN^UPDT^INFO utility.

A process that needs to update a token where the update does not involve changing the token length can simply update the token in place. The process calls the TKN^GET^INFO utility to determine the location of the token in the token buffer. Once the location of the token is known, the process can update the value in any field in the token, as long as the changes to the token data do not change the length of the token data.

To update a token where the update involves changing the length of the token data, the process calls the TKN^UPDT^INFO utility. This utility updates the token and moves the surrounding token data as required.

## TKN^UPDT^INFO Parameters

The parameters in the call to the TKN^UPDT^INFO utility are described, in call order, in the table below:

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-PTR | Input | An integer pointer to the buffer that contains the input data. This buffer can contain an internal message, a transaction log file record, or a token data string. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-OFST | Input | An integer variable containing the number of bytes the utility should offset into the input buffer to skip over non-token data. This value can be the length of the product-specific standard internal message (STM, PSTM, or TSTMH), the length of the transaction log file record, or the value 0.<br><br>**Note**: For a BASE24-pos standard internal message or BASE24-pos file record, this value should not include the USER-DATA length. The utility calculates the USER-DATA length. |
| IN-BUF-DATA-LGTH | Input | An integer variable containing the current length of data in the input buffer. |
| TKN-ID | Input | The two-byte token ID of the token being updated, in ASCII format. |
| TKN-BUF-PTR | Input | An integer pointer to the token data being updated. This parameter contains the location of the new token data to use when updating the token. |
| TKN-BUF-LGTH | Input | An integer variable containing the length of the token data being updated. |
| MAX-TKN-BUF-LGTH | Input | An integer variable containing the maximum length, in bytes, for the buffer pointed to by the IN-BUF-PTR parameter. The MAX-TKN-BUF-LGTH parameter is optional.<br><br>If this parameter is not included in the call, a default value of 3988 bytes (that is, the maximum number of bytes of data that can be placed in an XPNET process message) is used when length checking is performed. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TTL-TKN-DATA-LGTH | Output | An integer variable containing the total length of the token data contained in the input buffer. The length value includes the length of the token that was updated, if the token was updated successfully. This parameter is optional.<br><br>If the calling procedure does not require the token data length after updating a token, this parameter does not need to be included in the call. |
| TKN-DSPY-FRMT-FLG | Input | An integer flag indicating whether the token data is in ASCII display format. This parameter is optional.<br><br>If this parameter is included in the call and is set to true, the utility converts the length and count fields in the Header token into integer values to use in its processing.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the length and count fields in the Header token as they are. When the token data is not in ASCII display format, these fields already contain integer values. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| POS-USER-DATA-FLG | Input | An integer flag indicating whether the token buffer contains BASE24-pos USER-DATA. This parameter is optional. It should not be included in the call unless a BASE24-pos standard internal message or a BASE24-pos transaction log file record is passed in the input buffer.<br><br>If this parameter is included in the call and is set to true, the utility calculates a new offset into the input buffer to skip over the USER-DATA.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the value passed in the IN-BUF-OFST parameter as the offset into the input buffer. |
| TKN-DATA-PTR | Output | An integer pointer to the token data after it is updated in the token buffer. This parameter is optional. |

## Status Codes Returned by the TKN^UPDT^INFO Utility

The TKN^UPDT^INFO utility returns a status code indicating the outcome of the call to the procedure. Valid status codes are as follows:

0 = Call successful. The utility completed without any errors.
11 = No token data exists in the token buffer.
21 = The token caused the token buffer to exceed the maximum allowable length.
22 = A parameter or buffer address contained invalid data.
29 = A parameter was missing on the call to the utility.

## TKN^UPDT^INFO Processing

To update the token, the TKN^UPDT^INFO utility performs the following steps:

1. Verifies that all required parameters are present in the call. If a parameter is missing, the utility sets the status code to 29 and returns to the calling procedure.
2. Verifies that the token to be updated has an even, non-zero length. If the token has an odd length or a length of zero, the utility sets the status code to 22 and returns to the calling procedure.
3. Skips over any non-token data contained in the token buffer (for example, over the STM).
4. Determines whether there is any token data in the buffer. To determine whether there is token data, the utility performs the following steps:
    a. If the token buffer contains the PSTM or a PTLF record, and the PSTM or PTLF record includes USER-DATA, the utility skips over the USER-DATA.
    b. If the length of the data in the token buffer is less than the number of bytes that the utility is currently offset into the token buffer, a parameter passed to the utility contained invalid data. The utility sets the status code to 22 and returns to the calling procedure.
    c. If the length of the data in the token buffer is equal to the number of bytes that the utility is currently offset into the token buffer, the token buffer does not include any token data. The utility sets the status code to 11 and returns to the calling procedure.
    d. If the length of the data in the token buffer is greater than the number of bytes that the utility is currently offset into the token buffer, the utility checks for the Header token eye catcher (&) at the current offset. If the Header token eye catcher is found, processing continues with step 5. If the Header token eye catcher is not found at the current offset, the utility sets the status code to 11 and returns to the calling procedure.
5. Determines whether the token can be updated. To determine whether the token can be updated, the utility performs the following steps:
    a. Checks each token ID in the token buffer to determine whether the token being updated is present in the token buffer. If the token IDs match, processing continues with the next step (step 5b). If the token IDs do not match, the utility adds one to the token count and calculates the sum of all the token lengths. If the utility reaches the end of the token data in the token buffer without finding the token, the utility sets the status code

to 11 and returns to the calling procedure. The utility determines that it has reached the end of the token data by comparing the total number of tokens skipped and the total token length to the token count and length in the Header token. If there is a discrepancy between these two indicators (for example, the token count indicates there is another token, but the total length indicates the end of the token data has been reached), the utility sets the status code to 22 and returns to the calling procedure.

b. Determines whether the token can be updated without exceeding the maximum token buffer length. If updating the token exceeds the maximum length, the utility sets the status code to 21 and returns to the calling procedure.

c. Updates the token. To update the token the utility performs the following steps:

   1) If the new token data is the same length as the current token data, the utility replaces the current token data with the new token data. Note that in this case, the process could have updated the token in place, without calling the TKN^UPDT^INFO utility.

   2) If the new token data is longer than the current token data, the utility shifts all remaining token data in the token buffer to the right by the number of bytes that the new token data is longer. After the token data is shifted to the right, the utility replaces the current token data with the new token data, updates the token length field in the token header, and updates the total token length field in the Header token to reflect the new total token length.

   3) If the new token data is shorter than the current token data, the utility shifts all remaining token data in the token buffer to the left by the number of bytes that the new token data is shorter. After the token data is shifted to the left, the utility replaces the current token data with the new token data, updates the token length field in the token header, and updates the total token length field in the Header token to reflect the new total token length.

   4) If the TTL-TKN-DATA-LGTH parameter was included in the call to the utility, the utility sets the parameter to the length in the Header token.

   5) If the TKN-DATA-PTR parameter was included in the call to the utility, the utility sets the parameter to point to the first byte of token data for the token just updated in the token buffer. It also sets the status code to 0 and returns to the calling procedure.

# Moving Tokens to a Different Token Buffer

Tokens in BASE24 token buffers are moved to other token buffers to create an external message from an internal message and to create extract records from transaction log file records. Tokens are moved from one token buffer to another using a call to the TKN^SORT^INFO utility. The TKN^SORT^INFO utility performs the following three major functions:

- Moves selected tokens from the token buffer to an output buffer. The tokens to be moved are specified in the TKN. If a TKN record is not passed on the call to the utility, all tokens in the token buffer are moved to the output buffer.
- Sorts the tokens into the order specified in the TKN. If the TKN record is not passed to the utility, the tokens are placed in the output buffer in the same order in which they occur in the token buffer.
- Translates the Header token and token headers into display format or EBCDIC, if required.

## TKN^SORT^INFO Parameters

The parameters in the call to the TKN^SORT^INFO utility are described, in call order, in the table below:

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-PTR | Input | An integer pointer to the buffer that contains the input data. This buffer can contain an internal message, a transaction log file record, or a token data string. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-OFST | Input | An integer variable containing the number of bytes the utility should offset into the input buffer to skip over non-token data. This value can be the length of the product-specific standard internal message (STM, PSTM, or TSTMH), the length of the transaction log file record, or the value 0.<br><br>**Note**: For a BASE24-pos standard internal message or BASE24-pos file record, this value should not include the USER-DATA length. The utility calculates the USER-DATA length. |
| IN-BUF-DATA-LGTH | Input | An integer variable containing the current length of data in the input buffer. |
| TKN-ORDR-FLG | Input | An integer flag indicating whether the token data needs to be in a specific order in the output buffer. This parameter is required.<br><br>If this parameter is set to a value of true, the utility sorts the data into the specified order. When this parameter is set to true, the TKN-ID-ARRAY-PTR and TKN-ID-ARRAY-CNT parameters are required.<br><br>If this parameter is set to a value of false, the order of tokens in the output buffer is not important. The utility moves all token data from the input buffer to the output buffer. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-ID-ARRAY-PTR | Input | An integer pointer to an array of token IDs. The array identifies the tokens that should be placed in the output buffer. This parameter is required if the TKN-ORDR-FLG parameter is set to the value true. |
| TKN-ID-ARRAY-CNT | Input | An integer variable containing the number of entries in the token array pointed to by the TKN-ID-ARRAY-PTR parameter. The TKN-ID-ARRAY-CNT parameter is required if the TKN-ID-ARRAY-PTR parameter is included in the call. |
| MAX-TKN-BUF-LGTH | Input | An integer variable containing the maximum length, in bytes, for the buffer pointed to by the TKN-BUF-PTR parameter. The MAX-TKN-BUF-LGTH parameter is required. |
| TKN-BUF-PTR | Input | An integer pointer to the start of the output buffer. This parameter is required if the OUT-TKN-ID-ARRAY-PTR and OUT-TKN-ID-ARRAY-CNT parameters are **not** included in the call. |
| TKN-BUF-LGTH | Output | An integer variable containing the length of the token data in the output buffer. This parameter is required if the OUT-TKN-ID-ARRAY-PTR and OUT-TKN-ID-ARRAY-CNT parameters are **not** included in the call. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| OUT-TKN-ID-ARRAY-PTR | Output | An integer pointer to an array of token information. Each entry in the array contains three fields: a token ID field, which contains the two-byte ASCII representation of a token ID, a token pointer field, which contains an integer pointer to the token data in the output buffer, and a length field, which contains the length of the token data. The calling procedure is responsible for allocating space for this array.<br><br>If the TKN-EBCDIC-FLG parameter is present and set to true, the information in the array pertains to the tokens in the output buffer.<br><br>If the TKN-EBCDIC-FLG parameter is present and set to false or is not present in the call, the information in the array pertains to the tokens in the input buffer.<br><br>If this parameter is included in the call, the OUT-TKN-ID-ARRAY-CNT parameter is required. |
| OUT-TKN-ID-ARRAY-CNT | Output | An integer variable containing the number of entries in the array pointed to by the OUT-TKN-ID-ARRAY-PTR parameter. If the OUT-TKN-ID-ARRAY-PTR parameter is included in the call, the OUT-TKN-ID-ARRAY-CNT parameter is required, otherwise this parameter is optional. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| MAX-OUT-TKN-ID-ARRAY-ENTRIES | Input | An integer variable containing the maximum number of entries which can be placed in the array pointed to by the OUT-TKN-ID-ARRAY-PTR parameter. The MAX-OUT-TKN-ID-ARRAY-ENTRIES parameter allows the calling application to limit the amount of information returned in the array according to the amount of memory that the calling application allocated for the array. This parameter is optional, and should only be included in the call if the OUT-TKN-ID-ARRAY-PTR parameter is included in the call. |
| IN-TKN-DSPY-FRMT-FLG | Input | An integer flag indicating whether the token data in the input buffer is in ASCII display format. This parameter is optional.<br><br>If this parameter is included in the call and is set to true, the utility assumes the data in the input buffer is in ASCII display format. The utility verifies that the OUT-TKN-DSPY-FRMT-FLG parameter is also present and set to true. If the OUT-TKN-DSPY-FRMT-FLG parameter is not present or is set to the value false, the utility returns a status code of 29.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility assumes the data in the input buffer is in binary format. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| OUT-TKN-DSPY-FRMT-FLG | Input | An integer flag indicating whether the token data in the output buffer is in ASCII display format. This parameter is optional.<br><br>If this parameter is included in the call and is set to true, the utility checks the value in the IN-TKN-DSPY-FRMT-FLG parameter. If the IN-TKN-DSPY-FRMT-FLG parameter is set to the value true, the token data in the input buffer is in ASCII display format. Since the output buffer should also be in ASCII display format, the utility uses the length and count fields in the Header token and token headers as they are. If the IN-TKN-DSPY-FRMT-FLG parameter is set to the value false, the token data in the input buffer is in binary format. In this case, the utility converts the length and count fields in the Header token and token headers into ASCII values in the output buffer.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the length and count fields in the Header token as they are. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-EBCDIC-FLG | Input | An integer flag indicating whether the token character data will be translated to an EBCDIC format in the output buffer. This parameter is optional.<br><br>If this parameter is present and set to true, the utility converts the character fields in the token header (and in the Header token, if the token being added is the first token to be added to the token buffer) to EBCDIC.<br><br>If this parameter is not present, or is present and is set to false, the utility does not convert the character fields in the token header (or Header token) to EBCDIC.<br><br>**Note**: This parameter has no affect on the token data in the input buffer. In the output buffer, this parameter only affects the format of character fields in the Header token and token header. It does not affect character fields within the token itself. The character data in each token is translated to EBCDIC using a call to the TKN^MAIN^ CONVERT utility; this call must be made separately. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| POS-USER-DATA-FLG | Input | An integer flag indicating whether the token buffer contains BASE24-pos USER-DATA. This parameter is optional. It should not be included in the call unless a BASE24-pos standard internal message or a BASE24-pos transaction log file record is passed in the input buffer.<br><br>If this parameter is included in the call and is set to true, the utility calculates a new offset into the input buffer to skip over the USER-DATA.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the value passed in the IN-BUF-OFST parameter as the offset into the input buffer. |

## Status Codes Returned by the TKN^SORT^INFO Utility

The TKN^SORT^INFO utility returns a status code indicating the outcome of the call to the procedure. Valid status codes are as follows:

0 = Call successful. The utility completed without any errors.
11 = No token data exists in the token buffer.
21 = The token caused the token buffer to exceed the maximum allowable length.
22 = A parameter or buffer address contained invalid data.
29 = A parameter was missing on the call to the utility.

## TKN^SORT^INFO Processing

To move token data to the output buffer, the TKN^SORT^INFO utility performs the following steps:

1. Verifies that all required parameters are present in the call. If a parameter is missing, the utility sets the status code to 29 and returns to the calling procedure.
2. Skips over any non-token data contained in the token buffer (for example, over the STM).
3. Determines whether there is any token data in the buffer. To determine whether there is token data, the utility performs the following steps:
    a. If the token buffer contains the PSTM or a PTLF record, and the PSTM or PTLF record includes USER-DATA, the utility skips over the USER-DATA.
    b. If the length of the data in the token buffer is less than the number of bytes that the utility is currently offset into the token buffer, a parameter passed to the utility contained invalid data. The utility sets the status code to 22 and returns to the calling procedure.
    c. If the length of the data in the token buffer is equal to the number of bytes that the utility is currently offset into the token buffer, the token buffer does not include any token data. The utility sets the status code to 11 and returns to the calling procedure.
    d. If the length of the data in the token buffer is greater than the number of bytes that the utility is currently offset into the token buffer, the utility checks for the Header token eye catcher (&) at the current offset. If the Header token eye catcher is found, the utility continues with step 4. If the Header token eye catcher is not found at the current offset, the utility sets the status code to 11 and returns to the calling procedure.
4. If the TKN-ORDR-FLG parameter is set to false and the OUT-TKN-DSPY-FRMT-FLG is set to false, indicating that the token data is not to be sorted into a specific order and is not to be converted to ASCII format, the utility moves all tokens, including the Header token, into the output buffer. To move all tokens, the utility performs the following steps:
    a. Determines whether there is sufficient room in the output buffer for the token data. If there is sufficient room, the utility continues to step 4b. If moving the token data will cause the maximum output buffer length to be exceeded, the utility sets the status code to 21 and returns to the calling procedure.

b. Sets the TKN-BUF-PTR parameter to point to the token data. The tokens are moved as a group to the output buffer, not as individual tokens.

c. Sets the status code to 0 and returns to the calling procedure. Processing continues with step 7.

If the above conditions are not met, processing continues with step 5.

5. If the TKN-ORDR-FLG parameter is set to false and the OUT-TKN-DSPY-FRMT-FLG parameter is set to true, indicating that the token data is not to be sorted into a specific order and is to be converted to ASCII format, the utility moves all tokens, including the Header token, into the output buffer. To move each token, the utility performs the following steps:

   a. If the TKN-BUF-PTR parameter was included in the call, the utility determines whether there is sufficient room in the output buffer for the token. If there is sufficient room, the utility continues to step 5b. If moving the token will cause the maximum output buffer length to be exceeded, the utility sets the total token length for the output buffer to the length specified in the output buffer Header token, sets the status code to 21, and returns to the calling procedure.

   b. If the TKN-BUF-PTR parameter was included in the call, the utility adds the token to the token buffer as follows:

      1) If this is the first token to be added to the buffer, the utility creates and adds the Header token to the token buffer. The utility sets the CNT field to 2 and the LGTH field to the sum of the Header token length, the token data to be added, and the token header length. If the IN-TKN-DSPY-FRMT-FLG parameter was set to false or was not passed, the utility converts the length and count fields to ASCII values in the output buffer.

      2) Adds the token to the token buffer. The utility creates a token header that contains the token ID and token length and places it in the token buffer following the last token in the buffer. Since the OUT-TKN-DSPY-FRMT-FLG parameter was passed and was set to true, the utility converts the length field in the token header to an ASCII value in the output buffer. The utility then moves the token data to the token buffer.

      3) If this was not the first token to be added to the buffer, the utility updates the Header token by adding 1 to the count field and the token length (token header and token data) to the total token length field.

4) Updates the output token array and count. This step is only performed if the OUT-TKN-ID-ARRAY-PTR and OUT-TKN-ID-ARRAY-CNT parameters were included in the call. If the token array is full, the utility sets the total token length for the output buffer to the length specified in the output buffer Header token, sets the status code to 21, and returns to the calling procedure. If the token array is not full, the utility moves the token ID to the array and sets the data pointer to the location of the token data in the output buffer. The utility also adds 1 to the array count.

5) If there are more tokens to be moved, returns to step 5a. If there are no more tokens to be moved, continues with step 7.

If the above conditions are not met, processing continues with step 6.

6. If the TKN-ORDR-FLG parameter is set to true, indicating that the token data is to be sorted into a specific order, the utility performs the following steps for each token identified in the TKN record in the order they are specified:

   a. Searches through the token data until the utility finds the token or reaches the end of the token data. If the utility finds the token, the utility continues with step 6b. The utility determines that it has reached the end of the token data by comparing the total number of tokens skipped and the total token length to the token count and length in the Header token. If there is a discrepancy between these two indicators (for example, the token count indicates there is another token, but the total length indicates the end of the token data has been reached), the utility sets the status code to 22 and returns to the calling procedure. If the utility does not find the token before it reaches the end of the token data, it repeats this step, looking for the next token ID.

   b. If the TKN-BUF-PTR parameter was passed in the call, the utility determines whether there is sufficient room in the output buffer for the token. If there is sufficient room, the utility continues with step 6c. If moving the token will cause the maximum output buffer length to be exceeded, the utility sets the total token length for the output buffer to the length specified in the output buffer Header token, sets the status code to 21, and returns to the calling procedure.

   c. If the TKN-BUF-PTR parameter was passed in the call to the utility, the utility performs the following steps to add the token to the token buffer:

      1) If this is the first token to be added to the buffer, the utility creates and adds the Header token to the token buffer. The utility sets the CNT field to 2 and the LGTH field to the sum of the Header token length, the token data to be added, and the token header length. If the IN-TKN-DSPY-FRMT-FLG parameter was not passed or was

set to false, and if the OUT-TKN-DSPY-FRMT-FLG parameter was passed and was set to true, the utility converts the length and count fields in the Header token to an ASCII value in the output buffer.

2) Adds the token to the token buffer. The utility creates a token header that contains the token ID and token length and places it in the token buffer following the last token in the buffer. If the IN-TKN-DSPY-FRMT-FLG parameter was not passed or was set to false, and if the OUT-TKN-DSPY-FRMT-FLG parameter was passed and was set to true, the utility converts the length field in the token header to an ASCII value in the output buffer. The utility then moves the token data to the token buffer.

3) If this was not the first token to be added to the buffer, the utility updates the Header token by adding 1 to the count field and the token length (token header and token data) to the total token length field.

4) Updates the output token array and count. This step is only performed if the OUT-TKN-ID-ARRAY-PTR and OUT-TKN-ID-ARRAY-CNT parameters were included in the call. If the token array is full, the utility sets the total token length for the output buffer to the length specified in the output buffer Header token, sets the status code to 21, and returns to the calling procedure. If the token array is not full, the utility moves the token ID to the array and sets the data pointer to the location of the token data in the output buffer. The utility also adds 1 to the array count.

7. If the TKN-BUF-LGTH parameter was passed in the call to the utility, the utility updates the output buffer total token length with the total token length from the output buffer Header token once all tokens have been moved to the output buffer.

8. Sets the status code to 0 and returns to the calling procedure.

# Logging Tokens to Token Buffer

When records are written to the log files (TLF, PTLF, TTLF, ILF, and ITLF) and processing files (BASE24-atm Terminal Data files, BASE24-pos Terminal Data files, TTDF, AQF, TQF, and PRF), token data is written with the record. Tokens are written to a file using a call to the TKN^LOG^INFO utility.

## TKN^LOG^INFO Parameters

The parameters in the call to the TKN^LOG^INFO utility are described, in call order, in the table below:

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-PTR | Input | An integer pointer to the buffer that contains the input data. This buffer can contain an internal message, a transaction log file record, or a token data string. |
| IN-BUF-OFST | Input | An integer variable containing the number of bytes the utility should offset into the input buffer to skip over non-token data. This value can be the length of the product-specific standard internal message (STM, PSTM, or TSTMH), the length of the transaction log file record, or the value 0.<br><br>**Note**: For a BASE24-pos standard internal message or BASE24-pos file record, this value should not include the USER-DATA length. The utility calculates the USER-DATA length. |
| IN-BUF-DATA-LGTH | Input | An integer variable containing the current length of data in the input buffer. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-ID-ARRAY-PTR | Input | An integer pointer to an array of token IDs. The array identifies the tokens that should **not** be placed in the log file record. This parameter is optional.<br><br>If this parameter is not included in the call, the utility moves all token data in the input buffer to the output buffer. |
| TKN-ID-ARRAY-CNT | Input | An integer variable containing the number of entries in the token array. This parameter is required if the TKN-ID-ARRAY-PTR parameter is included in the call. |
| MAX-TKN-BUF-LGTH | Input | An integer variable containing the maximum length, in bytes, for the output buffer. This parameter is required. |
| TKN-BUF-PTR | Input | An integer pointer to the start of the output buffer. |
| TKN-BUF-LGTH | Output | An integer variable containing the length of the token data in the output buffer. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| POS-USER-DATA-FLG | Input | An integer flag indicating whether the token buffer contains BASE24-pos USER-DATA. This parameter is optional. It should not be included in the call unless a BASE24-pos standard internal message or a BASE24-pos transaction log file record is passed in the input buffer.<br><br>If this parameter is included in the call and is set to true, the utility calculates a new offset into the input buffer to skip over the USER-DATA.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the value passed in the IN-BUF-OFST parameter as the offset into the input buffer. |

## Status Codes Returned by the TKN^LOG^INFO Utility

The TKN^LOG^INFO utility returns a status code indicating the outcome of the call to the procedure. Valid status codes are as follows:

0 = Call successful. The utility completed without any errors.
11 = No token data exists in the token buffer.
21 = The token caused the token buffer to exceed the maximum allowable length.
22 = A parameter or buffer address contained invalid data.
29 = A parameter was missing on the call to the utility.

# TKN^LOG^INFO Processing

To write token data to an output buffer, the TKN^LOG^INFO utility performs the following steps:

1. Verifies that all required parameters are present in the call. If a parameter is missing, the utility sets the status code to 29 and returns to the calling procedure.
2. Skips over any non-token data contained in the token buffer (for example, over the STM).
3. Determines whether there is any token data in the buffer. To determine whether there is token data, the utility performs the following steps:
    a. If the token buffer contains the PSTM or a PTLF record, and the PSTM or PTLF record includes USER-DATA, the utility skips over the USER-DATA.
    b. If the length of the data in the token buffer is less than the number of bytes that the utility is currently offset into the token buffer, a parameter passed to the utility contained invalid data. The utility sets the status code to 22 and returns to the calling procedure.
    c. If the length of the data in the token buffer is equal to the number of bytes that the utility is currently offset into the token buffer, the token buffer does not include any token data. The utility sets the status code to 11 and returns to the calling procedure.
    d. If the length of the data in the token buffer is greater than the number of bytes that the utility is currently offset into the token buffer, the utility checks for the Header token eye catcher (&) at the current offset. If the Header token eye catcher is found, processing continues with step 4. If the Header token eye catcher is not found at the current offset, the utility sets the status code to 11 and returns to the calling procedure.
4. If a TKN record was not passed to the utility, the utility moves all tokens, including the Header token, into the output buffer. To move the token data the utility performs the following steps:
    a. Determines whether there is sufficient room in the output buffer for the token. If moving the token will cause the maximum output buffer length to be exceeded, the utility sets the total token length for the output buffer to the length specified in the output buffer Header token, sets the status code to 21, and returns to the calling procedure.

b. Adds each token to the token buffer as described below. When the utility reaches the end of the token data, processing continues with step 5. The utility determines that it has reached the end of the token data by comparing the total number of tokens skipped and the total token length to the token count and length in the Header token. If there is a discrepancy between these two indicators (for example, the token count indicates there is another token, but the total length indicates the end of the token data has been reached), the utility sets the status code to 22 and returns to the calling procedure. The utility performs the following steps to add a token to the token buffer:

   1) If this is the first token to be added to the buffer, the utility creates and adds the Header token to the token buffer. The utility sets the CNT field to 2 and the LGTH field to the sum of the Header token length, the token data to be added, and the token header length.

   2) Adds the token to the token buffer. The utility creates a token header that contains the token ID and token length and places it in the token buffer following the last token in the buffer. The utility then moves the token data to the token buffer.

   3) If this was not the first token to be added to the buffer, the utility updates the Header token by adding 1 to the count field and the token length (token header and token data) to the total token length field.

   4) Returns to step 4a.

5. If the TKN-ID-ARRAY-PTR parameter was passed to the utility, the utility moves those tokens not identified in the array to the output buffer. The array contains the token IDs of tokens that were configured not to be logged using the TKN. To move each token, the utility performs the following steps:

   **Note**: The TKN-ID-ARRAY-PTR parameter is only passed to the utility when records are being logged to the transaction log files (TLF, PTLF, TTLF, ILF, and ITLF). The TKN-ID-ARRAY-PTR parameter is not used to log tokens to processing files (BASE24-atm Terminal Data files, BASE24-pos Terminal Data files, TTDF, AQF, PRF, and TQF).

   a. Searches through the token data until the utility finds a token to be logged or reaches the end of the token data. The utility compares the token ID for each token in the input buffer to the token IDs in the array. If a token ID in the input buffer matches a token ID from the array, the token is skipped. If a token ID in the input buffer does not match any of the token IDs in the array, the utility continues with the next step (step 5b). The utility determines that it has reached the end of the token data by comparing the total number of tokens skipped and the total token length to the token count and length in the Header token. If there is a

discrepancy between these two indicators (for example, the token count indicates there is another token, but the total length indicates the end of the token data has been reached), the utility sets the status code to 22 and returns to the calling procedure. When the utility reaches the end of the token data, the utility continues with step 6.

b. Determines whether there is sufficient room in the output buffer for the token. If moving the token will cause the maximum output buffer length to be exceeded, the utility sets the total token length for the output buffer to the length specified in the output buffer Header token, sets the status code to 21, and returns to the calling procedure.

c. Adds the token to the token buffer as follows:

   1) If this is the first token to be added to the buffer, the utility creates and adds the Header token to the token buffer. The utility sets the CNT field to 2 and the LGTH field to the sum of the Header token length, the token data to be added, and the token header length.

   2) Adds the token to the token buffer. The utility creates a token header that contains the token ID and token length and places it in the token buffer following the last token in the buffer. The utility then moves the token data to the token buffer.

   3) If this was not the first token added to the buffer, the utility updates the Header token by adding 1 to the count field and the token length (token header and token data) to the total token length field.

   4) Returns to step 5a.

6. Once all tokens have been moved to the output buffer, the utility updates the output buffer total token length with the total token length from the output buffer Header token.

7. Sets the status code to 0 and returns to the calling procedure.

# Deleting Tokens

Processes can delete tokens from a token buffer by calling the TKN^DEL^INFO utility.

**Note**: Tokens should be deleted with caution. A token should only be deleted if the user can ascertain that the information in the token will not be needed for any future processing, including reversal processing. The token should also not be required for host processing (using extract) or as a historical record of how the transaction was processed (that is, it should not need to be logged). ACI recommends that to maintain transaction integrity, tokens should not be deleted from token buffers.

## TKN^DEL^INFO Parameters

The parameters in the call to the TKN^DEL^INFO utility are described, in call order, in the table below:

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-PTR | Input | An integer pointer to the buffer that contains the input data. This buffer can contain an internal message, a transaction log file record, or a token data string. |
| IN-BUF-OFST | Input | An integer variable containing the number of bytes the utility should offset into the input buffer to skip over non-token data. This value can be the length of the product-specific standard internal message (STM, PSTM, or TSTMH), the length of the transaction log file record, or the value 0.<br>**Note**: For a BASE24-pos standard internal message or BASE24-pos file record, this value should not include the USER-DATA length. The utility calculates the USER-DATA length. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-DATA-LGTH | Input | An integer variable containing the current length of data in the input buffer. |
| TKN-ID | Input | The two-byte token ID of the token being deleted, in ASCII format. This parameter is optional.<br>If this parameter is passed, the utility deletes the specific token identified in this parameter.<br>If this parameter is not passed, the utility deletes all token data that is present in the input buffer. |
| TTL-TKN-DATA-LGTH | Output | An integer variable containing the total length of the token data contained in the input buffer. The length does not include the length of the token that was deleted, if the token was deleted successfully. This parameter is optional.<br>If the calling procedure does not require the token data length after deleting a token, this parameter does not need to be included in the call. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-DSPY-FRMT-FLG | Input | An integer flag indicating whether the token data is in ASCII display format. This parameter is optional.<br><br>If this parameter is included in the call and is set to true, the utility converts the length and count fields in the Header token into integer values to use in its processing.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the length and count fields in the Header token as they are. When the token data is not in ASCII display format, these fields already contain integer values. |
| POS-USER-DATA-FLG | Input | An integer flag indicating whether the token buffer contains BASE24-pos USER-DATA. This parameter is optional. It should not be included in the call unless a BASE24-pos standard internal message or a BASE24-pos transaction log file record is passed in the input buffer.<br><br>If this parameter is included in the call and is set to true, the utility calculates a new offset into the input buffer to skip over the USER-DATA.<br><br>If this parameter is not included in the call, or is included in the call and is set to false, the utility uses the value passed in the IN-BUF-OFST parameter as the offset into the input buffer. |

## Status Codes Returned by the TKN^DEL^INFO Utility

The TKN^DEL^INFO utility returns a status code indicating the outcome of the call to the procedure. Valid status codes are as follows:

0 = Call successful. The utility completed without any errors.
11 = The specified token does not exist.
22 = A parameter or buffer address contained invalid data.
29 = A parameter was missing on the call to the utility.

## TKN^DEL^INFO Processing

To delete a token from the token buffer, the TKN^DEL^INFO utility performs the following steps:

1. Verifies that all required parameters are present in the call. If a parameter is missing, the utility sets the status code to 29 and returns to the calling procedure.
2. Skips over any non-token data contained in the token buffer (for example, over the STM).
3. Determines whether there is any token data in the buffer. To determine whether there is token data, the utility performs the following steps:
    a. If the token buffer contains the PSTM or a PTLF record, and the PSTM or PTLF record includes USER-DATA, the utility skips over the USER-DATA.
    b. If the length of the data in the token buffer is less than the number of bytes that the utility is currently offset into the token buffer, a parameter passed to the utility contained invalid data. The utility sets the status code to 22 and returns to the calling procedure.
    c. If the length of the data in the token buffer is equal to the number of bytes that the utility is currently offset into the token buffer, the token buffer does not include any token data. The utility sets the status code to 11 and returns to the calling procedure.
    d. If the length of the data in the token buffer is greater than the number of bytes that the utility is currently offset into the token buffer, the utility checks for the Header token eye catcher (&) at the current offset. If the Header token eye catcher is found, processing continues with step 4. If the Header token eye catcher is not found at the current offset, the utility sets the status code to 11 and returns to the calling procedure.

4. If a TKN-ID parameter was not passed to the utility, the utility deletes all tokens, including the Header token, from the token buffer. To delete the token data the utility performs the following steps:

   a. Sets the Header token eye catcher field to a value of 0. Because processes searching for token data look for the eye catcher, this effectively eliminates all of the token data.

   b. If the TTL-TKN-DATA-LGTH parameter is present, the utility sets the parameter to a value of 0 to indicate that there is no token data left in the input buffer.

   c. Returns to the calling procedure with a status of 0, indicating that the token data was successfully deleted. Processing continues with step 6.

   If a TKN-ID parameter was passed to the utility, processing continues with step 5.

5. If a TKN-ID parameter was passed to the utility, the utility must delete a specific token. To delete the specific token, the utility performs the following steps:

   a. Searches through the token data until the utility finds the token to be deleted or reaches the end of the token data. The utility compares the token ID for each token in the input buffer to the token ID in the TKN-ID parameter. If the token IDs do not match, the utility skips to the next token. If the token IDs do match, the utility has found the token to be deleted, and processing continues with the next step (step 5b). If the utility reaches the end of token data without finding the token, the utility sets the status code to 11 and returns to the calling procedure. The utility determines that it has reached the end of the token data by comparing the total number of tokens skipped and the total token length to the token count and length in the Header token. If there is a discrepancy between these two indicators (for example, the token count indicates there is another token, but the total length indicates the end of the token data has been reached), the utility sets the status code to 22 and returns to the calling procedure.

   b. Deletes the token from the token buffer as follows:

      1) If this is the only token in the token buffer, the utility must delete both the token and the Header token. To delete the token and the Header token the utility performs the following steps:

         a) Sets the Header token eye catcher field to a value of 0. Because processes searching for token data look for the eye catcher, this effectively eliminates all of the token data.

b) If the TTL-TKN-DATA-LGTH parameter is present, the utility sets the parameter to a value of 0 to indicate that there is no token data left in the input buffer.

c) Returns to the calling procedure with a status of 0, indicating that the token data was successfully deleted.

2) If there are other tokens in the token buffer, the utility must delete the token and update the Header token. To delete the token, the utility performs the following steps:

a) Shifts all token data to the right of the specified token left for the length of the token that is being deleted. The length to shift the token data is the length in the token header length field, plus the length of token header.

b) Updates the Header token by subtracting one from the token count field and by subtracting the length of the token that was deleted (that is, the length of the token data plus the length of token header) from the total length field.

c) If the TTL-TKN-DATA-LGTH parameter is present, the utility sets the parameter to a value in the total token length field of the Header token.

d) Sets the status code to 0 and returns to the calling procedure.

# Converting Token Data to Another Format

Processes that need to convert token data can do so by calling the TKN^MAIN^CONVERT utility. This utility allows processes to convert token data without each process having context for each token. (That is, a process can convert token data without being familiar with the contents and format of the token.) The TKN^MAIN^CONVERT utility performs the following functions:

- Converting binary fields in a token from binary to an ASCII display format
- Converting ASCII display format fields to binary
- Converting ASCII (character) fields to EBCDIC
- Creating an initialized buffer for a specific token
- Moving token data from an input buffer to an output buffer

The TKN^MAIN^CONVERT utility is actually a series of procedures separated by product. The token conversion utility source code is contained in files named *xx*TKNCVS, where *xx* identifies the product, and is one of the following two-character codes:

BA = Base (shared)
AT = BASE24-atm
PS = BASE24-pos
TR = BASE24-teller

Conversion procedures for tokens used by more than one product are in BATKNCVS, located on the BA*xx*SRC subvolume, where *xx* is the number of the current release. Base, BASE24-atm and BASE24-pos token conversion procedures are located on the BA*xx*SRC subvolume. Token conversion procedures are on the TR*xx*SRC subvolume.

**Note:** The code TB is reserved for the BASE24-telebanking product. However, this product does not use tokens in its processing and does not have a TBTKNCVS file.

# TKN^MAIN^CONVERT Parameters

The parameters in the call to the TKN^MAIN^CONVERT utility are described, in call order, in the table below:

| Parameter Name | Input or Output | Description |
|---|---|---|
| TKN-ID | Input | The two-byte token ID for the token being converted, in ASCII format. |
| PROD-ID | Input | An integer flag indicating the product with which the token is being used. Valid values are as follows:<br>0 = BASE24 Base<br>1 = BASE24-atm<br>2 = BASE24-pos<br>3 = BASE24-teller<br>14 = BASE24-telebanking |
| CONV-FLG | Input | An integer flag indicating the type of conversion to be done. Valid values are as follows:<br>0 = Convert the binary data from a binary format to an ASCII format.<br>1 = Convert the binary data from an ASCII format to a binary format.<br>2 = Convert the ASCII data from an ASCII character set to an EBCDIC character set.<br>3 = Move token data from the input buffer to the output buffer. This option requires that both an input buffer and an output buffer be specified in the call to the procedure. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| IN-BUF-PTR | Input | An integer pointer to the input buffer containing the token data for the token to be converted. This parameter is optional.<br><br>If this parameter is not included in the call, the utility initializes (that is, sets ASCII fields to spaces and binary fields to zero) an output buffer for the maximum size of the token. |
| OUT-BUF-PTR | Input | An integer pointer to the output buffer where the converted token data should be placed.<br><br>**Note**: This parameter is not required when converting ASCII data to EBCDIC character set format. Since this conversion does not change the size of the token data, the conversion is done directly on the data contained in the buffer pointed to by the IN-BUF-PTR parameter. |
| OUT-BUF-LGTH | Output | The integer length of the converted token data in the output buffer. This parameter should be included in the call if the token being converted is variable in length. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| FIXED-FRMT-FLG | Input | An integer flag indicating whether the converted token should be in fixed-length format. This parameter is optional.<br><br>If this parameter is present and contains a value of true, the conversion utilities expand the token data to the maximum size that the token data can be, regardless of the size of the input token data.<br><br>If this parameter is not present, or is present and contains a value of false, the conversion utilities convert the token, but do not expand the token data to its maximum size. |
| TKN-LGTH | Input | An integer variable containing the current length of data in the input buffer. This parameter is required when the token being converted is a variable-length token, otherwise, it is optional. |

| Parameter Name | Input or Output | Description |
|---|---|---|
| EXTR-FLG | Input | An integer flag indicating whether data converted from binary to ASCII display should be provided in a zoned data format (in which the trailing digit of the converted ASCII data signifies both a digit value and a sign value).<br><br>If this parameter is present and contains a value of true, the conversion utilities convert binary data to a zoned data format.<br><br>If this parameter is not present, or is present and contains a value of false, the conversion utilities convert the binary data to a standard ASCII display format—a minus sign (-) is placed in the left-most position for negative values, and no sign is provided for positive values. |

## Status Codes Returned by the TKN^MAIN^CONVERT Utility

The TKN^MAIN^CONVERT utility returns a status code indicating the outcome of the call to the procedure. Valid status codes are as follows:

0 = Call successful. The utility completed without any errors.
1 = An error occurred in converting the data contained in the token.
2 = The token ID specified in the call is not recognized by the conversion utility.
3 = A parameter was missing or invalid on the call to the utility.

# TKN^MAIN^CONVERT Processing

To convert a token, the TKN^MAIN^CONVERT utility performs the following steps:

1. Verifies that all required parameters are present in the call. If a parameter is missing, the utility sets the status code to 3 and returns to the calling procedure.
2. Checks the token ID to determine whether it has context for the token. (That is, the utility verifies that it recognizes the token and its format.) If the utility does not have context for the token in the Base conversion procedures, the utility calls the appropriate product-specific primary module based on the PROD-ID parameter passed to the procedure.

   **Note**: The primary procedure passes the CONV-FLG parameter to the actual conversion procedure to instruct the conversion procedure on how the data should be converted. The actual conversion procedure does the processing required to convert the token data.
3. Calls the token-specific conversion procedure if the Base conversion procedures or the product-specific conversion procedures have context for the token. The remaining steps (steps 4 through 7) describe the general processing for each token-specific conversion procedure. If neither the Base conversion procedures nor the product-specific conversion procedures have context for the token, the utility sets the status code to 2 and returns to the calling procedure.
4. Initializes the token data, if the IN-BUF-PTR parameter was not included in the call or if the FIXED-FRMT-FLG parameter contained the value true. To initialize the token data, the utility performs the following steps:
   a. Creates an output buffer for the maximum size of the token.
   b. Fills all ASCII fields within the token definition with blanks and all binary fields within the definition with zeros.
   c. If the conversion is not ASCII to EBCDIC and if the IN-BUF-PTR parameter was not passed in the call, the initialized token data is in the correct form. The utility sets the status code to 0 and returns to the calling application. If the conversion is ASCII to EBCDIC, or if the IN-BUF-PTR parameter was passed in the call, processing continues with the next step (step 5).

5. Converts the token based on the value in the CONV-FLG parameter. To convert the token, the utility performs one of the following procedures:

   a. If the procedure is to convert the binary data from a binary format to an ASCII display format, the utility performs the following steps:

      1) Processes each field sequentially, from the first field to the last field in the given token. The procedure must have context for each field in the token.

      2) Moves each ASCII field in the input buffer to the output buffer.

      3) Converts each binary field to an ASCII display format. To do this, the utility must call one of the following utilities based on the data type of the binary field:

         - INTEGER^ASCII if the field is an integer
         - DOUBLE^ASCII if the field is an int(32)
         - FIXED^ASCII if the field is fixed

      4) Places the converted binary field in the output buffer.

      5) If the binary field being converted contains a negative value, the utility performs one of the following steps:

         - If the EXTR-FLG parameter is present and set to true, the converted value needs to be in a zoned data format. In this case, the minus sign (-) is embedded in the last character of the ASCII representation of the binary value.
         - If the EXTR-FLG parameter is not present or is set to false, the minus sign (-) is placed in the ASCII data in the left-most position.

         Processing continues with step 6.

   b. If the procedure is to convert the binary data from an ASCII display format to a binary format, the utility performs the following steps:

      1) Processes each field sequentially, from the first field to the last field in the given token. The procedure must have context for each field in the token.

      2) Moves each ASCII field in the input buffer to the output buffer.

      3) Converts each ASCII display field to a binary format. To do this, the utility must call one of the following utilities based on the data type of the binary field:

         - ASCII^INTEGER if the field is an integer

- ASCII^DOUBLE if the field is an int(32)
- ASCII^FIXED if the field is fixed

4) The binary field is placed in the output buffer. Processing continues with step 6.

c. If the procedure is to convert the ASCII data from an ASCII character set format to an EBCDIC character set format, the utility performs the following steps:

1) Processes each field sequentially, from the first field to the last field in the token. The procedure must have context for each field in the token.

2) Converts each ASCII field in the input buffer to an EBCDIC character set format by calling the ASCII^TO^EBCDIC utility. Note that if ASCII fields occur sequentially within the token (that is, one ASCII field is followed by another ASCII field), the contiguous ASCII fields can be grouped together and converted at the same time with a single call to the ASCII^TO^EBCDIC utility.

Processing continues with step 6.

d. If the procedure is to move data from the input buffer to the output buffer, the utility moves the input buffer to the output buffer. If the TKN-LGTH parameter was passed in the call, the data should be moved for the length specified in the TKN-LGTH parameter.

6. If the conversion procedure is unable to convert the token data, it performs the following steps:

a. Sets the OUT-BUF-LGTH parameter to a value of zero.

b. Sets the status code to 1.

c. Returns to the TKN^MAIN^CONVERT utility.

d. The TKN^MAIN^CONVERT utility returns a value of false and the OUT-BUF-LGTH value to the calling procedure.

7. If the conversion is successful, the utility performs the following steps:

a. Sets the OUT-BUF-LGTH parameter to a length of the converted token data.

b. Sets the status code to 0.

c. Returns to the TKN^MAIN^CONVERT utility.

d. The TKN^MAIN^CONVERT utility returns a value of true and the OUT-BUF-LGTH value to the calling procedure.

# 3: User Tokens

BASE24 users can create their own tokens. Once a token is defined in the BASE24 system, it can be carried in the internal or external message, logged to a transaction log file, and extracted from the transaction log file. This section describes the procedures to use when adding token definitions to the BASE24 system.

**Note**: While any token defined with these procedures can be carried in BASE24 messages and logged to or extracted from the BASE24 transaction log files, the information in the token **is not** used for processing by BASE24 products. In addition, BASE24 products cannot create the token, although it can be received using the external message. In order for BASE24 products to process information in a user-defined token, custom software modifications (CSMs) are required.

**Note:** The Token (TKN) requester performs data checking of token IDs before displaying the TKN screens. The data checking includes checking for invalid IDs and for duplicate IDs. For this reason, defining large numbers of tokens can cause performance impacts to the Pathway system.

# Creating User Tokens

To add a user-defined token to the BASE24 system, perform the following steps:

1. Determine a token ID for the token. The token ID is a two-character code that uniquely identifies the token.

   **Note**: Token IDs P0 through RZ are provided for user-defined tokens. Token IDs in the ranges 00 through OZ and S0 through WZ are reserved for use by BASE24 products. Token IDs in the range X0 through ZZ are reserved for use by distributors.

2. Add the token data structure to the appropriate DDL source schema. There are four DDL source files which contain token data structures: one Base file, and one product-specific file each for BASE24-atm, BASE24-pos, and BASE24-teller. These files are named DDL*yy*TKN, where *yy* identifies the product, and is one of the following two-character codes:

   BA = Base (shared)
   AT = BASE24-atm
   PS = BASE24-pos
   TR = BASE24-teller

   The Base, BASE24-atm, and BASE24-pos DDL*xx*TKN files are located on the BA*xx*DDL subvolume, where *xx* is the number of the current release. The BASE24-teller DDLTRTKN file is located on the TR*xx*DDL subvolume. Tokens used by more than one product should be added to the Base file. Tokens used by a single product should be added to the product-specific file.

   If the token contains binary data fields, two token data structures should be added to the file: a binary definition and a display definition. The binary definition should be added first, followed immediately by the display definition. The standard naming convention for binary definitions is *name*-TKN, where *name* identifies the token. Display tokens have the same name as their binary counterpart, except the token name ends in TKNX. For example, the binary and display token definition names for a token containing XYZ data would be XYZ-TKN and XYZ-TKNX.

3. If desired, add global defines related to the token to the token globals. The token globals are defined in the file BAUTILS on the BA*xx*SRC subvolume, where *xx* is the number of the current release. The data block is named TKN^GLOBALS. These globals can then be used for processing the token data.

4. Write a token-specific conversion procedure and add it to the appropriate token conversion utilities. The token conversion utility source code is contained in files named *xx*TKNCVS, where *xx* identifies the product, and is one of the following two-character codes:

   BA = Base (shared)
   AT = BASE24-atm
   PS = BASE24-pos
   TR = BASE24-teller

   Conversion procedures for tokens used by more than one product should be added to BATKNCVS, located on the BA*xx*SRC subvolume, where *xx* is the number of the current release. Base, BASE24-atm and BASE24-pos token conversion procedures are located on the BA*xx*SRC subvolume. Token conversion procedures are on the TR*xx*SRC subvolume.

5. Identify and enhance each process that will actually use the data in the new token for processing. In addition to the actual data processing, the enhancement should include the following:

   - Statements to source in any required token utilities (TKN^GET^INFO, TKN^GET^IDS, TKN^ADD^INFO, TKN^DEL^INFO, TKN^MAIN^CONVERT, TKN^UPDT^INFO, TKN^SORT^INFO, TKN^LOG^INFO, and/or TKN^GET^INFOX, TKN^GET^IDSX, TKN^ADD^INFOX, TKN^DEL^INFOX, TKN^UPDT^INFOX, TKN^SORT^INFOX, TKN^LOG^INFOX), if they are not already being sourced in to the application.
   - Statements to source in the token globals from BATKNID, if they are not already being sourced in to the application. The token globals are defined in BA*xx*SRC.BATIKNID and are sourced in BA*xx*SRC.BAUTILS in a data block named TKN^GLOBALS.
   - Statements to source in the literals MAX^MSG^LGTH, INTRN^MSG^LMT and MAX^BLK^LGTH, if they are not already being sourced in to the application. These literals are defined in SPANNET.SKEL.
   - Statements to source in the binary format of the token.
   - Statements to source in the display format of the token, if the application calls TKN^MAIN^CONVERT to convert the token to or from display format.

6. Recompile the DDL*xx*TKN file to which the token was added.

7. Recompile the token conversion utilities to which the conversion procedure was added. A Make file exists to recompile each set of token conversion utilities. The naming conventions for the file is *xx*TKNCVM, where *xx* has

the same value as it does in the source code file (for example, the obey file to compile ATTKNCVS is ATTKNCVM). The Make file is located on the same subvolume as the source code.

8. Recompile and rebind all modules that use the conversion utilities. This should include the following modules:
   - ISO Host Interface process
   - BIC ISO Interface process
   - Super Extract process
9. Add the token to the PROD-TKN-TABLE in COBTKN. The PROD-TKN-TABLE contains the token ID, the product ID of the product that uses the token, and a descriptive token name. The information specified in COBTKN is used to display token information on the TKN screens. A sample PROD-TKN-TABLE is shown below:

```
1 PROD-TKN-TABLE.
05 POS1  PIC X(29) VALUE "0102POS ADDRESS VERIFICATION ".
05 ATM1  PIC X(29) VALUE "0201ATM STATEMENT PRINT INFO ".
05 ATM2  PIC X(29) VALUE "0301ATM RELEASE 5.0 TOKEN    ".
05 POS2  PIC X(29) VALUE "0402POS RELEASE 5.0 TOKEN    ".
05 POS3  PIC X(29) VALUE "0502POS CHECK AUTH           ".
05 ATM3  PIC X(29) VALUE "0601ATM PIN CHANGE           ".
05 ATM4  PIC X(29) VALUE "0701ATM SSB BASE APPLICATION ".
05 TLR15 PIC X(29) VALUE "0803TLR CUSTOMER SHORT NAME  ".
05 POS4  PIC X(29) VALUE "1002POS AMEX DATA            ".
05 POS5  PIC X(29) VALUE "1102POS ACH DEBIT TOKEN      ".
05 ATM5  PIC X(29) VALUE "1201ATM MICR DATA TOKEN      ".
05 TLR16 PIC X(29) VALUE "1303TLR CREDIT LINE TOKEN    ".
05 ATM6  PIC X(29) VALUE "1401ATM SSB CHECK APPLICATION".
05 ATM7  PIC X(29) VALUE "1501ATM SSB TERMINAL SETTL.  ".
05 POS6  PIC X(29) VALUE "1602POS ALTERNATE MERCHANT ID".
05 TLR0  PIC X(29) VALUE "T003TLR FINANCIAL TOKEN      ".
05 TLR1  PIC X(29) VALUE "T103TLR CAF INQUIRY TOKEN    ".
05 TLR2  PIC X(29) VALUE "T203TLR CAF UPDATE TOKEN     ".
05 TLR3  PIC X(29) VALUE "T303TLR NBF TOKEN            ".
05 TLR4  PIC X(29) VALUE "T403TLR PBF INQUIRY TOKEN    ".
05 TLR5  PIC X(29) VALUE "T503TLR PBF UPDATE TOKEN     ".
05 TLR6  PIC X(29) VALUE "T603TLR SPF INQUIRY TOKEN    ".
05 TLR7  PIC X(29) VALUE "T703TLR SPF UPDATE TOKEN     ".
05 TLR8  PIC X(29) VALUE "T803TLR WHFF INQUIRY TOKEN   ".
05 TLR9  PIC X(29) VALUE "T903TLR WHFF UPDATE TOKEN    ".
05 TLR10 PIC X(29) VALUE "TA03TLR ADMINISTRATIVE TOKEN ".
05 TLR11 PIC X(29) VALUE "TB03TLR ACCOUNT TOKEN        ".
05 TLR12 PIC X(29) VALUE "TC03TLR OVERRIDE TOKEN       ".
05 TLR13 PIC X(29) VALUE "TD03TLR PIN TOKEN            ".
05 TLR14 PIC X(29) VALUE "TE03TLR NATIVE MESSAGE TOKEN ".
05 POS7  PIC X(29) VALUE "1702POS PS2000 TOKEN         ".
05 ATM8  PIC X(29) VALUE "1801ATM ACCT QUALIFIER       ".
```

```
05 POS8  PIC X(29) VALUE "1902POS PS2000 OFFLINE       ".
05 POS9  PIC X(29) VALUE "2002POS INTERCHNG COMPLIANCE ".
05 ATM9  PIC X(29) VALUE "2101ATM PS2000/ATM           ".
05 ATM10 PIC X(29) VALUE "2201ATM ADDITIONAL HOPPERS   ".
05 ATM11 PIC X(29) VALUE "2301ATM TRACK1               ".
05 POS10 PIC X(29) VALUE "2302POS TRACK1               ".
05 TLR17 PIC X(29) VALUE "2303TLR TRACK1               ".
05 ATM12 PIC X(29) VALUE "2401ATM FLAG1                ".
05 ATM13 PIC X(29) VALUE "2501ATM SURCHARGE TOKEN      ".
05 POS11 PIC X(29) VALUE "C002POS RELEASE 5.1 TOKEN    ".
05 ATM14 PIC X(29) VALUE "B001ATM ACQUIRER SWITCH TOKEN".
05 ATM15 PIC X(29) VALUE "B101ATM ISSUER   SWITCH TOKEN".
05 POS12 PIC X(29) VALUE "B002POS ACQUIRER SWITCH TOKEN".
05 POS13 PIC X(29) VALUE "B102POS ISSUER   SWITCH TOKEN".
05 ATM16 PIC X(29) VALUE "2701ATM CARDHOLDER POSTAL CDE".
05 POS14 PIC X(29) VALUE "2702POS CARDHOLDER POSTAL CDE".
05 POS15 PIC X(29) VALUE "C102POS STATION ID TOKEN     ".
05 POS16 PIC X(29) VALUE "C202POS PURCHASING CARD TOKEN".
05 POS17 PIC X(29) VALUE "C302POS CERTIFICATE TOKEN    ".
05 ATM17 PIC X(29) VALUE "A001ATM SMART CARD PRIMARY   ".
05 ATM18 PIC X(29) VALUE "A201ATM SMART CARD REFRESH   ".
05 ATM19 PIC X(29) VALUE "A301ATM SMART CARD VISA      ".
05 ATM20 PIC X(29) VALUE "A401ATM SMART CARD TERM SETL ".
05 ATM21 PIC X(29) VALUE "A501ATM NON-CURRENCY DISPENSE".
05 ATM22 PIC X(29) VALUE "A601ATM INTERCHNG COMPLIANCE ".
05 POS18 PIC X(29) VALUE "C402POS POINT OF SERVICE DATA".
05 POS19 PIC X(29) VALUE "C502POS OPTIONAL DATA TOKEN  ".
05 POS20 PIC x(29) VALUE "C602POS TRANS STAIN XID TOKEN".
05 POS21 PIC x(29) VALUE "C702POS CARDHOLDER SERIAL NUM".
05 POS22 PIC x(29) VALUE "C802POS MERCHANT SERIAL NUM  ".
05 POS23 PIC x(29) VALUE "C902POS MHI ADDITIONAL DATA  ".
05 ATM23 PIC X(29) VALUE "B701ATM TLF DATA TOKEN       ".
05 ATM24 PIC X(29) VALUE "B801ATM TXN PROFILE TOKEN    ".
05 ATM25 PIC X(29) VALUE "B901ATM TXN DESCRIPTION TOKEN".
05 ATM26 PIC X(29) VALUE "BA01ATM ACQ ROUTING TOKEN    ".
05 POS24 PIC X(29) VALUE "B702POS TLF DATA TOKEN       ".
05 POS25 PIC X(29) VALUE "B802POS TXN PROFILE TOKEN    ".
05 POS26 PIC X(29) VALUE "B902POS TXN DESCRIPTION TOKEN".
05 POS27 PIC X(29) VALUE "BA02POS ACQ ROUTING TOKEN    ".

05 ATM27 PIC X(29) VALUE "B201ATM EMV REQUEST TOKEN    ".
05 ATM28 PIC X(29) VALUE "B301ATM EMV DISCR TOKEN      ".
05 ATM29 PIC X(29) VALUE "B401ATM EMV STATUS TOKEN     ".
05 ATM30 PIC X(29) VALUE "B501ATM EMV RESPONSE TOKEN   ".
05 ATM31 PIC X(29) VALUE "B601ATM EMV SCRIPT TOKEN     ".
05 ATM32 PIC X(29) VALUE "BD01ATM MULTI CURRENCY TOKEN ".
05 ATM33 PIC X(29) VALUE "BE01ATM ORIG CRNCY 60 TOKEN  ".



05 ATM34 PIC X(29) VALUE "AB01ATM BALANCE TOKEN        ".
```

```
05 POS28 PIC X(29) VALUE "B202POS EMV REQUEST TOKEN    ".
05 POS29 PIC X(29) VALUE "B302POS EMV DISCR TOKEN      ".
05 POS30 PIC X(29) VALUE "B402POS EMV STATUS TOKEN     ".
05 POS31 PIC X(29) VALUE "B502POS EMV RESPONSE TOKEN   ".
05 POS32 PIC X(29) VALUE "B602POS EMV SCRIPT TOKEN     ".
05 POS33 PIC X(29) VALUE "BD02POS MULTI CURRENCY TOKEN ".
05 POS34 PIC X(29) VALUE "BE02POS ORIG CRNCY 60 TOKEN  ".



05 POS35 PIC X(29) VALUE "CB02POS BALANCE TOKEN        ".



05 ATM35 PIC X(29) VALUE "2801ATM PRISM TOKEN          ".
05 ATM36 PIC X(29) VALUE "3001ATM ISS FEE/REBATE TOKEN ".
05 ATM37 PIC X(29) VALUE "BG01ATM TRACK3 TOKEN         ".

05 POS36 PIC X(29) VALUE "2802POS PRISM TOKEN          ".
05 POS37 PIC X(29) VALUE "BG02POS TRACK3 TOKEN         ".
05 POS38 PIC X(29) VALUE "2902POS CHECK AUTH 2 TOKEN   ".
05 POS39 PIC X(29) VALUE "3102POS CHECK CALLBACK TOKEN ".
05 POS40 PIC X(29) VALUE "U102POS EBT AVAILABLE BALANCE".
05 POS41 PIC X(29) VALUE "U002POS EBT VOUCHER NUMBER   ".


05 POS42 PIC X(29) VALUE "BH02POS REVERSAL DATE/TIME   ".
05 ATM38 PIC X(29) VALUE "BH01ATM REVERSAL DATE/TIME   ".

05 POS43 PIC X(29) VALUE "CA02POS DUKPT DATA           ".

05 POS44 PIC X(29) VALUE "CE02POS AUTHENTICATION DATA  ".
05 POS45 PIC X(29) VALUE "CF02POS IAVS DATA            ".
05 POS46 PIC X(29) VALUE "U202POS STORED VALUE         ".
05 ATM39 PIC X(29) VALUE "BK01ATM MULTIPLE LN          ".
05 POS47 PIC X(29) VALUE "BK02POS MULTIPLE LN          ".

05 ATM40 PIC X(29) VALUE "BJ01ATM EMV ISS SCRIPT RSLTS ".
05 POS48 PIC X(29) VALUE "BJ02POS EMV ISS SCRIPT RSLTS ".
05 ATM41 PIC X(29) VALUE "BL01ATM PSEUDO CARD NUMBER   ".
05 POS49 PIC X(29) VALUE "BL02POS PSEUDO CARD NUMBER   ".
05 ATM42 PIC X(29) VALUE "BM01ATM TRANSACTION SUBTYPE  ".
05 POS50 PIC X(29) VALUE "BM02POS TRANSACTION SUBTYPE  ".
05 ATM43 PIC X(29) VALUE "BN01ATM DATA ENCRYPTION KEY  ".
05 POS51 PIC X(29) VALUE "BN02POS DATA ENCRYPTION KEY  ".
05 POS52 PIC X(29) VALUE "CH02POS POS DATA1 TOKEN      ".
05 POS53 PIC X(29) VALUE "CI02POS POS MERCHANT TOKEN   ".
05 ATM44 PIC X(29) VALUE "AD01ATM CASH ACCEPT TERM SETL".
05 ATM45 PIC X(29) VALUE "A801ATM MBC BAG DEPOSIT      ".
05 ATM46 PIC X(29) VALUE "A901ATM MBC MONEY EXCHANGE   ".
05 ATM47 PIC X(29) VALUE "AA01ATM MBC SETTLEMENT       ".
```

```
   05 ATM48 PIC X(29) VALUE "BF01ATM PRE PAY RECEIPT TOKEN".
   05 POS54 PIC X(29) VALUE "BF02POS PRE PAY RECEIPT TOKEN".
   05 ATM49 PIC X(29) VALUE "BI01ATM PRE PAY TOP UP TOKEN ".
   05 POS55 PIC X(29) VALUE "BI02POS PRE PAY TOP UP TOKEN ".


01 PROD-TKN-TABLE-R          REDEFINES PROD-TKN-TABLE.

   02 WS-PROD-TKN-TBL              OCCURS 122 TIMES.




      05 PROD-TBL-TOKEN-ID         PIC X(2).
      05 PROD-TBL-PROD-ID          PIC X(2).
      05 PROD-TBL-TOKEN-DESC       PIC X(25).




01 PROD-TKN-TABLE-CNT              PIC  9(4) COMP VALUE 122.
```

To add a new token to this table, perform the following steps:

a. Copy one of the 05 lines in the table.

b. In the new line, change the field name descriptor (for example, POS1, ATM4, TLR14) to CSM*num*, where *num* is a count of the number of user-defined tokens. For example, the first user-defined token field name descriptor would be CSM1 and the tenth user-defined token field name descriptor would be CSM10.

   **Note:** Older modifications might use a naming convention of RPQ*num* for the field name descriptor.

c. In the VALUE field for the new line, enter the two-character token ID, the two-character product ID, and the 25-character token name. Valid product IDs are as follows:

   01 = BASE24-atm
   02 = BASE24-pos
   03 = BASE24-teller

   There should be no spaces between the token ID, the product ID, and the token name (although there can be spaces *within* the token name). The token name cannot exceed 25 characters in length. For example, to add a token for BASE24-atm named XYZ Data with the token ID P3, the new entry in the TKN-TABLE would appear as follows:

```
05 CSM4  PIC X(29) VALUE "P301XYZ DATA                 ".
```

d. Add 1 to the value in the OCCURS clause for the WS-PROD-TKN-TBL field, and to the VALUE field associated with the PROD-TKN-TABLE-CNT field. These values should be equal, and should match the number of entries in the PROD-TKN-TABLE.

**Note**: If the token is used by more than one product, add a line to this table for each product that uses the token. For example, if the XYZ Data token above is also used by BASE24-teller, you would add two new entries, appearing as follows:

```
05 CSM4  PIC X(29) VALUE "P301XYZ DATA                ".
05 CSM5  PIC X(29) VALUE "P303XYZ DATA                ".
```

10. Remake the Token requester.
11. Update all TKN records to indicate whether the new token is to be extracted, sent in the external message, or logged to a transaction log file.

# 4: Configuring Internal Message Token Processing

The Token File (TKN) is used to configure token processing. It specifies the following information:

- The tokens that get logged to the transaction log files. The transaction log files affected are the BASE24-atm Transaction Log File (TLF), POS Transaction Log File (PTLF), Teller Transaction Log File (TTLF), ITS Transaction Log File (ITLF), and Interchange Log Files (ILFs).
- The tokens that get extracted from the transaction log files by the Super Extract process. Extracting allows the token information to be processed by the host.
- The tokens that get sent to the host or co-network with each transaction in the ISO-based external message. Different tokens can be sent depending on the message type.

This section describes how to configure token logging, extracting, and inclusion in the external message using the TKN.

# Token File Screens

Four screens allow access to Token File (TKN) information. The first screen contains a field that allows the user to specify which of the remaining three screens is displayed. Each of the remaining screens contains information about a specific type of TKN record:

- TKN screen 2 contains fields specifying which tokens are logged to the transaction log files.
- TKN screen 3 contains fields specifying which tokens are extracted from the transaction log files.
- TKN screen 4 contains fields specifying which tokens are sent in the external message.

The remainder of this section contains instructions for adding records to the TKN. TKN screens 1 through 4, along with field descriptions, are also shown. These instructions assume you are logged on to the BASE24 CRT access system. For more information about the TKN screens, refer to the ***BASE24 Base Files Maintenance Manual.***

# Specifying Tokens to be Logged

The tokens that are logged to the transaction log files determine what token information is available for perusal online and what token information is available for extract. For the BASE24-teller product, the tokens that are logged also determine what information is displayed on the daily detail reports.

When a process has a transaction to be logged to a transaction log file, the process checks its internal table of TKN records to determine whether a token record for the specific type and subtype exists. Depending on what the process finds in its internal table, the following tokens are logged:

- If no record is found, all tokens in the internal message are logged to the log file.
- If a TKN record is found, but some tokens in the internal message are not identified in the TKN record, those tokens identified to be logged in the TKN record plus any tokens in the message that are not specified in the TKN record are logged. For example, assume the internal message includes tokens 01, 02, 04, and 06; and the TKN record indicates tokens 01, 03, 04, and 08 should be logged and tokens 02 and 05 should not be logged. In this case, the process logs tokens 01, 04, and 06. Tokens 01 and 04 are logged because the TKN record indicated that they should be logged. Token 06 is logged by default, because it is not identified **at all** in the TKN record.

  **Note:** The above situation should not occur often. This situation could occur after a new token is added to the system, if the TKN records are not updated when the new token is added. The TKN records should be updated whenever a token is added to the system.
- If a TKN record is found, and all tokens in the internal message are specified (either to be logged or not to be logged) in the TKN record, the process logs those tokens which are specified to be logged in the TKN record.

**Note:** Additional disk space may be required to store transaction log files (the BASE24-atm Transaction Log File (TLF), POS Transaction Log File (PTLF), Teller Transaction Log File (TTLF), ITS Transaction Log File (ITLF), and Interchange Log Files (ILFs)), depending on the current record length and the amount of token data you choose to log to the file. For more information on determining the impact of logging token data to a transaction log file, refer to appendix A.

# Key Field Settings

The key to records in the TKN is a combination of all of the fields on TKN screen 1: TOKEN GROUP, PRODUCT ID, TYPE, SUBTYPE, and FUNCTION TYPE. An understanding of how the processes which read the TKN use these fields is essential before creating TKN records. The following topics identify each process that uses TKN records to log transactions, together with the values that the process uses to read TKN records.

## BASE24-atm Authorization Process

During initialization, the BASE24-atm Authorization process reads the TKN for the records that the Authorization process should use to log transactions to the TLF. The Authorization process uses a partial key to retrieve all records where the TOKEN GROUP field contains the value ****, the PRODUCT ID field contains the value 01 (BASE24-atm), and the FUNCTION TYPE field contains the value 0 (logging record).

When the Authorization process writes a record to the TLF, the Authorization process checks the TKN records that it read into memory to determine what tokens should be logged with the record. To select a TKN record from memory, the Authorization process uses the HEAD.REC-TYP and *subtype*.ADMIN-CDE fields from the TLF record, where *subtype* is TERM-SETL, TERM-CASH, or SETL-TTL administrative record types. The values for these fields and the corresponding TYPE and SUBTYPE field values in the TKN are shown in the table below.

| **HEAD.REC-TYP** | ***subtype*.ADMIN-CDE** | **TYPE** | **SUBTYPE** |
|---|---|---|---|
| 01 (Financial) | Not applicable | 01 | ** |
| 20 (Exception) | Not applicable | 20 | ** |
| 21 (Exception) | Not applicable | 21 | ** |
| 04 (Administrative) | 01 (Initialize cash by Device Handler process) | 04 | 01 |
| 04 (Administrative) | 02 (Initialize cash by DCT Server process) | 04 | 02 |
| 04 (Administrative) | 03 (Increment cash by Device Handler process) | 04 | 03 |

| HEAD.REC-TYP | *subtype*.ADMIN-CDE | TYPE | SUBTYPE |
|---|---|---|---|
| 04 (Administrative) | 04 (Increment cash by DCT Server process) | 04 | 04 |
| 04 (Administrative) | 05 (Terminal Cutover by Device Handler process) | 04 | 05 |
| 04 (Administrative) | 06 (Terminal Cutover by DCT Server process) | 04 | 06 |
| 04 (Administrative) | 07 (Decrement cash by Device Handler process) | 04 | 07 |
| 04 (Administrative) | 08 (Decrement cash by DCT Server process) | 04 | 08 |
| 04 (Administrative) | 09 (Forced Cutover) | 04 | 09 |
| 04 (Administrative) | 20 (Terminal Totals by DCT Server process) | 04 | 20 |
| 04 (Administrative) | 21 (Terminal Totals by Device Handler process) | 04 | 21 |
| 04 (Administrative) | 22 (Terminal Totals by Settlement Initiator process) | 04 | 22 |

If a TKN record is not found and the HEAD.REC-TYP field contains the value 01, 20, or 21, the Authorization process checks for a TKN record with ** in the TYPE field. If the Authorization process still cannot find a TKN record, the Authorization process logs all token data.

If a TKN record is not found and the HEAD.REC-TYP field contains the value 04, the Authorization process checks for a TKN record with the TYPE specified in the table above and the value ** in the SUBTYPE field. If the Authorization process still cannot find a TKN record, the Authorization process checks for a TKN record with the value ** in both the TYPE and SUBTYPE fields. If a record is still not found, the Authorization process logs all token data.

In summary, the BASE24-atm Authorization process performs the following searches when looking for a TKN record to use for logging a record to the TLF:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | **** | 01 | See table above | See table above | 0 |
| Second choice | **** | 01 | See table above | ** | 0 |
| Third choice | **** | 01 | ** | ** | 0 |

If the Authorization process has not found a record after performing these searches, the Authorization process logs all tokens in the internal message to the TLF.

## BASE24-atm Settlement Initiator Process

During initialization, the BASE24-atm Settlement Initiator process reads the TKN for the records that the Settlement Initiator process should use to log transactions to the TLF. The Settlement Initiator process uses a partial key to retrieve records where the TOKEN GROUP field contains the value ****, the PRODUCT ID field contains the value 01 (BASE24-atm), and the FUNCTION TYPE field contains the value 0 (logging record).

When the Settlement Initiator process writes a record to the TLF, the Settlement Initiator process checks the TKN records that it read into memory to determine what tokens should be logged with the record. To select a TKN record from memory, the Settlement Initiator process uses the HEAD.REC-TYP and *subtype*.ADMIN-CDE fields from the TLF record, where *subtype* is TERM-SETL or SETL-TTL administrative record types. The values for these fields and the corresponding TYPE and SUBTYPE field values in the TKN are shown in the table below.

| HEAD.REC-TYP | *subtype*.ADMIN-CDE | TYPE | SUBTYPE |
|---|---|---|---|
| 04 (Administrative) | 09 (Forced Cutover) | 04 | 09 |
| 04 (Administrative) | 22 (Terminal Totals by Settlement Initiator) | 04 | 22 |

If a TKN record is not found, the Settlement Initiator process checks for a TKN record with the TYPE specified in the table above and the value ** in the SUBTYPE field. If the Settlement Initiator process still cannot find a TKN record, the Settlement Initiator process checks for a TKN record with the value ** in both the TYPE and SUBTYPE fields. If a record is still not found, the Settlement Initiator process logs all token data.

In summary, the BASE24-atm Settlement Initiator process performs the following searches when looking for a TKN record to use for logging a record to the TLF:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | **** | 01 | See table above | See table above | 0 |
| Second choice | **** | 01 | See table above | ** | 0 |
| Third choice | **** | 01 | ** | ** | 0 |

If the Settlement Initiator process has not found a record after performing these searches, the Settlement Initiator process logs all tokens that it can process to the TLF.

## Interchange Interface Processes

During initialization, the Interchange Interface process reads the TKN for the records that the Interchange Interface process should use to log transactions to the ILF. The Interchange Interface process reads at most two TKN records: one for the BASE24-atm product, and one for the BASE24-pos product.

**If the Interchange Interface process supports the BASE24-atm product.** The Interchange Interface process searches the TKN for a record using the following values in the key fields:

- A TOKEN GROUP field containing the interchange FIID (specified in the INTERCHANGE FIID field on Interchange Configuration File (ICF) or Enhanced Interchange Configuration File (ICFE) screen 1), a PRODUCT ID field containing the value 01 (BASE24-atm), TYPE and SUBTYPE fields containing the value **, and a FUNCTION TYPE field containing the value 0.

- A TOKEN GROUP field containing the value ****, a PRODUCT ID field containing the value 01, TYPE and SUBTYPE fields containing the value **, and a FUNCTION TYPE field containing the value 0.

If the Interchange Interface process finds a record using one of these keys, that record is stored in memory, and is used to log all BASE24-atm records to the ILF. Note that only one record is retrieved; that is, if a record is found using the first set of key field values, the second search is not performed. If the Interchange Interface process does not find a record using one of these keys, the Interchange Interface process logs all tokens in BASE24-atm messages to the ILF by default.

**If the Interchange Interface process supports the BASE24-pos product.** The Interchange Interface process searches the TKN for a record using the following values in the key fields:

- A TOKEN GROUP field containing the interchange FIID (specified in the INTERCHANGE FIID field on ICF or ICFE screen 1), a PRODUCT ID field containing the value 02 (BASE24-pos), TYPE and SUBTYPE fields containing the value **, and a FUNCTION TYPE field containing the value 0.
- A TOKEN GROUP field containing the value ****, a PRODUCT ID field containing the value 02, TYPE and SUBTYPE fields containing the value **, and a FUNCTION TYPE field containing the value 0.

If the Interchange Interface process finds a record using one of these keys, that record is stored in memory, and is used to log all BASE24-pos records to the ILF. Note that only one record is retrieved; that is, if a record is found using the first set of key field values, the second search is not performed. If the Interchange Interface process does not find a record using one of these keys, the Interchange Interface process logs all tokens in BASE24-pos messages to the ILF by default.

## BASE24-pos Authorization Module

During initialization, the BASE24-pos Authorization module reads the TKN for the records that the Authorization module should use to log transactions to the PTLF. The Authorization module uses a partial key to retrieve all records where the TOKEN GROUP field contains the value ****, the PRODUCT ID field contains the value 02 (BASE24-pos), and the FUNCTION TYPE field contains the value 0 (logging record).

When the Authorization module writes a record to the PTLF, the Authorization module checks the TKN records that it read into memory to determine what tokens should be logged with the record. To select a TKN record from memory, the

Authorization module uses the HEAD.REC-TYP and SET-REC1.SETL-TYP fields from the PTLF record. The values for these fields and the corresponding TYPE and SUBTYPE field values in the TKN are shown in the table below.

| HEAD.REC-TYP | SET-REC1.SETL-TYP | TYPE | SUBTYPE |
|---|---|---|---|
| 01 (Financial) | Not applicable | 01 | ** |
| 20 (Exception) | Not applicable | 20 | ** |
| 21 (Exception) | Not applicable | 21 | ** |
| 22 (Exception) | Not applicable | 22 | ** |
| 23 (Exception) | Not applicable | 23 | ** |
| 04 (Administrative) | 0 (Batch Totals) | 04 | 00 |
| 04 (Administrative) | 1 (Shift Totals) | 04 | 01 |
| 04 (Administrative) | 2 (Day Totals) | 04 | 02 |
| 04 (Administrative) | 3 (Network Totals) | 04 | 03 |
| 04 (Administrative) | 4 (Clerk Totals) | 04 | 04 |
| 04 (Administrative) | 8 (Second Services) | 04 | 08 |
| 04 (Administrative) | 9 (First Services) | 04 | 09 |

If a TKN record is not found and the HEAD.REC-TYP field contains the value 01, 20, 21, 22, or 23, the Authorization module checks for a TKN record with ** in the TYPE field. If the Authorization module still cannot find a TKN record, the Authorization module logs all token data.

If a TKN record is not found and the HEAD.REC-TYP field contains the value 04, the Authorization module checks for a TKN record with the TYPE specified in the table above and the value ** in the SUBTYPE field. If the Authorization module still cannot find a TKN record, the Authorization module checks for a TKN record with the value ** in both the TYPE and SUBTYPE fields. If a record is still not found, the Authorization module logs all token data.

In summary, BASE24-pos Authorization performs the following searches when looking for a TKN record to use for logging a record to the PTLF:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | **** | 02 | See table above | See table above | 0 |
| Second choice | **** | 02 | See table above | ** | 0 |
| Third choice | **** | 02 | ** | ** | 0 |

If the Authorization process has not found a record after performing these searches, the Authorization process logs all tokens in the internal message to the PTLF.

## BASE24-telebanking Integrated Authorization Server Process

During initialization, the BASE24-telebanking Integrated Authorization Server process reads the TKN for the record it should use to log transactions to the ITLF. The Integrated Authorization Server process uses the following key to retrieve the TKN record: the TOKEN GROUP field contains the value ****, the PRODUCT ID field contains the value 14 (BASE24-telebanking), the FUNCTION TYPE field contains the value 0 (logging record), the TYPE field contains the value 00, and the SUBTYPE field contains the value **.

When the Integrated Authorization Server process moves token data from the internal message to fields in the Internal Transaction Data (ITD), the Integrated Authorization Server process checks the TKN record that it read into memory to determine what tokens should be moved to the ITD field that is logged to the ITLF and what tokens should be moved to the ITD field that is not logged to the ITLF.

If a TKN record is not found, the Integrated Authorization Server process moves all token data from the internal message to the ITD field that is logged to the ITLF.

## BASE24-teller Authorization and Device Handler Processes

During initialization, the BASE24-teller Authorization process and BASE24-teller Device Handler processes read the TKN for the records that they should use to log transactions to the TTLF. These processes use a partial key to retrieve all records where the TOKEN GROUP field contains the value ****, the PRODUCT ID field contains the value 03 (BASE24-teller), and the FUNCTION TYPE field contains the value 0 (logging record).

When the Authorization process or a Device Handler process writes a record to the TTLF, it checks the TKN records that it read into memory to determine what tokens should be logged with the record. To select a TKN record from memory, the Authorization process or the Device Handler process uses the HEAD.REC-TYP field and the first two positions of the HEAD.ACCT.TRAN-TYP-CDE field from the TTLF record. The values for these fields and the corresponding TYPE and SUBTYPE field values in the TKN are shown in the table below.

| REC-TYP | TRAN-TYP-CDE (bytes 1 and 2) | TYPE | SUBTYPE |
|---|---|---|---|
| 01 (Financial) | Not applicable | 01 | ** |
| 02 (File Inquiry/Update) | 30 (PBF inquiry) | 02 | 30 |
| 02 (File Inquiry/Update) | 31 (PBF short inquiry) | 02 | 31 |
| 02 (File Inquiry/Update) | 32 (SPF inquiry) | 02 | 32 |
| 02 (File Inquiry/Update) | 33 (CAF inquiry) | 02 | 33 |
| 02 (File Inquiry/Update) | 34 (NBF inquiry) | 02 | 34 |
| 02 (File Inquiry/Update) | 35 (NBF print) | 02 | 35 |
| 02 (File Inquiry/Update) | 36 (NBF reprint) | 02 | 36 |
| 02 (File Inquiry/Update) | 37 (WHFF inquiry) | 02 | 37 |
| 02 (File Inquiry/Update) | 73 (Change CAF card status) | 02 | 73 |
| 02 (File Inquiry/Update) | 74 (Change CAF/PBF status) | 02 | 74 |
| 02 (File Inquiry/Update) | 75 (Verify PIN) | 02 | 75 |

| REC-TYP | TRAN-TYP-CDE (bytes 1 and 2) | TYPE | SUBTYPE |
|---|---|---|---|
| 02 (File Inquiry/Update) | 80 (Add stop) | 02 | 80 |
| 02 (File Inquiry/Update) | 81 (Delete stop) | 02 | 81 |
| 02 (File Inquiry/Update) | 82 (Change PBF account status) | 02 | 82 |
| 02 (File Inquiry/Update) | 83 (Change PBF stop pay/ warning status) | 02 | 83 |
| 02 (File Inquiry/Update) | 84 (Add warning) | 02 | 84 |
| 02 (File Inquiry/Update) | 85 (Add hold) | 02 | 85 |
| 02 (File Inquiry/Update) | 86 (Delete hold) | 02 | 86 |
| 02 (File Inquiry/Update) | 87 (Add float) | 02 | 87 |
| 02 (File Inquiry/Update) | 88 (Delete float) | 02 | 88 |
| 02 (File Inquiry/Update) | 89 (Delete warning) | 02 | 89 |
| 04 (Administrative) | 90 (Logon) | 04 | 90 |
| 04 (Administrative) | 91 (Logoff) | 04 | 91 |
| 04 (Administrative) | 92 (Sign on) | 04 | 92 |
| 04 (Administrative) | 93 (Sign off) | 04 | 93 |
| 04 (Administrative) | 94 (User-defined) | 04 | 94 |
| 04 (Administrative) | 95 (User-defined) | 04 | 95 |
| 04 (Administrative) | 96 (User-defined) | 04 | 96 |
| 04 (Administrative) | 97 (User-defined) | 04 | 97 |
| 04 (Administrative) | 98 (User-defined) | 04 | 98 |
| 04 (Administrative) | 99 (User-defined) | 04 | 99 |

If a TKN record is not found, the process checks for a TKN record with the TYPE specified in the table above and the value ** in the SUBTYPE field. If a TKN record still cannot be found, the process checks for a TKN record with the value ** in both the TYPE and SUBTYPE fields. If a record is still not found, the process logs all token data.

In summary, BASE24-teller Authorization and BASE24-teller Device Handler processes perform the following searches when looking for a TKN record to use for logging a record to the TTLF:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | **** | 03 | See table above | See table above | 0 |
| Second choice | **** | 03 | See table above | ** | 0 |
| Third choice | **** | 03 | ** | ** | 0 |

If the process has not found a record after performing these searches, the Authorization process logs all tokens in the internal message to the TTLF.

## BASE24-teller Default Token Records

Because BASE24-teller makes extensive use of tokens, ACI provides default TKN records for BASE24-teller. These TKN records are located in a file named TKN on the TR*xx*MISC subvolume, where *xx* is the number of the current release. This file can either be moved to the appropriate data file subvolume, or the contents can be copied to the TKN using the HP NonStop utility FUP. The table on the following page shows the default TKN records for logging transactions to the TTLF.

| FUNC-TYP | PROD-ID | TKN-GRP | TYP | SUB-TYP | ORDR-FLG | Tokens Logged to the TTLF |
|---|---|---|---|---|---|---|
| 0 | 03 | **** | 01 | ** | N | T1 (Financial token), TC (Override token), T3 (NBF token) |
| 0 | 03 | **** | 02 | 30 | N | T4 (PBF Inquiry token), TB (Account token), TC (Override token) |
| 0 | 03 | **** | 02 | 31 | N | TB (Account token), TC (Override token) |
| 0 | 03 | **** | 02 | 32 | N | T6 (SPF Inquiry token), TC (Override token) |
| 0 | 03 | **** | 02 | 33 | N | T1 (CAF Inquiry token), TC (Override token) |
| 0 | 03 | **** | 02 | 34 | N | T3 (NBF token), TC (Override token) |
| 0 | 03 | **** | 02 | 35 | N | T3 (NBF token), TC (Override token) |
| 0 | 03 | **** | 02 | 36 | N | T3 (NBF token), TC (Override token) |
| 0 | 03 | **** | 02 | 37 | N | T8 (WHFF Inquiry token), TC (Override token) |
| 0 | 03 | **** | 02 | 73 | N | T2 (CAF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 74 | N | T2 (CAF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 80 | N | T7 (SPF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 81 | N | T7 (SPF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 82 | N | T5 (PBF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 83 | N | T5 (PBF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 84 | N | T9 (WHFF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 85 | N | T9 (WHFF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 86 | N | T9 (WHFF Update token), TC (Override token) |

| FUNC-TYP | PROD-ID | TKN-GRP | TYP | SUB-TYP | ORDR-FLG | Tokens Logged to the TTLF |
|---|---|---|---|---|---|---|
| 0 | 03 | **** | 02 | 87 | N | T9 (WHFF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 88 | N | T9 (WHFF Update token), TC (Override token) |
| 0 | 03 | **** | 02 | 89 | N | T9 (WHFF Update token), TC (Override token) |
| 0 | 03 | **** | 04 | ** | N | TA (Administrative token) |

## Operator Procedures

To configure the tokens to be logged to a transaction log file (TLF, PTLF, ITLF, or ILF), perform the steps below.

**Note:** For BASE24-teller, default TKN logging records are provided. If you are using the BASE24-teller default records, you should update the default records rather than add new records. Refer to the topic "Updating TKN Logging Records" on the following page for more information.

1. Enter TKN in the FILE DESTINATION field at the bottom of the CRT access screen. From a menu screen, press the **F1** key. From a file screen, press the **F16** key. TKN screen 1 is displayed, as shown below. For detailed field descriptions, refer to the ***BASE24 Base Files Maintenance Manual***.

```
BASE24-BASE  TOKEN FILE               LLLL        YY/MM/DD  HH:MM  01 OF 04
TOKEN GROUP: ****                        PRODUCT ID:
       TYPE: **                             SUBTYPE: **


                   FUNCTION TYPE:   (0) - LOG CONFIGURATION RECORD
                                    (1) - EXTRACT CONFIGURATION RECORD
                                    (2) - ISO MESSAGE CONFIGURATION RECORD









********************************** BASE24 **********************************
                  FILE DESTINATION:        NEW LOGICAL NETWORK ID:
                   F12-HELP
```

2. Enter the key information in the following fields:

   TOKEN GROUP
   PRODUCT ID
   TYPE
   SUBTYPE

   For more information on what values to enter in these fields, refer to the topic "Key Field Settings," immediately before these operator procedures.

3. Enter the value 0 in the FUNCTION TYPE field.

4. Press the **F9** key. TKN screen 2 is displayed, as shown below. For detailed field descriptions, refer to the ***BASE24 Base Files Maintenance Manual***.

```
BASE24-BASE  TOKEN FILE                LLLL        YY/MM/DD  HH:MM  02 OF 04
 TOKEN GROUP: ****                         PRODUCT ID:    (***)
        TYPE: ** (DEFAULT)                    SUBTYPE: ** (DEFAULT)

 TKN  TRAN    TOKEN                       TKN  TRAN    TOKEN
 ID   LOG     DESCRIPTION                 ID   LOG     DESCRIPTION

 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________
 __    _     ____________________________  __    _     ____________________________

 ********************************** BASE24 **********************************
                   FILE DESTINATION:        NEW LOGICAL NETWORK ID:
  F7-DEFAULTS  F12-HELP  F14-SORT IN LOG ORDER  F15-SORT IN TOKEN ID ORDER
VIRTUAL SCREEN 01 OF 01 OF TOKEN IDS FOR THIS RECORD
```

5. Press the **F7** key. When the **F7** key is pressed, the TKN requester displays all tokens that have been defined in COBTKN for the product identified in the PRODUCT ID field. Up to 24 tokens are displayed at one time. By default, the TRAN LOG field for each token is set to the value Y.
6. For each token on the screen, determine whether the token should be logged. If the token should be logged, no changes are necessary. If the token should not be logged, use the **Tab** key to move to the TRAN LOG field for the token and enter the value N.
7. Press the **F3** key to add the record.
8. If the screen contained 24 tokens and more tokens exist, a message is displayed at the bottom of the screen indicating that there are more tokens. To display the next page of tokens, press the **Shift-F6** keys.
9. For each token on the screen, determine whether the token should be logged. If the token should be logged, no changes are necessary. If the token should not be logged, use the **Tab** key to move to the TRAN LOG field for the token and enter the value N.
10. Repeat steps 8 and 9 until all tokens have been checked and updated if needed.
11. Press the **F5** key to update the record.

# Updating TKN Logging Records

To update the tokens to be logged to a transaction log file (TLF, PTLF, TTLF, ITLF, or ILF), perform the steps below. These steps assume the TKN record already exists.

1. Enter TKN in the FILE DESTINATION field at the bottom of the CRT access screen. From a menu screen, press the **F1** key. From a file screen, press the **F16** key. TKN screen 1 is displayed.
2. Enter the key information in the following fields:

   TOKEN GROUP
   PRODUCT ID
   TYPE
   SUBTYPE

   For more information on what values to enter in these fields, refer to the previous topic, "Key Field Settings."
3. Enter the value 0 in the FUNCTION TYPE field.
4. Press the **F2** key. TKN screen 2 is displayed, with the information for the record that was read.
5. For each token on the screen, determine whether the token should be logged. If the token should be logged, enter the value Y in the TRAN LOG field. If the token should not be logged, enter the value N in the TRAN LOG field. Use the **Tab** key to move to the TRAN LOG field for the token and enter the appropriate value, if the value needs to be changed.
6. If the screen contained 24 tokens and more tokens exist, a message is displayed at the bottom of the screen indicating that there are more tokens. To display the next page of tokens, press the **Shift-F6** keys.
7. Repeat steps 5 and 6 until all tokens have been checked and updated if needed.
8. Press the **F5** key to update the record.

# Specifying Tokens to be Extracted

The tokens that are extracted from the transaction log files can be used by the host for host processing. Any token that is logged to a transaction log file can be extracted.

When the Super Extract process extracts transaction log file records, it checks its internal table of TKN records to determine whether a token record for the specific transaction exists. Depending on what the process finds in its internal table, the following tokens are extracted:

- If no record is found, no tokens are extracted from the transaction log file.
- If a TKN record is found, the Super Extract process extracts those tokens which are specified to be extracted in the TKN record.

**Note:** If the extract is a fixed format extract, and a token that is specified to be extracted in the TKN record is not found in the transaction log file record, the Super Extract process blank-fills the token in the extract record. For more information on fixed format extracts, refer to the ***BASE24 Refresh and Extract Operators Manual***.

## Key Field Settings

The key to records in the TKN is a combination of all of the fields on TKN screen 1: TOKEN GROUP, PRODUCT ID, TYPE, SUBTYPE, and FUNCTION TYPE. An understanding of how the Super Extract process uses these fields is essential before creating TKN records.

During initialization, the Super Extract process reads the TKN for the records that the Super Extract process should use to extract log records from the TLF, PTLF, TTLF, ITLF, and ILF. The Super Extract process uses a partial key to retrieve all records where the FUNCTION TYPE field contains the value 1 (extract record).

As the Super Extract process extracts each log file record, the Super Extract process checks the TKN records that it read into memory to determine what tokens should be extracted with the record. To select a TKN record from memory, the Super Extract process performs as described below, depending on the file being extracted.

## ILF Extracts

The Super Extract process performs the following processing for each record that is extracted from the ILF.

The Super Extract process determines a value for the PRODUCT ID field based on the REC-TYP field in the ILF. If the REC-TYP field contains the value 1, the Super Extract process uses the value 01 (BASE24-atm) for the PRODUCT ID field. If the REC-TYP field contains the value 2, the Super Extract process uses the value 02 (BASE24-pos) for the PRODUCT ID field.

The Super Extract process determines a value for the TOKEN GROUP field using the value from the ILF.GRP field in the ECF (that is, the SWITCH FIID field on ECF screen 1). If the ILF.GRP field contains the value ALL$\not b$ (where $\not b$ denotes a space), the Super Extract process uses the following criteria to select a TKN record:

1. A TOKEN GROUP field containing the value ****, a PRODUCT ID field containing the value as determined above, TYPE and SUBTYPE fields containing the value **, and a FUNCTION TYPE field containing the value 1.
2. If no record is found, the Super Extract process does not extract any token data from the ILF record.

If the ILF.GRP field in the ECF contains a value other than ALL$\not b$ (where $\not b$ denotes a space), the Super Extract process uses the following criteria to select a TKN record:

1. A TOKEN GROUP field containing the value from the ILF.GRP field, a PRODUCT ID field containing the value as determined above, TYPE and SUBTYPE fields containing the value **, and a FUNCTION TYPE field containing the value 1.
2. A TOKEN GROUP field containing the value ****, a PRODUCT ID field containing the value as determined above, TYPE and SUBTYPE fields containing the value **, and a FUNCTION TYPE field containing the value 1.
3. If no record is found, the Super Extract process does not extract any token data from the ILF record.

## TLF Extracts

The Super Extract process performs the following processing for each record that is extracted from the TLF.

The Super Extract process determines values for the TYPE and SUBTYPE fields, based on the HEAD.REC-TYP and *subtype*.ADMIN-CDE fields from the TLF record, where *subtype* is TERM-SETL, TERM-CASH, or SETL-TTL administrative record types. The values for these fields and the corresponding TYPE and SUBTYPE field values in the TKN are shown in the table below.

| **HEAD.REC-TYP** | ***subtype*.ADMIN-CDE** | **TYPE** | **SUBTYPE** |
|---|---|---|---|
| 01 (Financial) | Not applicable | 01 | ** |
| 20 (Exception) | Not applicable | 20 | ** |
| 21 (Exception) | Not applicable | 21 | ** |
| 04 (Administrative) | 01 (Initialize cash by Device Handler process) | 04 | 01 |
| 04 (Administrative) | 02 (Initialize cash by DCT Server process) | 04 | 02 |
| 04 (Administrative) | 03 (Increment cash by Device Handler process) | 04 | 03 |
| 04 (Administrative) | 04 (Increment cash by DCT Server process) | 04 | 04 |
| 04 (Administrative) | 05 (Terminal Cutover by Device Handler process) | 04 | 05 |
| 04 (Administrative) | 06 (Terminal Cutover by DCT Server process) | 04 | 06 |
| 04 (Administrative) | 07 (Decrement cash by Device Handler process) | 04 | 07 |
| 04 (Administrative) | 08 (Decrement cash by DCT Server process) | 04 | 08 |
| 04 (Administrative) | 09 (Forced Cutover) | 04 | 09 |

| HEAD.REC-TYP | *subtype*.ADMIN-CDE | TYPE | SUBTYPE |
|---|---|---|---|
| 04 (Administrative) | 20 (Terminal Totals by DCT Server process) | 04 | 20 |
| 04 (Administrative) | 21 (Terminal Totals by Device Handler process) | 04 | 21 |
| 04 (Administrative) | 22 (Terminal Totals by Settlement Initiator) | 04 | 22 |

After determining the values for the TYPE and SUBTYPE fields, the Super Extract process determines a value for the TOKEN GROUP field using the value from the TLF.GRP field in the ECF.

If the TLF.GRP field contains the value ALL$\not b$ (where $\not b$ denotes a space), the Super Extract process uses the following criteria to select a TKN record:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | **** | 01 | See table above | See table above | 1 |
| Second choice | **** | 01 | See table above | ** | 1 |
| Third choice | **** | 01 | ** | ** | 1 |

If no record is found, the Super Extract process does not extract any token data from the TLF record.

If the TLF.GRP field in the ECF contains a value other than ALL␣ (␣ denotes a space), the Super Extract process uses the following criteria to select a TKN record:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | TLF.GRP field | 01 | See table above | See table above | 1 |
| Second choice | TLF.GRP field | 01 | See table above | ** | 1 |
| Third choice | TLF.GRP field | 01 | ** | ** | 1 |
| Fourth choice | **** | 01 | See table above | See table above | 1 |
| Fifth choice | **** | 01 | See table above | ** | 1 |
| Sixth choice | **** | 01 | ** | ** | 1 |

If no record is found, the Super Extract process does not extract any token data from the TLF record.

## PTLF Extracts

The Super Extract process performs this processing for each record that is extracted from the PTLF.

The Super Extract process determines values for the TYPE and SUBTYPE fields, based on the HEAD.REC-TYP and SET-REC1.SETL-TYP fields from the PTLF record. The values for these fields and the corresponding TYPE and SUBTYPE field values in the TKN are shown in the table below.

| HEAD.REC-TYP | SET-REC1.SETL-TYP | TYPE | SUBTYPE |
|---|---|---|---|
| 01 (Financial) | Not applicable | 01 | ** |

| HEAD.REC-TYP | SET-REC1.SETL-TYP | TYPE | SUBTYPE |
|---|---|---|---|
| 20 (Exception) | Not applicable | 20 | ** |
| 21 (Exception) | Not applicable | 21 | ** |
| 22 (Exception) | Not applicable | 22 | ** |
| 23 (Exception) | Not applicable | 23 | ** |
| 04 (Administrative) | 0 (Batch Totals) | 04 | 00 |
| 04 (Administrative) | 1 (Shift Totals) | 04 | 01 |
| 04 (Administrative) | 2 (Day Totals) | 04 | 02 |
| 04 (Administrative) | 3 (Network Totals) | 04 | 03 |
| 04 (Administrative) | 4 (Clerk Totals) | 04 | 04 |
| 04 (Administrative) | 8 (Second Services) | 04 | 08 |
| 04 (Administrative) | 9 (First Services) | 04 | 09 |

After determining the values for the TYPE and SUBTYPE fields, the Super Extract process determines a value for the TOKEN GROUP field using the value from the PTLF.GRP field in the ECF. If the PTLF.GRP field contains a value other than ALL␣ (␣ denotes a space), the Super Extract process uses the value from the PTLF.GRP field as the value for the TOKEN GROUP field. If the PTLF.GRP field contains the value ALL␣ (␣ denotes a space), the Super Extract process uses the value **** for the TOKEN GROUP field. In this case, the Super Extract process uses the following criteria to select a TKN record:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | **** | 02 | See table above | See table above | 1 |
| Second choice | **** | 02 | See table above | ** | 1 |
| Third choice | **** | 02 | ** | ** | 1 |

If no record is found, the Super Extract process does not extract any token data from the PTLF record.

If the PTLF.GRP field in the ECF contains a value other than ALL␣ (␣ denotes a space), the Super Extract process uses the following criteria to select a TKN record:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | PTLF.GRP field | 02 | See table above | See table above | 1 |
| Second choice | PTLF.GRP field | 02 | See table above | ** | 1 |
| Third choice | PTLF.GRP field | 02 | ** | ** | 1 |
| Fourth choice | **** | 02 | See table above | See table above | 1 |
| Fifth choice | **** | 02 | See table above | ** | 1 |
| Sixth choice | **** | 02 | ** | ** | 1 |

If no record is found, the Super Extract process does not extract any token data from the PTLF record.

## ITLF Extracts

The Super Extract process performs the following processing for each record that is extracted from the ITLF.

The key used by the Super Extract process to retrieve the TKN record depends on the value in the TLF.GRP field in the BASE24-telebanking segment of the ECF.

If the TLF.GRP field in the BASE24-telebanking segment of the ECF contains the value ALL␣ (where ␣ denotes a space), the Super Extract process uses the following key: a value of **** in the TOKEN GROUP field, the value 14

(BASE24-telebanking) in the PRODUCT ID field, the value 1 (extract record) in the FUNCTION TYPE field, the value 00 in the TYPE field, and the value ** in the SUBTYPE field.

If the TLF.GRP field in the BASE24-telebanking segment of the ECF contains a value other than ALL␣ (where ␣ denotes a space), the Super Extract process uses the following key: the value from the TLF.GRP field in the TOKEN GROUP field, the value 14 (BASE24-telebanking) in the PRODUCT ID field, the value 1 (extract record) in the FUNCTION TYPE field, the value 00 in the TYPE field, and the value ** in the SUBTYPE field.

If a TKN record is not found, the Super Extract process does not extract any token data from the ITLF record.

## TTLF Extracts

The Super Extract process performs the processing for each record that is extracted from the TTLF.

The Super Extract process determines values for the TYPE and SUBTYPE fields based on the HEAD.REC-TYP and HEAD.ACCT.TRAN-TYP-CDE fields from the TTLF record. The values for these fields and the corresponding TYPE and SUBTYPE field values in the TKN are shown in the table below.

| REC-TYP | TRAN-TYP-CDE (bytes 1 and 2) | TYPE | SUBTYPE |
|---|---|---|---|
| 01 (Financial) | Not applicable | 01 | ** |
| 02 (File Inquiry/Update) | 30 (PBF inquiry) | 02 | 30 |
| 02 (File Inquiry/Update) | 31 (PBF short inquiry) | 02 | 31 |
| 02 (File Inquiry/Update) | 32 (SPF inquiry) | 02 | 32 |
| 02 (File Inquiry/Update) | 33 (CAF inquiry) | 02 | 33 |
| 02 (File Inquiry/Update) | 34 (NBF inquiry) | 02 | 34 |
| 02 (File Inquiry/Update) | 35 (NBF print) | 02 | 35 |
| 02 (File Inquiry/Update) | 36 (NBF reprint) | 02 | 36 |
| 02 (File Inquiry/Update) | 37 (WHFF inquiry) | 02 | 37 |

| REC-TYP | TRAN-TYP-CDE (bytes 1 and 2) | TYPE | SUBTYPE |
|---|---|---|---|
| 02 (File Inquiry/Update) | 73 (Change CAF card status) | 02 | 73 |
| 02 (File Inquiry/Update) | 74 (Change CAF/PBF status) | 02 | 74 |
| 02 (File Inquiry/Update) | 80 (Add stop) | 02 | 80 |
| 02 (File Inquiry/Update) | 81 (Delete stop) | 02 | 81 |
| 02 (File Inquiry/Update) | 82 (Change PBF account status) | 02 | 82 |
| 02 (File Inquiry/Update) | 83 (Change PBF stop pay/warning status) | 02 | 83 |
| 02 (File Inquiry/Update) | 84 (Add warning) | 02 | 84 |
| 02 (File Inquiry/Update) | 85 (Add hold) | 02 | 85 |
| 02 (File Inquiry/Update) | 86 (Delete hold) | 02 | 86 |
| 02 (File Inquiry/Update) | 87 (Add float) | 02 | 87 |
| 02 (File Inquiry/Update) | 88 (Delete float) | 02 | 88 |
| 02 (File Inquiry/Update) | 89 (Delete warning) | 02 | 89 |
| 04 (Administrative) | 90 (Logon) | 04 | 90 |
| 04 (Administrative) | 91 (Logoff) | 04 | 91 |
| 04 (Administrative) | 92 (Sign on) | 04 | 92 |
| 04 (Administrative) | 93 (Sign off) | 04 | 93 |
| 04 (Administrative) | 94 (User-defined) | 04 | 94 |
| 04 (Administrative) | 95 (User-defined) | 04 | 95 |
| 04 (Administrative) | 96 (User-defined) | 04 | 96 |
| 04 (Administrative) | 97 (User-defined) | 04 | 97 |
| 04 (Administrative) | 98 (User-defined) | 04 | 98 |

| REC-TYP | TRAN-TYP-CDE (bytes 1 and 2) | TYPE | SUBTYPE |
|---|---|---|---|
| 04 (Administrative) | 99 (User-defined) | 04 | 99 |

After determining the values for the TYPE and SUBTYPE fields, the Super Extract process determines a value for the TOKEN GROUP field using the value from the TTLF.GRP field in the ECF.

If the TTLF.GRP field contains the value ALL␣ (␣ denotes a space), the Super Extract process uses the following criteria to select a TKN record:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | **** | 03 | See table above | See table above | 1 |
| Second choice | **** | 03 | See table above | ** | 1 |
| Third choice | **** | 03 | ** | ** | 1 |

If no record is found, the Super Extract process does not extract any token data from the TTLF record.

If the TTLF.GRP field in the ECF contains a value other than ALL␣ (where ␣ denotes a space), the Super Extract process uses the following criteria to select a TKN record:

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | TTLF.GRP field | 03 | See table above | See table above | 1 |
| Second choice | TTLF.GRP field | 03 | See table above | ** | 1 |

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| Third choice | TTLF.GRP field | 03 | ** | ** | 1 |
| Fourth choice | **** | 03 | See table above | See table above | 1 |
| Fifth choice | **** | 03 | See table above | ** | 1 |
| Sixth choice | **** | 03 | ** | ** | 1 |

If no record is found, the Super Extract process does not extract any token data from the TTLF record.

## BASE24-teller Default Token Records

Because BASE24-teller makes extensive use of tokens, ACI provides default TKN records for BASE24-teller. These TKN records are located in a file named TKN on the TR*xx*MISC subvolume, where *xx* is the number of the current release. This file can either be moved to the appropriate data file subvolume, or the contents can be copied to the TKN using the HP NonStop utility FUP. The table below shows the default TKN records for extracting tokens from the TTLF. Tokens are placed in the extract record in the order that they are identified in the last column.

| FUNC-TYP | PROD-ID | TKN-GRP | TYP | SUB-TYP | ORDR-FLG | Tokens Extracted from the TTLF |
|---|---|---|---|---|---|---|
| 1 | 03 | **** | 01 | ** | Y | T1 (Financial token), TC (Override token), T3 (NBF token) |
| 1 | 03 | **** | 02 | 30 | Y | T4 (PBF Inquiry token), TB (Account token), TC (Override token) |

| FUNC-TYP | PROD-ID | TKN-GRP | TYP | SUB-TYP | ORDR-FLG | Tokens Extracted from the TTLF |
|---|---|---|---|---|---|---|
| 1 | 03 | **** | 02 | 31 | Y | TB (Account token), TC (Override token) |
| 1 | 03 | **** | 02 | 32 | Y | T6 (SPF Inquiry token), TC (Override token) |
| 1 | 03 | **** | 02 | 33 | Y | T1 (CAF Inquiry token), TC (Override token) |
| 1 | 03 | **** | 02 | 34 | Y | T3 (NBF token), TC (Override token) |
| 1 | 03 | **** | 02 | 35 | Y | T3 (NBF token), TC (Override token) |
| 1 | 03 | **** | 02 | 36 | Y | T3 (NBF token), TC (Override token) |
| 1 | 03 | **** | 02 | 37 | Y | T8 (WHFF Inquiry token), TC (Override token) |
| 1 | 03 | **** | 02 | 73 | Y | T2 (CAF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 74 | Y | T2 (CAF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 80 | Y | T7 (SPF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 81 | Y | T7 (SPF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 82 | Y | T5 (PBF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 83 | Y | T5 (PBF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 84 | Y | T9 (WHFF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 85 | Y | T9 (WHFF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 86 | Y | T9 (WHFF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 87 | Y | T9 (WHFF Update token), TC (Override token) |

| FUNC-TYP | PROD-ID | TKN-GRP | TYP | SUB-TYP | ORDR-FLG | Tokens Extracted from the TTLF |
|---|---|---|---|---|---|---|
| 1 | 03 | **** | 02 | 88 | Y | T9 (WHFF Update token), TC (Override token) |
| 1 | 03 | **** | 02 | 89 | Y | T9 (WHFF Update token), TC (Override token) |
| 1 | 03 | **** | 04 | ** | Y | TA (Administrative token) |

## Operator Procedures

To configure the tokens to be extracted from a transaction log file (TLF, PTLF, ITLF, or ILF), perform the steps below.

**Note:** For BASE24-teller, default TKN extract records are provided. If you are using the BASE24-teller default records, you should update the default records rather than add new records. Refer to the topic "Updating TKN Extract Records" for more information.

1. Enter TKN in the FILE DESTINATION field at the bottom of the CRT access screen. From a menu screen, press the **F1** key. From a file screen, press the **F16** key. TKN screen 1 is displayed.
2. Enter the key information in the following fields:

   TOKEN GROUP
   PRODUCT ID
   TYPE
   SUBTYPE

   For more information on what values to enter in these fields, refer to the topic "Key Field Settings," immediately before these operator procedures.
3. Enter the value 1 in the FUNCTION TYPE field.
4. Press the **F9** key. TKN screen 3 is displayed, as shown below. For detailed field descriptions, refer to the ***BASE24 Base Files Maintenance Manual***.

```
BASE24-BASE  TOKEN FILE                LLLL        YY/MM/DD  HH:MM  03 OF 04
TOKEN GROUP: ****                         PRODUCT ID:    (***)
       TYPE: ** (DEFAULT)                    SUBTYPE: ** (DEFAULT)
 ORDER FLAG: Y (Y/N)
TKN  EXTR   TOKEN                       TKN  EXTR   TOKEN
ID   ORDER  DESCRIPTION                 ID   ORDER  DESCRIPTION

__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________

********************************** BASE24 ***********************************
                   FILE DESTINATION:        NEW LOGICAL NETWORK ID:
 F7-DEFAULTS  F12-HELP  F14-SORT IN EXTR ORDER  F15-SORT IN TOKEN ID ORDER
VIRTUAL SCREEN 01 OF 01 OF TOKEN IDS FOR THIS RECORD
```

5. Press the **F7** key. When the **F7** key is pressed, the TKN requester process displays all tokens that have been defined in COBTKN for the product identified in the PRODUCT ID field. Up to 24 tokens are displayed at one time. By default, the EXTR ORDER field for each token is set to contain blanks, indicating the token should not be extracted.

6. Determine whether the tokens should be extracted in a particular order. If the tokens should be sorted and extracted in a specific order, allow the ORDER FLAG field to default to Y. If the tokens should be extracted in the same order in which they appear in the log file record (that is, the order in which they were added to the internal message), use the **Tab** key to move to the ORDER FLAG field and enter the value N.

   **Note:** When the ORDER FLAG field contains the value N, all tokens in the transaction log file record are extracted. You cannot specify that only certain tokens be extracted without specifying an order in which they should be extracted.

7. If the tokens should be extracted in order, determine for each token on the screen whether the token should be extracted. If the token should not be extracted, no changes are necessary. If the token should be extracted, use the **Tab** key to move to the EXTR ORDER field for the token and enter a value ranging from 1 to the number of tokens present in the record (maximum of 360). The numbers assigned must be consecutive, starting at 1 and with no gaps between numbers. This number indicates the order in which the token is extracted from the log record, if the ORDER FLAG field is set to Y. If the

ORDER FLAG field is set to N, the order and the tokens extracted from the log record depend on whether the extract is variable length or fixed length. For more information on how the Super Extract process handles tokens in variable- and fixed-length extracts, refer to the ***BASE24 Refresh and Extract Operators Manual***. Use the **Tab** key to move to the EXTR ORDER field for the token and make any necessary changes.

8. Press the **F3** key to add the record. If the ORDER FLAG field contains the value N, no further steps are necessary. If the ORDER FLAG field contains the value Y, continue with the remaining steps.
9. If the screen contained 24 tokens and more tokens exist, a message is displayed at the bottom of the screen indicating that there are more tokens. To display the next page of tokens, press the **Shift-F6** keys.
10. For each token on the screen, determine whether the token should be extracted. If the token should not be extracted, no changes are necessary. If the token should be extracted, use the **Tab** key to move to the EXTR ORDER field for the token and enter a value ranging from 1 to the number of tokens present in the record (maximum of 360). This number indicates the order in which the token is extracted from the log record.
11. Repeat steps 9 and 10 until all tokens have been checked and updated if needed.
12. Press the **F5** key to update the record.

## Updating TKN Extract Records

To update the tokens to be extracted from a transaction log file (TLF, PTLF, TTLF, ITLF, or ILF), perform the steps below. These steps assume the TKN record already exists.

1. Enter TKN in the FILE DESTINATION field at the bottom of the CRT access screen. From a menu screen, press the **F1** key. From a file screen, press the **F16** key. TKN screen 1 is displayed.
2. Enter the key information in the following fields:

   TOKEN GROUP
   PRODUCT ID
   TYPE
   SUBTYPE

   For more information on what values to enter in these fields, refer to the topic "Key Field Settings."

3. Enter the value 1 in the FUNCTION TYPE field.
4. Press the **F2** key. TKN screen 3 is displayed, with the values from the record identified by the key fields.
5. Determine whether the tokens should be extracted in a particular order. If the tokens should be sorted and extracted in a specific order, the ORDER FLAG field must contain the value Y. If the tokens should be extracted in the same order in which they appear in the log file record (that is, the order in which they were added to the internal message), the ORDER FLAG field must contain the value N. Use the **Tab** key to move to the ORDER FLAG field and enter the appropriate value, if necessary.

   **Note:** When the ORDER FLAG field contains the value N, all tokens in the transaction log file record are extracted. You cannot specify that only certain tokens be extracted without specifying an order in which they should be extracted.
6. If the tokens should be extracted in order, determine for each token on the screen whether the token should be extracted. If the token should not be extracted, the EXTR ORDER field for the token must not contain a value. If the token should be extracted, the EXTR ORDER field for the token must contain a value ranging from 1 to the number of tokens present in the record. The numbers assigned must be consecutive, starting at 1 and with no gaps between numbers. This number indicates the order in which the token is extracted from the log record, if the ORDER FLAG field is set to Y. If the ORDER FLAG field is set to N, all tokens are extracted from the log record in the order in which they appear in the record. Use the **Tab** key to move to the EXTR ORDER field for the token and make any necessary changes.
7. Press the **F5** key to update the record. If the ORDER FLAG field contains the value N, no further steps are necessary. If the ORDER FLAG field contains the value Y, continue with the remaining steps.
8. If the screen contained 24 tokens and more tokens exist, a message is displayed at the bottom of the screen indicating that there are more tokens. To display the next page of tokens, press the **Shift-F6** keys.
9. Repeat steps 6 through 8 until all tokens have been checked.
10. Press the **F5** key to update the record.

# Specifying Tokens to be Sent in the External Message

The tokens that are sent to the host in the external message determine what information the host has available when authorizing transactions. Any token which is carried in the internal message can be sent in the external message.

When the ISO Host Interface process or BIC ISO Interface process creates an external message, it checks its internal table of TKN records to determine whether a token record for the specific transaction exists. Depending on what the process finds in its internal table, the following tokens are sent in the message:

- If no record is found, no tokens are sent in the external message.
- If a TKN record is found, the ISO Host Interface process or BIC ISO Interface process sends those tokens which are specified to be sent in the TKN record.

**Note:** If the message is a fixed format message, and a token that is specified to be sent in the TKN record is not found in the internal message, the ISO Host Interface process or BIC ISO Interface process blank-fills the token in the external message. For more information on how the ISO Host Interface process handles fixed format messages, refer to the ***BASE24 External Message Manual***. For more information on how the BIC ISO Interface process handles fixed format messages, refer to the ***BASE24 BIC ISO Standards Manual***.

## Key Field Settings

The key to records in the TKN is a combination of all of the fields on TKN screen 1: TOKEN GROUP, PRODUCT ID, TYPE, SUBTYPE, and FUNCTION TYPE. An understanding of how the processes which read the TKN use these fields is essential before creating TKN records. In the paragraphs that follow, each process that use TKN records to configure external messages is identified, together with the values that the process uses to read TKN records.

### ISO Host Interface Process

During initialization, the ISO Host Interface process reads the TKN for the records that the ISO Host Interface process should use when sending external messages to the host. The ISO Host Interface process uses a partial key to retrieve all records

where the FUNCTION TYPE field contains the value 2 (external message record) and the PRODUCT ID field identifies a product supported by the ISO Host Interface process.

As the ISO Host Interface process prepares each message to send to a host, the ISO Host Interface process checks the TKN records that it read into memory to determine what tokens should be sent with the message. To select a TKN record from memory, the ISO Host Interface process performs the following steps as described below, depending on the product for which the message is being sent.

**BASE24-atm.** The ISO Host Interface process uses a PRODUCT ID field containing the value 01, a TYPE field containing the value 01, a SUBTYPE field containing the value **, and a FUNCTION TYPE field containing the value 2. The ISO Host Interface process then determines a value for the TOKEN GROUP field based on the values in the TKN-GRP field in the EMF or HCF, as described below.

**Note:** If the TKN-GRP field in the EMF contains blanks at the time the record is read during initialization, the ISO Host Interface process uses the value from the TKN-GRP field in the HCF.

1. If an EMF record for the message type and direction exists, the value from the TKN-GRP field in the EMF is used as the value for the TOKEN GROUP field.
2. If a TKN record could not be found in step 1 and the TKN-GRP field did not contain the value ****, the ISO Host Interface process uses the value **** as the value for the TOKEN GROUP field.
3. If a TKN record could not be found in step 2, the ISO Host Interface process does not send any tokens with the message.

**BASE24-pos.** The ISO Host Interface process uses a PRODUCT ID field containing the value 02 and a FUNCTION TYPE field containing the value 2. The ISO Host Interface process then determines values for the TYPE and SUBTYPE fields based on the values in the REC-TYP and SETL-TYP fields in the PSTM. The values for these fields and the corresponding TYPE and SUBTYPE field values in the TKN are shown in the table below.

| REC-TYP | SETL-TYP | TYPE | SUBTYPE |
|---|---|---|---|
| 01 (Financial) | Not applicable | 01 | ** |
| 20 (Exception) | Not applicable | 20 | ** |

| REC-TYP | SETL-TYP | TYPE | SUBTYPE |
|---|---|---|---|
| 21 (Exception) | Not applicable | 21 | ** |
| 22 (Exception) | Not applicable | 22 | ** |
| 23 (Exception) | Not applicable | 23 | ** |
| 04 (Administrative) | 0 (Batch Totals) | 04 | 00 |
| 04 (Administrative) | 1 (Shift Totals) | 04 | 01 |
| 04 (Administrative) | 2 (Day Totals) | 04 | 02 |
| 04 (Administrative) | 3 (Network Totals) | 04 | 03 |
| 04 (Administrative) | 4 (Clerk Totals) | 04 | 04 |
| 04 (Administrative) | 8 (Second Services) | 04 | 08 |
| 04 (Administrative) | 9 (First Services) | 04 | 09 |

The ISO Host Interface process then determines the value for the TOKEN GROUP field based on the value in the TKN-GRP field in the EMF or HCF, as described below. These steps also show how the ISO Host Interface process selects a TKN record.

**Note:** If the TKN-GRP field in the EMF contains blanks at the time the record is read during initialization, the ISO Host Interface process uses the value from the TKN-GRP field in the HCF.

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | TKN-GRP field | 02 | See table above | See table above | 2 |
| Second choice* | TKN-GRP field | 02 | See table above | ** | 2 |
| Third choice | TKN-GRP field | 02 | ** | ** | 2 |

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| Fourth choice | **** | 02 | See table above | See table above | 2 |
| Fifth choice † | **** | 02 | See table above | ** | 2 |
| Sixth choice | **** | 02 | ** | ** | 2 |

* If the SUBTYPE field contained the value ** in step 1, this step is skipped.

† If the SUBTYPE field contained the value ** in step 4, this step is skipped.

If a TKN record could not be found using the above criteria, the ISO Host Interface process does not send any tokens with the message.

**BASE24-telebanking.** The ISO Host Interface process uses a PRODUCT ID field containing the value 14, a TYPE field containing the value 00, a SUBTYPE field containing the value **, and a FUNCTION TYPE field containing the value 2. The ISO Host Interface process then determines a value for the TOKEN GROUP field based on the values in the TKN-GRP field in the EMF or HCF, as described below.

**Note:** If the TKN-GRP field in the EMF contains blanks at the time the record is read during initialization, the ISO Host Interface process uses the value from the TKN-GRP field in the HCF.

1. If an EMF record for the message type and direction exists, the value from the TKN-GRP field in the EMF is used as the value for the TOKEN GROUP field.
2. If a TKN record could not be found in step 1 and the TKN-GRP field did not contain the value ****, the ISO Host Interface process uses the value **** as the value for the TOKEN GROUP field.
3. If a TKN record could not be found in step 2, the ISO Host Interface process does not send any tokens with the message.

**BASE24-teller.** The ISO Host Interface process uses a PRODUCT ID field containing the value 03 and a FUNCTION TYPE field containing the value 2. For the initial searches, the ISO Host Interface process uses a TOKEN GROUP field with the value from the TKN-GRP field in the EMF for the message type and direction. The ISO Host Interface process then determines values for the TYPE and SUBTYPE fields based on the values in the message type, as described below.

**Note:** If the TKN-GRP field in the EMF contains blanks at the time the record is read during initialization, the ISO Host Interface process uses the value from the TKN-GRP field in the HCF.

- If the message is a financial transaction message or a reversal message (0200, 0220, 0420, or 0421), the Host Interface process uses the value 01 for the TYPE field and the value ** for the SUBTYPE field. If a record cannot be found, the ISO Host Interface process uses the value ** for both the TYPE and SUBTYPE fields.
- If the message is a file update/inquiry message (0300 or 0320), the ISO Host Interface process uses the value 02 for the TYPE field and the value from the RQST.TRAN-CDE field in the TSTMH for the SUBTYPE field. If a record cannot be found, the ISO Host Interface process uses the value 02 for the TYPE field and the value ** for the SUBTYPE field. If a record is still not found, the ISO Host Interface process uses the value ** for both the TYPE and SUBTYPE fields.
- If the message is an administrative message (0600 or 0620), the ISO Host Interface process uses the value 04 for the TYPE field and the value from the RQST.TRAN-CDE field in the TSTMH for the SUBTYPE field. If a record cannot be found, the ISO Host Interface process uses the value 04 for the TYPE field and the value ** for the SUBTYPE field. If a record is still not found, the ISO Host Interface process uses the value ** for both the TYPE and SUBTYPE fields.

If a TKN record is not located using any of the previous keys and the default TOKEN GROUP field value **** has not been used (that is, the EMF or HCF TKN-GRP field did not contain the value ****), the ISO Host Interface process performs the same searches as above, except with the value **** in the TOKEN GROUP field.

If a TKN record is not found, the ISO Host Interface process does not send any nonstandard tokens in the message.

**Note:** BASE24-teller uses standard and nonstandard tokens. Each standard token has its own corresponding data element in the external message, rather than being placed in a group data element along with all other token data to be sent. The

BASE24-teller ISO Host Interface process determines whether these tokens are to be sent in the external message based on settings in the External Message File (EMF). The Credit Line token and the Customer Name token are nonstandard tokens. They are configured to be included in the external message using the TKN, using the procedures described in the topic "Operator Procedures."

## BASE24-teller Default Token Records

Because BASE24-teller standard tokens are not configured using the TKN, ACI does not provide default TKN records for external messages.

## BIC ISO Interface Process

During initialization, the BIC ISO Interface process reads the TKN for the records that the BIC ISO Interface process is to use when sending external messages to the host. The BIC ISO Interface process uses a partial key to retrieve all records where the FUNCTION TYPE field contains the value 2 (external message record) and the PRODUCT ID field identifies a product supported by the BIC ISO Interface process.

As the BIC ISO Interface process prepares each message to send to a host, the BIC ISO Interface process checks the TKN records that it read into memory to determine which tokens must be sent with the message. To select a TKN record from memory, the BIC ISO Interface process performs as described below, depending on the product for which the message is being sent.

**BASE24-atm.** The BIC ISO Interface process uses a PRODUCT ID field containing the value 01, a TYPE field containing the value 01, a SUBTYPE field containing the value **, and a FUNCTION TYPE field containing the value 2. The BIC ISO Interface process then determines a value for the TOKEN GROUP field based on the value in the TKN-GRP field in the EMF or the FIID field in the ICF or ICFE, as described below.

**Note:** If the TKN-GRP field in the EMF contains blanks at the time the record is read during initialization, the BIC ISO Interface process uses the value from the FIID field in the ICF or ICFE.

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | TKN-GRP field | 01 | 01 | ** | 2 |
| Second choice | TKN-GRP field | 01 | ** | ** | 2 |
| Third choice | **** | 01 | 01 | ** | 2 |
| Fourth choice | **** | 01 | ** | ** | 2 |

If a TKN record can not be found, the BIC ISO Interface process does not send any tokens with the BASE24-atm message.

**BASE24-pos.** The BIC ISO Interface process uses a PRODUCT ID field containing the value 02, a TYPE field containing the value 01, a SUBTYPE field containing the value **, and a FUNCTION TYPE field containing the value 2. The BIC ISO Interface process determines a value for the TOKEN GROUP field based on the value in the TKN-GRP field in the EMF or the FIID field in the ICF or ICFE, as described below.

**Note:** If the TKN-GRP field in the EMF contains blanks at the time the record is read during initialization, the BIC ISO Interface process uses the value from the FIID field in the ICF or ICFE.

| | TOKEN GROUP | PRODUCT ID | TYPE | SUBTYPE | FUNCTION TYPE |
|---|---|---|---|---|---|
| First choice | TKN-GRP field | 02 | 01 | ** | 2 |
| Second choice | TKN-GRP field | 02 | ** | ** | 2 |
| Third choice | **** | 02 | 01 | ** | 2 |
| Fourth choice | **** | 02 | ** | ** | 2 |

If a TKN record can not be found, the BIC ISO Interface process does not send any tokens with the BASE24-pos message.

## Operator Procedures

To configure the tokens to be sent in the external message, perform the following steps:

1. Enter TKN in the FILE DESTINATION field at the bottom of the CRT access screen. From a menu screen, press the **F1** key. From a file screen, press the **F16** key. TKN screen 1 is displayed.
2. Enter the key information in the following fields:

   TOKEN GROUP
   PRODUCT ID
   TYPE
   SUBTYPE

   For more information on what values to enter in these fields, refer to the topic "Key Field Settings," immediately before these operator procedures.
3. Enter the value 2 in the FUNCTION TYPE field.
4. Press the **F9** key. TKN screen 4 is displayed, as shown below. For detailed field descriptions, refer to the ***BASE24 Base Files Maintenance Manual***.

```
BASE24-BASE  TOKEN FILE                LLLL        YY/MM/DD  HH:MM  04 OF 04
TOKEN GROUP: ****                         PRODUCT ID: 01 (ATM)
       TYPE: ** (DEFAULT)                     SUBTYPE: ** (DEFAULT)
 ORDER FLAG: Y (Y/N)
TKN  SEND   TOKEN                       TKN  SEND   TOKEN
ID   ORDER  DESCRIPTION                 ID   ORDER  DESCRIPTION

__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________
__   ____   ____________________________  __   ____   ____________________________

********************************** BASE24 **********************************
                    FILE DESTINATION:        NEW LOGICAL NETWORK ID:
  F7-DEFAULTS  F12-HELP  F14-SORT IN SEND ORDER  F15-SORT IN TOKEN ID ORDER
VIRTUAL SCREEN 01 OF 01 OF TOKEN IDS FOR THIS RECORD
```

5. Press the **F7** key. When the **F7** key is pressed, the TKN requester displays all tokens that have been defined in COBTKN for the product identified in the PRODUCT ID field. Up to 24 tokens are displayed at one time. By default, the SEND ORDER field for each token is set to contain blanks, indicating the token should not be sent.

6. Determine whether the tokens should be sent in a particular order. If the tokens should be sorted and sent in a specific order, allow the ORDER FLAG field to default to Y. If the tokens should be sent in the same order in which they appear in the internal message (that is, the order in which they were added to the internal message), use the **Tab** key to move to the ORDER FLAG field and enter the value N.

7. If the tokens should be sent in a specific order, determine for each token on the screen whether the token should be sent in the external message. If the token should not be sent, no changes are necessary. If the token should be sent, use the **Tab** key to move to the SEND ORDER field for the token and enter a value ranging from 1 to the number of tokens present in the record (maximum of 360). This number indicates the order in which the token is added to the external message, if the ORDER FLAG field is set to Y.

   If the ORDER FLAG field is set to N, all tokens in the internal message are sent in the external message. The order in which the tokens are sent depends on the value in the product-specific MESSAGE FORMAT field in the Host Configuration File (HCF).

   If the product-specific MESSAGE FORMAT field is set to 00 (fixed format), all of the tokens for which token IDs are specified in this TKN record are sent in the message, in the order that the token IDs appear on the screen. This means that all tokens defined in COBTKN at the time the TKN record was last updated are sent in the message. If a token that is specified in the TKN record is not present in the internal message, the ISO Host Interface process creates an *empty* token for the maximum size of the token, and sends the empty token. To create the empty token, the ISO Host Interface process pads the token with spaces (alphanumeric fields) or zeros (numeric fields).

   If the product-specific MESSAGE FORMAT field is set to 01 (variable format), all tokens in the internal message are sent in the external message. The tokens are sent in the same order in which the tokens appear in the internal message.

8. Press the **F3** key to add the record. If the ORDER FLAG field contains the value N, no further steps are required. If the ORDER FLAG field contains the value Y, continue with steps 9 through 12.

9. If the screen contained 24 tokens and more tokens exist, a message is displayed at the bottom of the screen indicating that there are more tokens. To display the next page of tokens, press the **Shift-F6** keys.

10. For each token on the screen, determine whether the token should be sent in the external message. If the token should not be sent, no changes are necessary. If the token should be sent, use the **Tab** key to move to the SEND ORDER field for the token and enter a value ranging from 1 to the number of tokens present in the record (maximum of 360). This number indicates the order in which the token is added to the external message, if the ORDER FLAG field is set to Y.
11. Repeat steps 9 and 10 until all tokens have been checked.
12. Press the **F5** key to update the record.

# 5: BASE24 Base Tokens

This section describes the BASE24 Base message tokens. Tokens are described in alphanumeric order, according to token ID. The Header token and token header, which do not have token IDs, are described first. The table below identifies the BASE24 Base tokens and their corresponding token IDs. For tokens with ASCII formats, the ASCII formats follow the corresponding binary format.

| ID | Token Name |
|---|---|
| N/A | Header token |
| N/A | Token header |
| 08 | Customer Name token[1] |
| 12 | MICR Data token[2] |
| 13 | Credit Line token[1] |
| 18 | Account Qualifier token |
| 23 | Track 1 token |
| 25 | Surcharge Data token |
| 27 | Cardholder Postal Code token |
| 28 | ACI Proactive Risk Manager token |
| 30 | Issuer Fee Rebate token |
| 32 | PRM Real Time token |
| B0 | Switch token (Acquirer) |
| B1 | Switch token (Issuer) |
| B2 | EMV Request Data token |
| B3 | EMV Discretionary Data token |
| B4 | EMV Status token |
| B5 | EMV Response Data token |

| ID | Token Name |
|---|---|
| B6 | EMV Script Data token |
| B7 | TLF token |
| B8 | Transaction Profile token |
| B9 | Transaction Description token |
| BA | Acquirer Routing token |
| BB | Pre-Pay Generic Receipt token |
| BC | TSS Index token |
| BD | Multiple Currency token |
| BE | Original Currency Release 6.0 token |
| BF | Pre-Pay Receipt token |
| BG | Track 3 token |
| BH | Reversal Date and Time token |
| BI | Pre-Pay Top-Up token |
| BJ | EMV Issuer Script Results token |
| BK | Multiple Logical Network token |
| BL | Virtual Primary Account Number token |
| BM | Transaction Subtype token |
| BN | Data Encryption Key token |
| BO | Encrypted Balance token |
| BP | Person-to-Person Transaction token |
| BQ | Completion Required token |
| BR | Split Transaction Routing token |
| BS | Pre-Pay Switch token |

| ID | Token Name |
|---|---|
| BT | Pre-Pay Response token |
| BU | Pre-Pay Selection token |
| BV | Pre-Pay Voucher Receipt token |
| BW | Pre-Pay Online Receipt token |
| BX | Pre-Pay Original Data token |
| BY | Switch Common Data token |
| M1 | Migration ATM Data1 token |
| M2 | Migration POS Data1 token |
| M4 | Migration EPS HISO token |
| M5 | Migration Customer Data token |
| N8 | Inventory Voucher token |
| S0 | Intra Country Data token |
| S1 | Gateway Info token |
| S2 | Dynamic Currency Conversion (DCC) Status token |
| S3 | Dynamic Currency Conversion (DCC) Processing token |
| S4 | EMV Supplementary Data token |
| S6 | Track 2 token |
| S7 | Person-to-Person Transaction 2 token |
| S8 | PAN Mapping token |
| S9 | Additional Authorization Data token |
| SA | Generic Data token |

[1] This token is currently used by the BASE24-teller product only.

[2] This token is currently used by the BASE24-atm product only.

# Header Token—Binary Format

The fields in the binary format Header token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–6 | | HEADER-TKN | |
| 1 | 02 | EYE-CATCHER | PIC X(1) |
| | | Indicates the start of token data. The only valid value is an ampersand (&). | |
| 2 | 02 | USER-FLD1 | PIC X(1) |
| 3–4 | 02 | CNT | TYPE BINARY 16 |
| | | The count of the number of tokens, including the Header token, that are present in the token data buffer. | |
| 5–6 | 02 | LGTH | TYPE BINARY 16 |
| | | The length of all token data, including the Header token and token header structures, present in a token data buffer. | |

# Header Token—ASCII Format

The fields in the ASCII format Header token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–12 | | HEADER-TKNX | |
| 1 | 02 | EYE-CATCHER | PIC X(1) |
| 2 | 02 | USER-FLD1 | PIC X(1) |
| 3–7 | 02 | CNT | PIC 9(5) |
| 8–12 | 02 | LGTH | PIC 9(5) |

# Token Header—Binary Format

The fields in the binary format token header are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–6 | | TKN-HEADER | |
| 1 | 02 | EYE-CATCHER | PIC X(1) |
| | | Indicates the start of an individual token. The only valid value is an exclamation point (!).<br>**Note:** If the Super Extract process converts a token to EBCDIC, the exclamation point in this field is translated to a vertical bar (\|). | |
| 2 | 02 | USER-FLD1 | PIC X(1) |
| 3–4 | 02 | TKN-ID | PIC X(2) |
| | | The two-byte ASCII representation of the token ID. The token ID uniquely identifies a token. | |
| 5–6 | 02 | LGTH | TYPE BINARY 16 |
| | | The length of the token data for the token identified by the TKN-ID field. | |

# Token Header—ASCII Format

The fields in the ASCII format token header are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–10 | | TKN-HEADERX | |
| 1 | 02 | EYE-CATCHER | PIC X(1) |
| 2 | 02 | USER-FLD1 | PIC X(1) |
| 3–4 | 02 | TKN-ID | PIC X(2) |
| 5–9 | 02 | LGTH | PIC 9(5) |
| 10 | 02 | USER-FLD2 | PIC X(1) |

# Token 08 Customer Name Token—Binary Format

The fields in the binary format Customer Name token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–128 | | NAM-TKN | |
| 1–2 | 02 | ACCT-CNT<br>The number of occurrences of the customer name information.<br>The information can be available for a maximum of three accounts. | TYPE BINARY 16 |
| 3–128 | 02 | ACCT<br>The customer name information. Customer name information can be provided for a maximum of three different accounts, depending on the value in the ACCT-CNT field.<br>The length of each ACCT occurrence is 42 positions. | OCCURS 0 TO 3 TIMES DEPENDING ON ACCT-CNT |
| | 04 | IND<br>A code indicating the account number with which this information is associated. Valid values are as follows:<br>C = Credit line/backup account<br>F = *From* account<br>T = *To* account | PIC X(1) |
| | 04 | CUST-SHORT-NAM<br>The customer name from the Customer Short Name segment of the PBF. | PIC X(40) |
| | 04 | USER-FLD1 | PIC X(1) |

# Token 08 Customer Name Token—ASCII Format

The fields in the ASCII format Customer Name token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–128 | | NAM-TKNX | |
| 1–2 | 02 | ACCT-CNT | PIC 9(2) |
| 3–128 | 02 | ACCT | OCCURS 0 TO 3 TIMES DEPENDING ON ACCT-CNT |
| | 04 | IND | PIC X(1) |
| | 04 | CUST-SHORT-NAM | PIC X(40) |
| | 04 | USER-FLD1 | PIC X(1) |

# Token 12 MICR Data Token

The fields in the MICR Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–64 | | MICR-DATA-TKN | |
| 1–64 | 02 | MICR-DATA<br>The Magnetic Ink Character Recognition (MICR) data.<br>MICR data is the string of characters at the bottom of most checks. | PIC X(64) |

# Token 13 Credit Line Token—Binary Format

The fields in the binary format Credit Line token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–30 | | CR-LINE-TKN | |
| 1–2 | 02 | ACCT-TYP<br>The credit line or backup account type, if a credit line or backup account is involved in the transaction. Valid values are as follows:<br>01–09 = DDA (checking)<br>11, 14–19 = Savings<br>21 = NOW (interest-bearing checking)<br>31, 33–39 = Credit card<br>32 = Credit line | PIC 9(2) |
| 3–21 | 02 | ACCT<br>The credit line or backup account number associated with the primary account, if the account is accessed in order to authorize the transaction. | |
| 3–21 | 04 | ACCT-NUM | PIC X(19) |
| 22 | 02 | USER-FLD | PIC X(1) |
| 23–30 | 02 | XFER-AMT<br>The credit line transfer amount.<br>The amount that was transferred from the credit line or backup account to the primary account in order to authorize the transaction. | TYPE BINARY 64 SIGNED |

# Token 13 Credit Line Token—ASCII Format

The fields in the ASCII format Credit Line token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–42 | | CR-LINE-TKNX | |
| 1–2 | 02 | ACCT-TYP | PIC 9(2) |
| 3–21 | 02 | ACCT | |
| 3–21 | 04 | ACCT-NUM | PIC X(19) |
| 22–23 | 02 | USER-FLD | PIC X(2) |
| 24–42 | 02 | XFER-AMT | PIC X(19) |

# Token 18 Account Qualifier Token

The fields in the Account Qualifier token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–6 | | ACCT-QUAL-TKN | |
| 1–3 | 02 | FROM-ACCT<br>The account qualifier data for the *from* account. | |
| 1–2 | 04 | TYP<br>The account type associated with the *from* account. | PIC X(2) |
| 3 | 04 | IDX<br>An index value indicating the specific account for the account type specified in the FROM-ACCT.TYP field. | PIC X(1) |
| 4–6 | 02 | TO-ACCT<br>The account qualifier data for the *to* account. | |
| 4–5 | 04 | TYP<br>The account type associated with the *to* account. | PIC X(2) |
| 6 | 04 | IDX<br>An index value indicating the specific account for the account type specified in the TO-ACCT.TYP field. | PIC X(1) |

# Token 23 Track 1 Token

The fields in the Track 1 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–82 | 02 | VAL | PIC X(82) |

The Track 1 data from the card. This field is variable length up to 82 characters and includes the start and end sentinels. This is the ISO definition. The BIC ISO Interface process and the ISO Host Interface process use the field in the external message to pass Track 1 data. Track 1 data is defined as follows:

| Subfield | Description | Length |
|---|---|---|
| STX | Start sentinel | 1 character (%) |
| FC | Format code | 1 character |
| PAN | Identification | 1–19 characters |
| FS | Field separator | 1 character (^) |
| CC | Country code (only present when the PAN starts with 59) | 3 characters |
| NM | Name | 1–26 characters |
| FS | Field separator | 1 character (^) |
| ED | Expiration date | 4 characters (YYMM) |
| SC | Service code | 3 characters |
| DD | Discretionary data | 1–21 characters |
| ES | End sentinel | 1 character (?) |
| LRC | Longitudinal redundancy check (optional) | 1 character |

# Token 25 Surcharge Data Token—Binary Format

The fields in the binary format Surcharge Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–34 | | SURCHARGE-DATA-TKN<br>This token is used to store surcharge assessment information defined in the BASE24-atm Terminal Data files and Surcharge File (SURF) and other information filled in at the entry point and by the Authorization module.<br>This token is also used to store a transaction fee (e.g., surcharge, access fee, convenience fee) received in request messages from a Host or Interface, for both ATM and POS transactions.<br>All amounts in the token are specified in the currency defined in STM.RQST.ORIG^CRNCY^CDE. | |
| 1–8 | 04 | TRAN-FEE<br>The transaction surcharge amount assessed. If the amount in this field is a negative amount, it must be preceded by a minus sign (–). This field is filled in by the Authorization module (acquirer traffic) or Interface process (issuer traffic). This field is updated on reversals by the Device Handler process (acquirer traffic) or the Interface process (issuer traffic). | TYPE BINARY 64 SIGNED |
| 9–16 | 04 | ORIG-FEE<br>The original transaction surcharge assessed. If the amount in this field is a negative amount, it must be preceded by a minus sign (–). This field is filled in by the Authorization module (acquirer traffic) or the Interface process (issuer traffic). | TYPE BINARY 64 SIGNED |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 17–20 | 04 | TERM-SUR-PROFILE | PIC X(4) |
| | | The surcharge profile assigned to the terminal. This field corresponds to the ATDS1.TERM-SUR-PROFILE or TDF. TERM-SUR-PROFILE field. This field is filled in by the Device Handler process and is used by the Authorization module. | |
| 21 | 04 | RVSL-CDE | PIC X(1) |
| | | A code specifying the surcharge requirements for partial reversals. This field is filled in by the Authorization module and is used by the Device Handler process. Valid values are as follows:<br>0 = No fee on partial reversals<br>1 = Fee on partial reversals | |
| 22 | 04 | USER-FLD1 | PIC X(1) |
| 23–30 | 04 | FLAT-FEE | TYPE BINARY 64 SIGNED |
| | | The static surcharge amount. If the amount in this field is a negative amount, it must be preceded by a minus sign (–). This field is filled in by the Authorization module and is used by the Device Handler process for reversal processing. | |
| 31–32 | 04 | PCNT-FEE | TYPE BINARY 16 SIGNED |
| | | The surcharge percentage in one hundredths of a percent (for example 100 = 1%). If the value in this field is a negative percentage, it must be preceded by a minus sign (–). This field is filled in by the Authorization process and used by the Device Handler process for reversal processing. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 33 | 04 | MIN-MAX | PIC X(1) |

An indicator specifying the interaction between the FLAT-FEE and PCNT-FEE fields. Valid values are as follows:

0 = The surcharge is the greater amount of the flat fee and the percent fee
1 = The surcharge is the lesser amount of the flat fee and the percent fee

This field is filled in by the Authorization process and is used by the Device Handler process for reversal processing.

| | | | |
|---|---|---|---|
| 34 | 04 | AUTH-IND | PIC X(1) |

A code specifying the surcharge notification process required by the ATM and Device Handler process. This value is provided in the STM 0200 request by the Device Handler process. If set to the value 1, the authorization process routes the transaction with the surcharge for authorization without notifying the cardholder first. This value is updated by the Device Handler process with the value D when the cardholder is notified of the surcharge and does not accept the surcharge fee. This value is updated by the Device Handler process with the value M when the actual fee calculation exceeds the maximum surcharge fee approved by the cardholder. This value is updated by the Authorization process or the Host Interface Process with the value Z when the fee is loaded into the ORIG^FEE field. Valid values are as follows:

b = Request notification (where b is a blank character)
0 = Request notification
1 = Response notification
D = Fee assessment/notification is complete; surcharge fee declined
M = Misconfiguration
Z = Fee assessment/notification is complete

# Token 25 Surcharge Data Token—ASCII Format

The fields in the ASCII format Surcharge Data token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–70 | | SURCHARGE-DATA-TKNX | |
| 1–19 | 04 | TRAN-FEE | PIC X(19) |
| 20–38 | 04 | ORIG-FEE | PIC X(19) |
| 39–42 | 04 | TERM-SUR-PROFILE | PIC X(4) |
| 43 | 04 | RVSL-CDE | PIC X(1) |
| 44–62 | 04 | FLAT-FEE | PIC X(19) |
| 63–67 | 04 | PCNT-FEE | PIC X(5) |
| 68 | 04 | MIN-MAX | PIC X(1) |
| 69 | 04 | AUTH-IND | PIC X(1) |
| 70 | 04 | USER-FLD1 | PIC X(1) |

# Token 27 Cardholder Postal Code Token

The fields in the Cardholder Postal Code token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–10 | | CRD-POSTAL-CDE-TKN | |
| 1–10 | 02 | POSTAL-CDE<br>The postal code for the cardholder. The Authorization process retrieves this information from the address verification segment of the Cardholder Authorization File (CAF) record, if present, and appends the token to the STM or PSTM before sending it to the Fraud Control process. | PIC X(10) |

# Token 28 ACI Proactive Risk Manager Token

The ACI Proactive Risk Manager token contains data used as input to the Scoring Engine Input process. The fields in the ACI Proactive Risk Manager Code token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–88 | | PRISM-TKN<br>The ACI Proactive Risk Manager token is used to pass information from the CAF, CPF, and PBF to the Scoring Engine Input process. The token is added by the Authorization process or module. | |
| 1–4 | 02 | EXP-DAT<br>The card expiration date. | PIC X(4) |
| 5–16 | 02 | CUR-AUTH-AMT<br>The total amount of cash advanced against credit accounts using the BASE24-atm product or the total amount of purchases and cash advances made against credit accounts using the BASE24-pos product during the current usage accumulation period. | PIC X(12) |
| 17–28 | 02 | PEND-AUTHS<br>The number of times the card has been used by the BASE24-atm and BASE24-pos products during the current usage accumulation period. | PIC X(12) |
| 29–30 | 02 | PREFIX-LGTH<br>The length of the card prefix. The maximum value that can be placed in this field is 11. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 31-42 | 02 | OPEN-TO-BUY | PIC X(12) |
| | | The open-to-buy amount is the amount of available credit for credit accounts and the available balance for debit accounts at the time of authorization. | |
| 43–54 | 02 | AUTH-LMT | PIC X(12) |
| | | The credit limit for credit accounts. | |
| 55–66 | 02 | AUTH-BAL | PIC X(12) |
| | | The current credit balance for credit accounts. The amount in this field is the total amount of charges on the account.<br>Purchases and cash advances made from credit accounts are added to this amount. The amount of refunds for purchases from a credit account is subtracted from this amount. | |
| 67–72 | 02 | LAST-PMNT-DAT | [DAT] |
| | | The date (YYMMDD) of the last payment to the account. | |
| 67–68 | 04 | YY | PIC X(2) |
| 69–70 | 04 | MM | PIC X(2) |
| 71–72 | 04 | DD | PIC X(2) |
| 73–84 | 02 | LAST-PMNT-AMT | PIC X(12) |
| | | The amount of the last payment to the account. | |
| 85 | 02 | ACCT-STAT | PIC X(1) |
| | | The current status of the application account. For valid values, please refer to the following field in the PBF: PBFBASE. ACCT-STAT. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 86 | 02 | CRD-STAT<br>The card status. For valid values, please refer to the following field in the CAF: CAFBASE.CRD-STAT. | PIC X(1) |
| 87 | 02 | AUTH-LMT-IND<br>Indicates whether this token contains limit data from the CPF, CAF only, CAF/PBF, NEG only, NEG/UAF, or from the host. Valid values are as follows:<br>0 = CPF data only<br>C = CAF data included<br>P = CAF and PBF data included<br>N = NEG only<br>U = NEG/UAF<br>H = Host | PIC X(1) |
| 88 | 02 | USER-FLD1<br>This field ensures word alignment. | PIC X(1) |

# Token 30 Issuer Fee Rebate Token—Binary Format

The Issuer Fee Rebate token contains an issuer fee or rebate assessed by the transaction issuer. The fields in the binary format of the Issuer Fee Rebate token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–30 | | ISSUER-FEE-REBATE-TKN | |
| 1–8 | 02 | TRAN-FEE | TYPE BINARY 64 SIGNED |
| | | The transaction issuer fee or rebate amount assessed. If the amount in this field is a negative amount, it must be preceded by a minus sign (–). This field is set by the Authorization module for acquirer traffic or by the Interchange Interface process for issuer traffic. This field may be updated on reversals by the Interchange Interface process for issuer traffic. | |
| 9–16 | 02 | ORIG-FEE | TYPE BINARY 64 SIGNED |
| | | The original transaction issuer amount assessed. If the amount in this field is a negative amount, it must be preceded by a minus sign (–). This field is set by the Authorization module for acquirer traffic or by the Interchange Interface process for issuer traffic.<br>This field is not currently used. | |
| 17 | 02 | RVSL-CDE | PIC X(1) |
| | | This field indicates the issuer fee or rebate requirements for partial reversals of cash dispenses. This field is set by the Authorization module and used by the Device Handler module. Valid values are as follows:<br>0 = No fee on partial reversals.<br>1 = Fee on partial reversals.<br>This field is not currently used. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 18 | 02 | USER-FLD1<br>This field ensures word alignment. | PIC X(1) |
| 19–26 | 02 | FLAT-FEE<br>The static issuer fee or rebate amount, in the currency defined in the RQST.ORIG-CRNCY-CDE field of the BASE24-atm Standard Internal Message (STM). For U.S. currency, it is expressed in dollars and cents. If the amount in this field is a negative amount, it must be preceded by a minus sign (–). This field is set by the Authorization module and is used by the Device Handler module for reversal processing.<br>This field is not currently used. | TYPE BINARY 64 SIGNED |
| 27–28 | 02 | PCNT-FEE<br>The issuer fee or rebate percentage, expressed in hundredths of a percent (e.g., 100 = 1%). If the value in this field is a negative percentage, it must be preceded by a minus sign (–). This field is set by the Authorization module and is used by the Device Handler module for reversal processing.<br>This field is not currently used. | TYPE BINARY 16 SIGNED |
| 29 | 02 | MIN-MAX<br>A code indicating the interaction between the FLAT-FEE and the PCNT-FEE fields. The issuer fee or rebate amount, FLAT-FEE, or (PCNT-FEE ×transaction amount), is selected based on whichever amount is greater. When this field contains a value of 1, the transaction issuer fee or rebate amount is selected based on whichever amount is less. This field is set by the Authorization module and is used by the Device Handler module for reversal processing.<br>This field is not currently used. | PIC X(1) |
| 30 | 02 | AUTH-IND<br>This field is not currently used. | PIC X(1) |

# Token 30 Issuer Fee Rebate Token—ASCII Format

The fields in the ASCII format Issuer Fee Rebate token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–66 | | ISSUER-FEE-REBATE-TKNX | |
| 1–19 | 02 | TRAN-FEE | PIC X(19) |
| 20–38 | 02 | ORIG-FEE | PIC X(19) |
| 39 | 02 | RVSL-CDE | PIC X(1) |
| 40 | 02 | USER-FLD1 | PIC X(1) |
| 41–59 | 02 | FLAT-FEE | PIC X(19) |
| 60–64 | 02 | PCNT-FEE | PIC X(5) |
| 65 | 02 | MIN-MAX | PIC X(1) |
| 66 | 02 | AUTH-IND | PIC X(1) |

# Token 32 PRM Real Time Token

The PRM Real Time token is used to pass and store information for the real time processing features of ACI Proactive Risk Manager. The token is initially added by the Authorization system (ATM or POS).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–12 | | PRM-RT-TKN | |
| 1 | 02 | REAL-TIME-REQ-TYPE | PIC X(1) |

The type of Real Time Request. Valid values are as follows:

␣ = Allow interface process to set and control this token field (where ␣ indicates a blank space)
0 = Real Time Score only
1 = Real Time Rules only
2 = Both Real Time Score and Real Time Rules

| | | | |
|---|---|---|---|
| 2 | 02 | PIP-RESPONSE-REQUEST | PIC X(1) |

The type of message that should be returned to the Authorization system. Valid values are as follows:

␣ = Allow interface process to set and control this token field (where ␣ indicates a blank space)
0 = Response to Authorization is a 0200 message
1 = Response to Authorization is a 0210 message

| | | | |
|---|---|---|---|
| 3 | 02 | REAL-TIME-STATUS | PIC X(1) |

A processing control field to assist with the type of processing required. Valid values are as follows:

␣ = Allows interface process to set and control this token field (where ␣ indicates a blank space)
0 = Initial state. Transaction not yet processed in real time.
1 = Transaction processed by PSE in real time
2 = Transaction processed by RTR in real time
3 = Transaction processed by both PSE and RTR in real time

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 4–6 | 02 | REAL-TIME-SCORE<br>The real time fraud score generated by the PRM Scoring Engine (PSE). | PIC 9(3) |
| 7–10 | 02 | RTR-RULE-FIRED<br>The real time rule number fired during real time rule processing. | PIC 9(4) |
| 11 | 02 | RTR-DISPOSITION<br>The PRM-recommended disposition of the real time transaction. Valid values are as follows:<br>A = Authorization recommended<br>D = Decline recommended<br>R = Referral recommended | PIC X(1) |
| 12 | 02 | RTR-FILLER<br>This field reserved for future use. | PIC X(1) |
| 12 | 02 | RTS-DISPOSITION<br>The value in this field is used to populate the real time score disposition. | REDEFINES RTR-FILLER |

# Token B0 Switch Token (Acquirer) and Token B1 Switch Token (Issuer)—Binary Format

The token ID for the Switch Token is B0 for Acquirer and B1 for Issuer. Both tokens use the same DDL structure; therefore these tokens are documented together.

Listed in this description is the data that is carried in the token. Because the data carried in the token varies by interchange, the information is documented for each interchange individually.

The fields in the binary format Switch token for acquirers (token ID B0) and issuers (token ID B1) are described below.

**Note:** The B0 and B1 tokens are specific to individual switch interfaces. Other BASE24 processes do not access the information in these tokens other than for transmission to the host or co-network and for logging to the TLF or PTLF.

## Interface FIIDs and Version IDs

Each interface has a unique FIID that identifies it to the BASE24 system. This value is hard-coded in the FIID field of the Switch token and is not checked against the value in the INTERFACE FIID field on screen 1 of the Interchange Configuration File (ICF) or the Enhanced Interchange Configuration File (ICFE). FIID values less than 4 characters in length are left-justified.

Also, each interface has a version ID assigned to it that identifies the current version of the interface. This value is carried in the VER or VER-ID field of the Switch token.

The IDs for each interface B0 and B1 token format documented in this manual are provided in the following table so that readers can look up the values in one place.

| Interchange Interface | FIID | Version ID |
|---|---|---|
| Alaska Option ISO | AOI | 02 |
| American Express CAPN ISO (AXCI) | AXCI | 01 |
| American Express Global Network (GNS) | AEGN | 02 |
| Banknet ISO | BNET | 16 |

| Interchange Interface | FIID | Version ID |
|---|---|---|
| BIC ANSI | Does not use B0/B1 | |
| BIC ISO | BICI | 01 |
| Cash Station ISO | CSSI | 02 |
| Deluxe ISO | DIGI | 01 |
| DIAS | DIAS | 02 |
| Discover ISO | DSCV | 05 |
| EPS-Net | EURO | 03 |
| FDR ISO | Does not use B0/B1 | |
| JCB ISO (JCBI) | Does not use B0/B1 | |
| LINK (LIS5) | LINK | 07 |
| MAC MASM (MACI) | MACI | 06 |
| MDS Cirrus ISO | MDS | 17 |
| Money Station | MONY | 02 |
| MPS | Does not use B0/B1 | |
| NBGC | NBGC | 01 |
| NPC ISO | NPCI | 01 |
| NYCE ISO | NYCI | 07 |
| Networks ISO | NETI | 01 |
| Plus ISO | PISO | 06 |
| Pulse ISO | PULI | 05 |
| SPAN2 | SPAN | 02 |
| Star ISO | STRI | 14 |
| SVS | SVS | 01 |

| Interchange Interface | FIID | Version ID |
|---|---|---|
| Shazam ISO | SHZM | 02 |
| ValueLink | VLNK | None |
| Visa Debit Processing Service (DPS) | VDPS | 07 |
| VisaNet ISO | VISA | 22 |

## Alaska Option ISO

The following fields define the values in the B0 and B1 tokens that are used by the Alaska Option ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–38 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–38 | 02 | AOI<br>Alaska Option ISO specific data used in the generic switch token. | REDEFINES BUF |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–12 | 04 | NETWRK-ID<br>The external message from the interchange will contain the network ID needed to fill this field. | PIC X(4) |
| 13 | 04 | ON-PREMISE | PIC X(1) |
| 14 | 04 | BILL-PAY<br>This field contains one of the following values to indicate the billpay status of the transaction:<br>b = Not a billpay transaction<br>1 = ATM billpay transaction<br>2 = POS billpay transaction | PIC X(1) |
| 15–24 | 04 | SWI-TXN-DAT-TIM<br>The external switch date and time. | PIC X(10) |
| 25–27 | 04 | SWI-RESP-CDE<br>The external switch response code. | PIC X(3) |
| 28–29 | 04 | ACQ-FROM-ACCT<br>The external message from the acquirer contains the “from” account type needed to fill this field. | PIC X(2) |
| 30–31 | 04 | ACQ-TO-ACCT<br>The external message from the acquirer contains the “to” account type needed to fill this field. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 32–34 | 04 | FROM-ACCT-QUAL | PIC X(3) |
| | | This field supports credit loan suffixes, money market, and IRA accounts. These values are received from members and passed to members as an account type19 along with bit 120 turned on and populated with the appropriate account qualifier value of MM or IRA. These are passed within BASE24 as account type 11. | |
| 35–37 | 04 | TO-ACCT-QUAL | PIC X(3) |
| | | This field supports credit loan suffixes, money market, and IRA accounts. These values are received from members and passed to members as an account type19 along with bit 120 turned on and populated with the appropriate account qualifier value of MM or IRA. These are passed within BASE24 as account type 11. | |
| 38 | 04 | FULL-TRK2-DATA-PRSN | PIC X(1) |
| | | If data element 35 is present in the external request/advice message sent inbound to the interface, FULL-TRK2-DATA-PRSN will contain value Y. Otherwise, it will be blank. | |

## American Express CAPN ISO (AXCI)

The following fields define the values in the B0 and B1 tokens that are used by the American Express CAPN ISO (AXCI) interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–54 | | SWI-TKN | |
| 1–2 | 02 | LGTH | TYPE BINARY 16 SIGNED |
| | | The length of the token data. The length includes the FIID and the data present in the data buffer. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–54 | 02 | AXCI-SWI-TKN<br>AXCI ISO specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The switch token version ID. | PIC X(2) |
| 9–14 | 04 | APPRV-CDE<br>The approval code. | PIC X(6) |
| 15–17 | 04 | ACT-CDE<br>The action code. | PIC X(3) |
| 18–42 | 04 | ADNL-RESP-DATA<br>The additional response data. | PIC X(25) |
| 43–54 | 04 | PT-SVC-DATA-CDE | PIC X(12) |

# American Express Global Network (GNS)

The following fields define the values in the B0 and B1 tokens that are used by the American Express Global Network interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–56 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–56 | 02 | AEGN-SWI-TKN-DATA<br>American Express Global Network specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER-ID<br>The switch token version ID. | PIC X(2) |
| 9–20 | 04 | PT-SVC-DATA-CDE<br>The point-of-service data code. | PIC X(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 21–32 | 04 | RETRV-REF-NUM | PIC X(12) |
| | | A number assigned by the message initiator to uniquely identify a transaction. This number remains unchanged for all messages throughout the life of a transaction. | |
| 33–38 | 04 | APPRV-CDE | PIC X(6) |
| | | A code assigned by the authorizing institution to indicate approval. | |
| 39–41 | 04 | ACT-CDE | PIC 9(3) |
| | | A code set by the transaction authorizer to indicate the disposition of the message. | |
| 42–56 | 04 | AEGN-ACQ-REF-DATA | PIC X(15) |
| | | The American Express Global Network acquirer reference data, containing the transaction ID. | |

## BankNet

The following fields define the values in the B0 and B1 tokens that are used by the BankNet interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–234 | | SWI-TKN | |
| 1–2 | 02 | LGTH | TYPE BINARY 16 SIGNED |
| | | The length of the token data. The length includes the FIID and the data present in the data buffer. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–6 | 02 | FIID | PIC X(4) |
| | | The FIID assigned to the interface. The value in this field is hard-coded. | |
| 7–448 | 02 | BUF | PIC X(442) |
| | | The generic data, which has a variable length up to 442 characters. | |
| 7–234 | 02 | BNET-SWI-TKN-DATA | REDEFINES BUF |
| | | BankNet specific data used in the generic switch token. | |
| 7–8 | 04 | VER-ID | PIC X(2) |
| | | The switch token version ID. | |
| 9–14 | 04 | LOCAL-TIM | PIC 9(6) |
| | | The local time in hhmmss format. Message field: DE12. | |
| 15–18 | 04 | LOCAL-DAT | PIC 9(4) |
| | | The local date in MMDD format. Message field: DE13. | |
| 19–21 | 04 | ADVICE-RSN-CDE | PIC 9(3) |
| | | The advice reason code. Message field: DE60 SE1. | |
| 22–24 | 04 | POS-ENTRY-MDE | PIC 9(3) |
| | | The point-of-service entry mode. The field contains two codes. The first code is two digits in length and indicates the method by which Track 2 data or the primary account number (PAN) was entered into the system. The second code is one digit in length and indicates the entry capabilities available at the point of service. Message field: DE22. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 25–26 | 04 | RESP-CDE<br>The response code. Message field: DE39. | PIC 9(2) |
| 27 | 04 | CRD-VRFY-RESULT<br>The card verification result. Message field: DE48 SE87. | PIC X(1) |
| 28 | 04 | VISA-XSTNG-DBT-IND<br>The Visa existing debt indicator. Message field: DE48 SE85. | PIC X(1) |
| 29 | 04 | VISA-SVC-DVLPMT-IND<br>The Visa service development indicator. Message field: DE48 SE86. | PIC X(1) |
| 30 | 04 | VISA-DEF-BILL-IND<br>The Visa deferred billing indicator. Message field: DE48 SE78. | PIC X(1) |
| 31–36 | 04 | PROC-CDE<br>The transaction processing code. Message field: DE3. | PIC X(6) |
| 37–66 | 04 | ON-BEHALF | OCCURS 10 TIMES |
| | 06 | SVC<br>A code containing the Banknet On-Behalf Services data. Message field: DE48 SE71, subfield 1.<br>Valid values are as follows:<br>01 = M/Chip to Magnetic Stripe Conversion<br>02 = M/Chip Cryptogram Pre-Validation/ Post-Generation | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

03 = Dynamic M/Chip Stand-In
05 = MasterCard SecureCode AAV Verification Service
06 = MasterCard SecureCode Dynamic AAV Verification in Stand-In

06 RSLT-1 PIC X(1)

A code indicating the Banknet On-Behalf Result 1 value. Message field: DE48 SE71, subfield 2.

Valid values are as follows:

C = Conversion of M/Chip transaction to a magnetic stripe transaction completed.
F = Format error. No check on Cryptogram, status of TVR/CVR unknown.
G = Application Cryptogram is valid but not an ARQC. Status of TVR/CVR unknown.
I = Invalid. Application Cryptogram is incorrect. Status of TVR/CVR unknown.
T = Valid ARQC, TVR/CVR invalid.
U = Unable to process. No check of Cryptogram. Status of TVR/CVR unknown.
V = Valid ARQC, valid TVR/CVR.
b = No value present (where b indicates a blank space).

67–68 04 MCHIP-PRO-IND PIC X(2)

A code containing M/Chip processing information. This code gives acquirers more information about cryptogram validation. Message field: DE48 SE74, subfield 1.

Valid values are as follows:

02 = MasterCard On-Behalf Service – M/Chip Cryptogram pre-validation
03 = MasterCard On-Behalf Service – M/Chip Cryptogram validation in stand-in processing
50 = Issuer chip validation

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 69 | 04 | MCHIP-PROC-INFO<br>A code containing M/Chip processing information. This code gives acquirers more information about cryptogram validation. Message field: DE48 SE74, subfield 2.<br>Valid values are as follows:<br>i = Application Cryptogram invalid<br>U = Application Cryptogram not validated due to technical error<br>F = Format error in DE 55<br>G = Cryptogram in application is valid but is not an ARQC<br>T = Application Cryptogram is valid but TVR/CVR is invalid<br>X = Issuer provided incorrect value in subfield 2 | PIC X(1) |
| 70-78 | 04 | TRAN-FEE<br>The transaction fee. Message field: DE28. | PIC X(9) |
| 79-81 | 04 | E-COM-SEC-LVL-IND<br>E-commerce security level indicator. Message field: DE48 SE42, subfield 1. | PIC 9(3) |
| 82 | 04 | AVS-RESULT<br>Message field: DE48 SE83. | PIC X(1) |
| 83 | 04 | ACCT-NUM-IND<br>Account number indicator. Message field: DE48 SE33, subfield 1. | PIC X(1) |
| 84–87 | 04 | ADVC-DETL-CDE<br>Message field: DE60 SE2. | PIC 9(4) |
| 88–93 | 04 | AUTH-AGENT-ID-CDE | PIC X(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | Message field: DE121. | |
| 94–96 | 04 | PMNT-TXN-TYP-IND | PIC X(3) |
| | | Message field: DE48 SE77. | |
| 97–146 | 04 | CHIP-BIT-ERR-RSLTS | OCCURS 10 TIMES |
| | | Message field: DE48 SE79. | |
| | 06 | CVR-TVR-ID | PIC X(1) |
| | 06 | BYTE-ID | PIC X(2) |
| | 06 | BIT-ID | PIC X(1) |
| | 06 | BIT-ERR | PIC X(1) |
| 147–148 | 04 | CRD-LVL-RSLT | PIC X(2) |
| | | Message field: DE48 SE46. | |
| 149–156 | 04 | MC-GTWY-TXN-IND | PIC X(8) |
| | | Message field: DE48 SE47. | |
| 157–162 | 04 | MC-ASGN-ID-SUB-FLD-32 | PIC X(6) |
| | | Message field: DE48 SE32. | |
| 163–181 | 04 | VPAN | PIC X(19) |
| | | Message field: DE48 SE33 subfield 2. | |
| 163–181 | 04 | PAYPASS-MAPPED-PAN | REDEFINES VPAN. |
| 163–181 | 04 | PAYPASS-PAN | REDEFINES VPAN. |
| 182 | 04 | RTE-IND | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | Message field: DE48 SE12. | |
| 183 | 04 | VISA-MKT-SPCFC-DATA | PIC X(1) |
| | | Message field: DE48 SE96. | |
| 184–194 | 04 | NTL-POS-DATA | |
| | 06 | COND-CDE | PIC 9(11) |
| | | Message field: DE61 SE1 through SE11. | |
| 195–196 | 04 | PIN-SVC-CDE | PIC 9(2) |
| | | Message field: DE48 SE80. | |
| 197–198 | 04 | PIN-CAP-CDE | PIC 9(2) |
| | | Message field: DE26. | |
| 199–208 | 04 | AUTH-SYS-ADV-DAT-TIM | PIC 9(10) |
| | | Message field: DE48 SE15. | |
| 209–220 | 04 | FRAUD-DATA | |
| | 06 | SCORE | PIC X(3) |
| | | Message field: DE48 SE75 subfield 1. | |
| | 06 | RSN-CDE | PIC X(2) |
| | | Message field: DE48 SE75 subfield 2. | |
| | 06 | RULE-ADJ-SCORE | PIC X(3) |
| | | Message field: DE48 SE75 subfield 3. | |
| | 06 | RULE_RSN-CDE-1 | PIC X(2) |
| | | Message field: DE48 SE75 subfield 4. | |
| | 06 | RULE_RSN-CDE-2 | PIC X(2) |
| | | Message field: DE48 SE75 subfield 5. | |
| 221–222 | 04 | PMNT-INITIATION-CHAN | PIC X(2) |
| | | Message field: DE48 SE23. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 223–234 | 04 | GRATUITY-AMT | PIC 9(12) |
| | | Message field: DE54 amount type 44. | |

## BIC ANSI

The BIC ANSI interface does not use the B0 and B1 token.

## BIC ISO

The following fields define the values in the B0 and B1 tokens that are used by the BIC ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–26 | | SWI-TKN | |
| 1–2 | 02 | LGTH | TYPE BINARY 16 SIGNED |
| | | The length of the token data. The length includes the FIID and the data present in the data buffer. | |
| 3–6 | 02 | FIID | PIC X(4) |
| | | The FIID assigned to the interface. The value in this field is hard-coded. | |
| 7–448 | 02 | BUF | PIC X(442) |
| | | The generic data, which has a variable length up to 442 characters. | |
| 7–26 | 02 | BICI-SWI-TKN-DATA | REDEFINES BUF |
| | | BIC ISO specific data used in the generic switch token. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–18 | 04 | SWI-TXN-DAT-TIM<br>The transaction date and time. | PIC X(10) |
| 19–24 | 04 | TRACE-NUM<br>The system trace audit number. | PIC X(6) |
| 25–26 | 04 | RESP-CDE<br>The response code. | PIC X(2) |

## Cash Station ISO

The following fields define the values in the B0 and B1 tokens that are used by the Cash Station ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–24 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–24 | 02 | CSSI<br>Cash Station specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–14 | 04 | STAN<br>The system trace audit number. | PIC X(6) |
| 15–17 | 04 | ACT-CDE<br>The action code. | PIC X(3) |
| 18–23 | 04 | RSRVD-62-CSSI<br>Bit map position 62 (reserved for private use). | PIC X(6) |
| 24 | 04 | OFF-PREMISE<br>The point of service data code. | PIC X(1) |

# Deluxe ISO

The following fields define the values in the B0 and B1 tokens that are used by the Deluxe ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–30 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–30 | 02 | DIGI<br>Deluxe ISO specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–18 | 04 | SWI-TXN-DAT-TIM<br>The switch transaction date and time. | PIC X(10) |
| 19–24 | 04 | STAN<br>The system trace audit number. | PIC X(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 25–26 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 27–29 | 04 | NETWK-ID<br>Bit map position 63 (DIGI data). | PIC X(3) |
| 30 | 04 | FILLER<br>Blank-filled. | PIC X(1) |

## DIAS

The following fields define the values in the B0 and B1 tokens that are used by the DIAS interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–38 | | TKN-DATA | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–38 | 02 | DIAS<br>Discover ISO specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–18 | 04 | SWI-TXN-DAT-TIM<br>Bit map position 7 (Transmission Date and Time). | PIC X(10) |
| 19–24 | 04 | TRACE-NUM<br>Bit map position 11 (System Trace Audit Number) | PIC X(6) |
| 25–26 | 04 | RESP-CDE<br>Bit map position 39 (Response Code). | PIC X(2) |
| 27–38 | 04 | RETRVL-REF-NUM<br>Bit map position 37 (Retrieval Reference Number) | PIC X(12) |

## Discover ISO

The following fields define the values in the B0 and B1 tokens that are used by the Discover ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–78 | | TKN-DATA | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–6 | 02 | FIID | PIC X(4) |
| | | The FIID assigned to the interface. The value in this field is hard-coded. | |
| 7–448 | 02 | BUF | PIC X(442) |
| | | The generic data, which has a variable length up to 442 characters. | |
| 7–78 | 02 | DSCV-SWI-TKN-DATA | REDEFINES BUF |
| | | Discover ISO specific data used in the generic switch token. | |
| 7–8 | 04 | VER | PIC X(2) |
| | | The Switch token version ID. | |
| 9–18 | 04 | XMIT-DAT-TIM | PIC X(10) |
| | | Tthe transmission date and time. | |
| 19–24 | 04 | STAN | PIC X(6) |
| | | The system trace audit number. | |
| 25–26 | 04 | RESP-CDE | PIC X(2) |
| | | The response code. | |
| 27–39 | 04 | ADNL-RESP-DATA-DSCV | PIC X(13) |
| | | Bit map position 44 (additional response data). | |
| 40–54 | 04 | NETWK-REF-ID | PIC X(15) |
| | | Bit map position 48 (network information). | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 55–60 | 04 | PROC-CDE | PIC X(6) |
| | | Bit map position 3 (processing code). | |
| 61–73 | 04 | POS-SRVC-DATA | |
| | | Bit map position 61 ((point of service data). The following subfields have corresponding subfields in field 61. | |
| 61 | 06 | POS-TERM-ATTND-IND | PIC X(1) |
| 62 | 06 | PARTIAL-APPRV-IND | PIC X(1) |
| 63 | 06 | POS-TERM-LOC-IND | PIC X(1) |
| 64 | 06 | POS-CRDHLDR-PRSN-IND | PIC X(1) |
| 65 | 06 | POS-CRD-PRSN-IND | PIC X(1) |
| 66 | 06 | POS-CRD-CAPTR-CAP-IND | PIC X(1) |
| 67 | 06 | POS-TXN-STAT-IND | PIC X(1) |
| 68 | 06 | POS-TXN-SEC-IND | PIC X(1) |
| 69-70 | 06 | USER-FLD | PIC X(2) |
| 71 | 06 | POS-CRD-TERM-INPUT-CAP-IND | PIC X(1) |
| 72-73 | 06 | POS-AUTH-LIFE-CYC | PIC X(2) |
| 74-75 | 04 | TXN-QUAL | |
| | | Bit map position 124 (transaction qualifier). The following subfields have corresponding subfields in field 124. | |
| 74 | 06 | TRK1-DATA-IND | PIC X(1) |
| 75 | 06 | TRK2-DATA-IND | PIC X(1) |
| 76-78 | 04 | PT-SVC-ENTRY-MDE | PIC X(3) |

# EPS-Net

The following fields define the values in the B0 and B1 tokens that are used by the EPS-Net interface (EURO).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–52 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–52 | 02 | EURO-SWI-TKN-DATA<br>EPS-Net specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER-ID<br>The switch token version ID. | PIC X(2) |
| 9–14 | 04 | STAN<br>The systems trace audit number. | PIC 9(6) |
| 15–17 | 04 | POS-ENTRY-MODE<br>The point of service entry mode. | PIC 9(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 18–28 | 04 | FWD-INST-ID-CDE<br>The forwarding institution ID number. | PIC X(11) |
| 29–40 | 04 | RET-REF-NUM<br>The retrieval reference number. | PIC X(12) |
| 41–42 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 43–46 | 04 | EXP-DAT<br>The card expiration date. | PIC 9(4) |
| 47–52 | 04 | PROC-CDE<br>The processing code. | PIC 9(6) |

## FDR ISO

The FDR ISO interface does not use the B0 and B1 token.

## JCB ISO

The JCB ISO interface does not use the B0 and B1 token.

# LINK (LIS5)

The following fields define the values in the B0 and B1 tokens that are used by the LINK (LIS5) interface (LINK).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–370 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–370 | 02 | LINK-SWI-TKN-DATA<br>LINK (LIS5) specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER-ID<br>The switch token version ID. | PIC X(2) |
| 9–14 | 04 | PROC-CDE<br>The processing code. | PIC 9(6) |
| 15–20 | 04 | STAN<br>The systems trace audit number. | PIC 9(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 21–23 | 04 | POS-ENTRY-MDE<br>The point of service entry mode. | PIC 9(3) |
| 24–25 | 04 | POS-COND-CDE<br>The point of service condition code. | PIC 9(2) |
| 26–36 | 04 | FWD-INST-ID-CDE<br>The forwarding institution ID number. | PIC X(11) |
| 37–38 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 39–40 | 04 | RVSL-RSN<br>The reversal reason code. | PIC X(2) |
| 41 | 04 | CRD-DATA-INPUT-CAP<br>The card data input capability. | PIC X(1) |
| 42 | 04 | CRDHLDR-AUTH-CAP<br>The cardholder authentication capability. | PIC 9(1) |
| 43 | 04 | CRD-CAPTR-CAP<br>The card capture capability. | PIC 9(1) |
| 44 | 04 | OPERATING-ENVMT<br>The operating environment. | PIC 9(1) |
| 45 | 04 | CRDHLDR-PRSN<br>A code indicating if the cardholder was present for the transaction. | PIC 9(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 46 | 04 | CRD-PRSN<br>A code indicating if the card was present for the transaction. | PIC 9(1) |
| 47 | 04 | CRD-DATA-INPUT-MDE<br>The card data input mode. | PIC 9(1) |
| 48 | 04 | CRDHLDR-AUTH-METHOD<br>The cardholder authentication method. | PIC 9(1) |
| 49 | 04 | CRDHLDR-AUTH-ENTY<br>The cardholder authentication entity. | PIC 9(1) |
| 50 | 04 | CRD-DATA-OUTPUT-CAP<br>The card data output capability. | PIC 9(1) |
| 51 | 04 | TERM-OUTPUT-CAP<br>The terminal output capability. | PIC 9(1) |
| 52 | 04 | PIN-CAPTR-CAP<br>The pin capture capability. | PIC X(1) |
| 53 | 04 | SOCL-DEPRIVATN-AREA<br>Area of social deprivation indicator. | PIC X(1) |
| 54 | 04 | OUTSIDE-HOME-TERRITORY-IND<br>Outside home territory indicator. | PIC X(1) |
| 55-60 | 04 | P61-RSRVD-1<br>Not used. | PIC X(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 61-70 | 04 | POST-CDE<br>Postal code. | PIC X(10) |
| 71-80 | 04 | P61-RSRVD-2<br>Not used. | PIC X(10) |
| 81–86 | 04 | ISS-TRACE-ID<br>The issuer trace ID. | PIC X(6) |
| 87-88 | 04 | PROD-TYP<br>Product type for Non-Cash Products.<br>Bit Map Position = 123 - subfield 1 | PIC X(2) |
| 89-187 | 04 | BILATERAL-DISC-DATA<br>Bilateral discretionary Data for Non-Cash Products.<br>Bit Map Position = 123 - subfield 18 | PIC X(99) |
| 188-227 | 04 | SENDER-PROXY<br>Sender proxy for LINK MPT (Mobile Payment Transactions).<br>Bit Map Position = 123 - subfield 41 | PIC X(40) |
| 228-267 | 04 | RECIPIENT-PROXY<br>Recipient proxy for LINK MPT transactions.<br>Bit Map Position = 123 - subfield 42 | PIC X(40) |
| 268-285 | 04 | TXN-REF<br>Transaction reference for LINK MPT transactions.<br>Bit Map Position = 123 - subfield 43 | PIC X(18) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 286-296 | 04 | NSC-BIC<br>NSC/BIC for LINK MPT transactions.<br>Bit Map Position = 123 - subfield 44 | PIC X(11) |
| 297-330 | 04 | ACCT-IBAN<br>Account/IBAN for LINK MPT transactions.<br>Bit Map Position = 123 - subfield 45 | PIC X(34) |
| 331-370 | 04 | PMNT-REF<br>Payment reference for LINK MPT transactions.<br>Bit Map Position = 123 - subfield 46 | PIC X(40) |

## MAC MASM (MACI)

The following fields define the values in the B0 and B1 tokens that are used by the MAC MASM (MACI) interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–70 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–70 | 02 | MACI<br>MACI specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The switch token version. | PIC X(2) |
| 9–11 | 04 | TYP-QUAL<br>The message type qualifier. | PIC X(3) |
| 12–17 | 04 | TRACE-NUM<br>The trace number. This code uniquely identifies a cardholder transaction. | PIC X(6) |
| 18–19 | 04 | NETWK-ID<br>The network ID. | PIC X(2) |
| 20–21 | 04 | DENIAL-CDE<br>The denial code. For denied transactions, this code indicates the reason for the denial. | PIC X(2) |
| 22–27 | 04 | TERM-TIM<br>The terminal time. | PIC X(6) |
| 28–42 | 04 | PRE-AUTH-KEY<br>The preauthorization key. | PIC X(15) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 43 | 04 | ICHG-RATE-IND<br>The interchange rate indicator. | PIC X(1) |
| 44–55 | 04 | MCCR-FEE-AMT<br>The MasterCard Conversion Fee amount. This code provides a fee assessed by MasterCard whenever the acquirer country code does not correspond with the issuer country code. | PIC X(12) |
| 56–70 | 04 | AIRLN-TCKT-NUM<br>The airline ticket number. | PIC X(15) |

## MDS Cirrus ISO

The following fields define the values in the B0 and B1 tokens that are used by the MDS Cirrus ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–284 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–284 | 02 | MDS-SWI-TKN-DATA<br>MDS specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER-ID<br>The switch token version ID. | PIC X(2) |
| 9–10 | 04 | NTWK-ID-CDE<br>The network ID code. | PIC X(2) |
| 11–22 | 04 | REF-NUM<br>The transaction reference number. | PIC X(12) |
| 23–24 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 25–27 | 04 | POS-ENTRY-MODE<br>The point-of-service entry mode. The field contains two codes. The first code is two digits in length and indicates the method by which Track 2 data or the primary account number (PAN) was entered into the system. The second code is one digit in length and indicates the entry capabilities available at the point of service. | PIC 9(3) |
| 28 | 04 | TERM-LOC<br>The terminal location. | PIC 9(1) |
| 29–37 | 04 | SWI-REF-NUM<br>The switch reference number. | PIC X(9) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 38–52 | 04 | AIRLINE-TCKT-NUM<br>The airline ticket number. | PIC X(15) |
| 53–58 | 04 | PROC-CDE<br>The transaction processing code. | PIC X(6) |
| 59–61 | 04 | SETL-SRVC-DATA<br>The MDS settlement service indicator. | PIC X(3) |
| 62–65 | 04 | ADVC-DETL-CDE<br>The MDS advice detail code. | PIC 9(4) |
| 66–71 | 04 | ON-BEHALF | OCCURS 2 TIMES |
| | 06 | SVC<br>A code containing the MDS On-Behalf Services data. Valid values are as follows:<br>01 = M/Chip to Magnetic Stripe Conversion<br>02 = M/Chip Cryptogram Pre-Validation/Post-Generation<br>03 = Dynamic M/Chip Stand-In<br>04 = Reserved<br>05 = Accountholder Authentication Verification Service<br>06 = Dynamic Accountholder Authentication Verification Service | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 06 | RSLT-1<br>A code indicating the MDS On-Behalf Result 1 value. Valid values are as follows:<br>C = Conversion of the M/Chip transaction to a magnetic stripe transaction completed.<br>G = Application Cryptogram is valid, but it is not an ARQC. Status of TVR/CVR unknown.<br>I = Application Cryptogram (AC) is incorrect or invalid. Status of TVR/CVR unknown.<br>T = Valid ARQC, TVR/CVR invalid.<br>U = Unable to process. No check on cryptogram. Status of TVR/CVR unknown.<br>V = Valid ARQC, valid TVR/CVR.<br>b = No value present (where b indicates a blank space). | PIC X(1) |
| 72-83 | 04 | CRDHLDR-BILLING-AMT<br>The MDS cardholder billing amount. | PIC 9(12) |
| 84–95 | 04 | MCCR-AMT<br>The MDS MCCR amount. | PIC 9(12) |
| 96–103 | 04 | ICCR-AMT<br>The MDS ICCR amount. | PIC 9(8) |
| 104–109 | 04 | TIERED-MERCH-ID<br>The MDS tiered merchant identification code. | PIC 9(6) |
| 110–112 | 04 | QCK-PYMNT-SRV-IND<br>The MDS quick payment services indicator. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 113–117 | 04 | GCMS-PROC-DAT-CYC | PIC X(5) |
| | | The GCMS settlement date and settlement cycle number. Identifies when the settlement date and cycle are different from the reconciliation date and cycle. | |
| 118–122 | 04 | GCMS-ADNL-POS-DATA | PIC X(5) |
| | | A code containing additional POS data. This code gives issuers additional information about the conditions surrounding the transaction. | |
| 123–134 | 04 | GCMS-BUS-ACTIVITY | PIC X(12) |
| | | A code indicating the type of business activity. This code identifies the type of business arrangement applied to this transaction. | |
| 135–153 | 04 | GCMS-SETL-DATA | PIC X(19) |
| | | A code containing settlement data. This code gives the issuer additional information about the settlement of the transaction. | |
| 154–155 | 04 | MCHIP-PRO-IND | PIC X(2) |
| | | A code containing M/Chip processing information. This code gives acquirers more information about cryptogram validation.<br>02 = MasterCard On-Behalf Service – M/Chip Cryptogram pre-validation<br>03 = MasterCard On-Behalf Service – M/Chip Cryptogram validation in stand-in processing<br>50 = Issuer chip validation | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 156 | 04 | MCHIP-PROC-INFO | PIC X(1) |
| | | A code containing M/Chip processing information. This code gives acquirers more information about cryptogram validation.<br>I = Application Cryptogram invalid<br>U = Application Cryptogram not validated due to technical error<br>F = Format error in DE 55<br>G = Cryptogram in application is valid but is not an ARQC<br>T = Application Cryptogram is valid but TVR/CVR is invalid<br>X = Issuer provided incorrect value in subfield 2 | |
| 157-165 | 04 | ORIG-SWI-SERL-NUM | PIC X(9) |
| 166 | 04 | ACCT-CAT | PIC X(1) |
| 167 | 04 | PAYPASS-DEV-RQST-RESP | PIC X(1) |
| 168 | 04 | ICHG_RATE-IND | PIC X(1) |
| | | Carries the Interchange Rate Indicator from data element 63, subfield 2 of the Interchange message. | |
| 169 | 04 | CVC2_PRG-IND | PIC X(1) |
| | | Carries the Program Participation Indicator from data element 94 of the Interchange message. | |
| 170 | 04 | CHRGBCK-IND | PIC X(1) |
| | | Message field: data element 48, subelement 94, subfield 2. | |
| 171-172 | 04 | PMNT-INITIATE-CHAN | PIC X(2) |
| | | Message field: data element 48, subelement 23. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 173 | 04 | PRMTN-CDE | PIC X(1) |
| | | Message field: data element 48, subelement 95 | |
| 174-176 | 04 | USER-FLD-ACI | PIC X(3) |
| | | Reserved for future use. | |
| 177-185 | 04 | TRAN-FEE | PIC X(9) |
| 186 | 04 | ACCT-NUM-IND | PIC X(1) |
| 187-205 | 04 | VPAN | PIC X(19) |
| 206-255 | 04 | CHIP-BIT-ERR-RSLTS | OCCURS 10 TIMES |
| | 06 | CVR-TVR-ID | PIC X(1) |
| | 06 | BYTE-ID | PIC X(2) |
| | 06 | BIT-ID | PIC X(1) |
| | 06 | BIT-ERR | PIC X(1) |
| 256-258 | 04 | PROD-ID | PIC X(3) |
| 259-260 | 04 | BUS-APPL-ID | PIC X(2) |
| 261-272 | 04 | FRAUD DATA | |
| | | Carries the fraud scoring data from data element 75 of the Interchange message. | |
| | 06 | SCORE | PIC X(3) |
| | 06 | RSN-CDE | PIC X(2) |
| | 06 | RULE-ADJ-SCORE | PIC X(3) |
| | 06 | RULE-RSN-CDE-1 | PIC X(2) |
| | 06 | RULE-RSN-CDE-2 | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 273-284 | 04 | GRATUITY-AMT<br>Carries the gratuity amount. | PIC 9(12) |

## Money Station (MONY)

The following fields define the values in the B0 and B1 tokens that are used by the Money Station (MONY) interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–34 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 28 characters. | PIC X(442) |
| 7–34 | 02 | MONY<br>MONY-specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The switch token version. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9–14 | 04 | STAN<br>The System Trace Audit Number. This code uniquely identifies a cardholder transaction. | PIC X(6) |
| 15–16 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 17–20 | 04 | NETWK-ID<br>The network ID for issuers; otherwise, blanks. | PIC X(4) |
| 21–26 | 04 | SWI-DAT<br>The switch date. | PIC X(6) |
| 27–32 | 04 | SWI-TIM<br>The switch time. | PIC X(6) |
| 33 | 04 | OFF-PREMISE<br>A flag indicating whether the ATM where the transaction was initiated was off-premise. A value of Y indicates off-premise; a blank indicates not off-premise. | PIC X(1) |
| 34 | 04 | USER-FLD | PIC X(1) |

## MPS

The MPS interface does not use the B0 and B1 token.

# NBGC

The following fields define the values in the B0 and B1 tokens that are used by the NBGC interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–102 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–102 | 02 | NBGC<br>NPC ISO specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER-ID<br>The switch token version ID. | PIC X(2) |
| 9–12 | 04 | MSG-TYP<br>Message Type Identifier. | PIC 9(4) |
| 13–18 | 04 | PROC-CDE<br>Bit map position 3 (ISO Processing Code). | PIC 9(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 19–24 | 04 | STAN<br>Bit map position 11 ( System Trace Audit Number). | PIC 9(6) |
| 25–28 | 04 | EXP-DAT<br>Bit map position 14 (Expiration Date). | PIC 9(4) |
| 29–31 | 04 | ACQ-CNTRY-CDE<br>Bit map position 19 (Acquiring Institution Country Code). | PIC 9(3) |
| 32–34 | 04 | POS-ENTRY-MDE<br>Bit map position 22 (Point of Service Entry Mode). | PIC 9(3) |
| 35–36 | 04 | POS-COND-CDE<br>Bit map position 25 (Point of Service Condition Code). | PIC 9(2) |
| 37–47 | 04 | FWD-INST-ID-CDE<br>Bit map position 33 (Forwarding Institution Id Number). | PIC X(11) |
| 48–59 | 04 | RET-REF-NUM<br>Bit map position 37 (Retrieval Reference Number). | PIC X(12) |
| 60–61 | 04 | RESP-CDE<br>Bit map position 39 (Response Code). | PIC X(2) |
| 62–76 | 04 | CRD-ACPT-ID-CDE<br>Bit map position 42 (Card Acceptor Identification Code). | PIC X(15) |
| | 04 | POS-DATA<br>Bit map position 63 (Private use- POS Data). | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 77–87 | 06 | COND-CDE | PIC 9(11) |
| 88–89 | 06 | AUTH-LIFE-CYCLE | PIC 9(2) |
| 90–92 | 06 | CNTRY-CDE | PIC 9(3) |
| 93–102 | 06 | POSTAL-CDE | PIC X(10) |

## NPC ISO

The following fields define the values in the B0 and B1 tokens that are used by the NPC ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–44 | | SWI-TKN | |
| 1–2 | 02 | LGTH | TYPE BINARY 16 SIGNED |
| | | The length of the token data. The length includes the FIID and the data present in the data buffer. | |
| 3–6 | 02 | FIID | PIC X(4) |
| | | The FIID assigned to the interface. The value in this field is hard-coded. | |
| 7–448 | 02 | BUF | PIC X(442) |
| | | The generic data, which has a variable length up to 442 characters. | |
| 7–44 | 02 | NPCI | REDEFINES BUF |
| | | NPC ISO specific data used in the generic switch token. | |
| 7–8 | 04 | VER | PIC X(2) |
| | | The Switch token version ID. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9–10 | 04 | AUTH-NETWORK-ID<br>The issuer trace data network ID. | PIC X(2) |
| 11–12 | 04 | AUTH-NETWORK-RESP<br>The issuer trace data network response. | PIC X(2) |
| 13–37 | 04 | ADDL-RESP-DATA<br>This field contains additional response data. | PIC X(25) |
| 38–43 | 04 | STAN<br>The system trace audit number. | PIC X(6) |
| 44 | 04 | USER-FLD<br>Blank-filled. | PIC X(1) |

## NYCE ISO

The following fields define the values in the B0 and B1 tokens that are used by the NYCE ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–104 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–104 | 02 | NYCI<br>NYCE ISO specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–18 | 04 | SWI-TXN-DAT-TIM<br>The switch transaction date and time. | PIC X(10) |
| 19–24 | 04 | STAN<br>The system trace audit number. | PIC X(6) |
| 25–26 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 27–29 | 04 | NETWK-ID<br>This field contains NYCI data. | PIC X(3) |
| 30 | 04 | OFF-PREMISE<br>The off-premise flag. | PIC X(1) |
| 31 | 04 | BILL-PAY<br>The billpay transaction indicator. Valid values are as follows:<br>Y = Yes, this is a billpay transaction.<br>␣ = No, this is not a billpay transaction (where ␣ indicates a blank space). | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 32–37 | 04 | PROC-CDE<br>The processing code. | PIC X(6) |
| 38 | 04 | BAL-RTRND-BY-HOST<br>The balances returned by host indicator. Valid values are as follows:<br>Y = Yes, the host has returned balances.<br>b = No, the host has not returned balances (where b indicates a blank space). | PIC X(1) |
| 39–50 | 04 | DDA-BAL<br>Balance of checking account returned by host. If customer does not have a checking account or balances have not been returned by a host, this field will be spaces. | PIC X(12) |
| 51–62 | 04 | SAV-BAL<br>Balance of savings account returned by host. If customer does not have a savings account or balances have not been returned by a host, this field will be spaces. | PIC X(12) |
| 63–74 | 04 | CR-BAL<br>Balance of credit account returned by host. If customer does not have a credit account or balances have not been returned by a host, this field will be spaces. | PIC X(12) |
| 75–86 | 04 | OTH-BAL<br>Balance of other account returned by host. If customer does not have an other account or balances have not been returned by a host, this field will be spaces. | PIC X(12) |
| 87 | 04 | ISS-CAT<br>Issuer Category field. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 88-90 | 04 | PT-SVC-ENTRY-MDE | PIC X(3) |
| | | Point of Entry Mode field. | |
| 91-101 | 04 | NATL-PT-SVC-COND-CDE | |
| | | National Point of Service Code field. | |
| | 06 | TERM-CLASS | PIC X(3) |
| | 06 | PRSTN-DATA | PIC X(4) |
| | 06 | SEC-COND | PIC X(1) |
| | 06 | TERM-TYPE | PIC X(2) |
| | 06 | CRD-DATA-INPUT-CAP | PIC X(1) |
| 102-103 | 04 | PIN-DESCR-DATA | PIC X(2) |
| | | PIN Description Data field. | |
| 104 | 04 | PIN-IND-DATA | PIC X(1) |
| | | PIN Indicator Data field. | |

## Networks ISO

The following fields define the values in the B0 and B1 tokens that are used by the Networks ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–30 | | SWI-TKN | |
| 1–2 | 02 | LGTH | TYPE BINARY 16 SIGNED |
| | | The length of the token data. The length includes the FIID and the data present in the data buffer. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–30 | 02 | NETI<br>NetWorks ISO specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–18 | 04 | SWI-TXN-DAT-TIM<br>The switch transaction date and time. | PIC X(10) |
| 19–24 | 04 | STAN<br>The system trace audit number. | PIC X(6) |
| 25–26 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 27–29 | 04 | NETWK-ID<br>This field contains NETI data. | PIC X(3) |
| 30 | 04 | FILLER<br>Blank-filled. | PIC X(1) |

# PLUS ISO

The following fields define the values in the B0 and B1 tokens that are used by the PLUS ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–174 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–174 | 02 | PISO-SWI-TKN-DATA<br>PLUS ISO specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER-ID<br>The switch token version ID. | PIC X(2) |
| 9–12 | 04 | NETWK-ID<br>The network ID. | PIC X(4) |
| 13–23 | 04 | FRWD-INST-ID<br>The identification of the forwarding institution. | PIC 9(11) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 24–63 | 04 | PSI-REF-INFO<br>Reference information. | PIC X(40) |
| 64–74 | 04 | RCV-INST-ID<br>The identification of the receiving institution. | PIC 9(11) |
| 75–82 | 04 | TRAN-FEE<br>The transaction fee. | PIC 9(8) |
| 83–84 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 85–87 | 04 | ENTRY-MDE<br>The point-of-service entry mode. The field contains two codes. The first code is two digits in length and indicates the method by which Track 2 data or the primary account number (PAN) was entered into the system. The second code is one digit in length and indicates the entry capabilities available at the point of service. | PIC 9(3) |
| 88–93 | 04 | TRACE-NUM<br>The trace number. | PIC 9(6) |
| 94–96 | 04 | CRD-ISS-INST-CNTRY-CDE<br>The card-issuing institution’s country code. | PIC 9(3) |
| 97–99 | 04 | ACQ-INST-CNTRY-CDE<br>The acquiring institution’s country code. | PIC 9(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 100 | 04 | DOC-INFO-IND<br>A code that identifies the status of the supporting documentation. | PIC X(1) |
| 101–106 | 04 | EXCPT-REF-NUM<br>The exception reference number. The acquirer uses this code to track adjustments for control purposes. The issuer uses this code for matching chargebacks between internal systems and Visa/Plus. Use of this code is optional for both the acquirer and the issuer. | PIC 9(6) |
| 107–156 | 04 | MBR-MSG-TXT<br>The member message text code. This code contains unformatted text describing an adjustment chargeback or representment. | PIC X(50) |
| 157 | 04 | ISA-FEE<br>The international service assessment fee. | PIC 9(1) |
| 158–160 | 04 | FEE-PGM<br>The fee program indicator. | PIC X(3) |
| 161–172 | 04 | MCCR-FEE<br>The amount calculated by MasterCard in the cardholder billing currency. | PIC X(12) |
| 173–174 | 04 | CRD-LVL-PROD-ID-VAL<br>The card-level produciton identification value. | PIC X(2) |

# Pulse ISO

The following fields define the values in the B0 and B1 tokens that are used by the Pulse ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–160 | | SWI-TKN | |
| 1–2 | 02 | LGTH | TYPE BINARY 16 SIGNED |
| | | The length of the token data. The length includes the FIID and the data present in the data buffer. | |
| 3–6 | 02 | FIID | PIC X(4) |
| | | The FIID assigned to the interface. The value in this field is hard-coded. | |
| 7–160 | 02 | BUF | PIC X(154) |
| | | The generic data, which has a variable length up to 154 characters. | |
| 7–160 | 02 | PULI | REDEFINES BUF |
| | | PULSE ISO specific data used in the generic switch token. | |
| 7–8 | 04 | VER | PIC X(2) |
| | | The Switch token version ID. | |
| 9–20 | 04 | CHB-AMT | PIC X(12) |
| | | The cardholder billing amount. | |
| 21–30 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| | | The switch transaction date and time. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 31–38 | 04 | CHB-CONV-RATE<br>The cardholder conversion rate. | PIC X(8) |
| 39–44 | 04 | STAN<br>The system trace audit number. | PIC X(6) |
| 45–46 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 47–49 | 04 | NETWK-ID<br>This field contains PULI data. | PIC X(3) |
| 50 | 04 | OFF-PREMISE | PIC X(1) |
| 51–150 | 04 | ISS-TRC-DATA<br>The issuer trace data. | PIC X(100) |
| 151-160 | 04 | NATL-PT-SVC-COND-CDE<br>The national POS condition code. | PIC X(10) |

# SPAN2

The following fields define the values in the B0 and B1 tokens that are used by the SPAN2 interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–94 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 94 characters. | PIC X(442) |
| 7–94 | 02 | SPAN<br>SPAN-specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER-ID<br>The switch token version ID. | PIC X(2) |
| 9–14 | 04 | STAN<br>Bit map position 11 (System Trace Audit Number). | PIC X(6) |
| 15–26 | 04 | PT-SVC-DATA-CDE<br>Bit map position 22 (Point of Service Data Code). | PIC X(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 27–38 | 04 | RETRV-REF-NUM<br>Bit map position 37 (Retrieval Reference Number). | PIC X(12) |
| 39–44 | 04 | APPRV-CDE<br>Bit map position 38 (Approval Code). | PIC X(6) |
| 45–47 | 04 | ACT-CDE<br>Bit map position 39 (Action Code). | PIC X(3) |
| 48–70 | 04 | XPORT-DATA<br>Bit map position 59 (Transport Data). | PIC X(23) |
| 71–93 | 04 | RSRVD-62-SPAN<br>Bit map position 62 (Private Field - Terminal Status). | PIC X(23) |
| 94 | 04 | USR-FLD | PIC X(1) |

## Star ISO

The following fields define the values in the B0 and B1 tokens that are used by the Star ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–142 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–6 | 02 | FIID | PIC X(4) |
| | | The FIID assigned to the interface. The value in this field is hard-coded. | |
| 7–448 | 02 | BUF | PIC X(442) |
| | | The generic data, which has a variable length up to 442 characters. | |
| 7–142 | 02 | STRI | REDEFINES BUF |
| | | STAR ISO specific data used in the generic switch token. | |
| 7–8 | 04 | VER | PIC X(2) |
| | | The Switch token version ID. | |
| 9–18 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| | | The transmission date and time. | |
| 19–24 | 04 | STAN | PIC X(6) |
| | | The system trace audit number. | |
| 25–26 | 04 | RESP-CDE | PIC X(2) |
| | | The response code. | |
| 27–29 | 04 | NETWK-ID | PIC X(3) |
| | | The network ID for the issuer or acquirer. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 30 | 04 | BILL-PAY | PIC X(1) |

The billpay transaction indicator. Valid values are as follows:

Y = Yes, this is a billpay transaction.
b = No, this is not a billpay transaction (where b indicates a blank space).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 31 | 04 | OFF-PREMISE | PIC X(1) |

The off premise indicator. Valid values are as follows:

Y = Yes, this is an off premise transaction.
b = No, this is not an off premise transaction (where b indicates a blank space).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 32–33 | 04 | COMPL-CNT | PIC X(2) |

The number of completions in the series for Visa Check Signature Debit transactions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 34–39 | 04 | PROC-CDE | PIC X(6) |

The processing code.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 40 | 04 | LAST-COMPL-IND | PIC X(1) |

A flag that indicates the last completion in a series for Visa Check Signature Debit transactions. Valid values are as follows:

L = The last completion in a series
b = Not the last completion in a series (where b indicates a blank space)

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 41–55 | 04 | MRCH-ACQ-TCKT-NUM | PIC X(15) |

The merchant acquirer railway or airline ticket number.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 56–57 | 04 | ICHG-RATE-IND | PIC X(2) |

The interchange rate indicator.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 58–60 | 04 | PGM-RGSTR-ID<br>The program registration ID. | PIC X(3) |
| 61–72 | 04 | MCCR-AMT<br>The cardholder billing amount. | PIC X(12) |
| 73 | 04 | MCCR-FEE-CR-DB-IND<br>The additional fees indicator. | PIC X(1) |
| 74–81 | 04 | MCCR-FEE-AMT<br>The additional fees amount. | PIC X(8) |
| 82 | 04 | MCCR-FEE-RPLMT-CR-DB-IND<br>The additional replacement fees indicator. | PIC X(1) |
| 83–90 | 04 | MCCR-FEE-RPLMT-AMT<br>The additional replacement fees amount. | PIC X(8) |
| 91–92 | 04 | CRD-LVL-RSLTS | PIC X(2) |
| 93 | 04 | CRDHLDR-ID-METHOD | PIC X(1) |
| 94–97 | 04 | MSG-RSN-CDE | PIC X(4) |
| 98 | 04 | RECUR-PMNT-IND | PIC X(1) |
| 99–101 | 04 | SVC-DVLP-FLD | PIC X(3) |
| 102 | 04 | CVC2-VALID-PRTCPT-IND | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 103 | 04 | CHRGBCK-ELIGIBILITY-IND | PIC X(1) |
| 104–105 | 04 | FRAUD-SCORE-RSN-CDE | PIC X(2) |
| 106–107 | 04 | BUS-APPL-ID | PIC X(2) |
| 108 | 04 | AVS-RSLT-CDE | PIC X(1) |
| 109–116 | 04 | VISA-WATCH-LIST | |
| | 06 | MGMT-VALID-CDE | PIC X(4) |
| | 06 | MGMT-RSLTS-CDE | PIC X(3) |
| | 06 | VMT-ACTVTY-CHK-RSLT | PIC X(1) |
| 117 | 04 | REMT-PMNT-PGM-TYP-ID | PIC X(1) |
| 118-119 | 04 | PMNT-INIT-CHAN-DEV-TYP | PIC X(2) |
| 120 | 04 | MC-RATE-IND | PIC X(1) |
| 121-123 | 04 | STAR-ISSUER-IGI | PIC X(3) |
| 124-133 | 04 | NATL-PT-SVC-COND-CDE | PIC X(10) |
| 134-141 | 04 | TXN-DESCR | |
| | 06 | INTERLINK-ATTR | PIC X(1) |
| | 06 | ICHG-GRP-ID | PIC X(3) |
| | 06 | AGGR-IND | PIC X(1) |
| | 06 | AUTH-PGM | PIC X(1) |
| | 06 | TXN-SUPTYP | PIC X(1) |
| | 06 | PROD-ID | PIC X(1) |
| 142 | 04 | USER-FLD-ACI | PIC X(1) |

# SVS

The following fields define the values in the B0 and B1 tokens that are used by the SVS ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–44 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–44 | 02 | SVS<br>SVS specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–14 | 04 | SVS-TRAN-CDE<br>The processing code. | PIC X(6) |
| 15–26 | 04 | APPROVAL-AMT<br>The transaction amount. | PIC X(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 27–36 | 04 | XMIT-DAT-TIM<br>The transmission date and time. | PIC X(10) |
| 37–42 | 04 | BAL-AMT<br>The approval code. | PIC X(6) |
| 43–44 | 04 | RESP-CDE<br>The response code. | PIC X(2) |

## Shazam ISO

The following fields define the values in the B0 and B1 tokens that are used by the Shazam ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–126 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–126 | 02 | SHZM<br>Shazam specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The Switch token version ID. | PIC X(2) |
| 9–18 | 04 | SWI-TXN-DAT-TIM<br>The transmission date and time. | PIC X(10) |
| 19–24 | 04 | STAN<br>The systems trace audit number. | PIC X(6) |
| 25–27 | 04 | FNCT-CDE<br>The function code. | PIC X(3) |
| 28–30 | 04 | ACT-CDE<br>The action code. | PIC X(3) |
| 31–33 | 04 | NETWORK-ID | PIC X(3) |
| 34 | 04 | BILL-PAY<br>Billpay transaction indicator. Valid values are as follows:<br>Y = Yes, this is a billpay transaction.<br>b = No, this is not a billpay transaction (where b indicates a blank space). | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 35–50 | 04 | BP-ORG-NAME | PIC X(16) |
| 51–67 | 04 | BP-ACCT-NUM | PIC X(17) |
| 68 | 04 | TRAN-IND | PIC X(1) |
| 69–70 | 04 | RETRV-REF-NUM<br>The retrieval reference number. | PIC X(2) |
| 71–76 | 04 | KEY-SEQ-NUM<br>The KEY-SEQ-NUM is used to format the PSTM prikey with data elements from the external request message. | PIC X(6) |
| 77-125 | 04 | CRD-ACCPT-NAM-LOC<br>Holds the card acceptor name location data for Shazam Billpay transactions. | PIC X(49) |
| 126 | 04 | USER-FLD-ACI | PIC X(1) |

## ValueLink

The following fields define the values in the B0 and B1 tokens that are used by the ValueLink interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–44 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–6 | 02 | FIID | PIC X(4) |
| | | The FIID assigned to the interface. The value in this field is hard-coded. | |
| 7–448 | 02 | BUF | PIC X(442) |
| | | The generic data, which has a variable length up to 442 characters. | |
| 7–44 | 02 | VLNK | REDEFINES BUF |
| | | ValueLink specific data used in the generic switch token. | |
| 7–8 | 04 | VER | PIC X(2) |
| | | The Switch token version ID. | |
| 9–16 | 04 | PREV-BAL | PIC 9(8) |
| | | The balance before the transaction was applied. | |
| 17–24 | 04 | NEW-BAL | PIC 9(8) |
| | | The customer account balance after the transaction request is completed. | |
| 25–28 | 04 | CRD-CLS | PIC 9(4) |
| | | Merchant-defined identifier that is stored in the ValueLink database and returned in a response message. A value of 0 indicates the card does not belong to any card class. | |
| 29–36 | 04 | CB-AMT | PIC 9(8) |
| | | The amount of cash the customer should receive as a result of one of the following:<br>• The balance going below the minimum balance for the card type. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | • The customer requesting a cash out (less the transaction amount, which could be zero). | |
| 37–44 | 04 | SALE-AMT<br>The sale amount of the card. If the sale amount of the card is different from the account balance due to a programmed discount amount, this field will be present during an activation or reload. | PIC 9(8) |

## Visa Debit Processing Service (DPS)

The following fields define the values in the B0 and B1 tokens that are used by the Visa Debit Processing Service (DPS) interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–90 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–90 | 02 | VDPS<br>Visa Debit Processing Service (DPS) specific data used in the generic switch token. | REDEFINES BUF |
| 7–8 | 04 | VER<br>The switch token version ID. | PIC X(2) |
| 9–18 | 04 | SWI-TXN-DAT-TIM<br>The transaction date and time. | PIC X(10) |
| 19–24 | 04 | STAN<br>The system trace audit number. | PIC X(6) |
| 25–36 | 04 | RETRVL-REF-NUM<br>The retrieval reference number. | PIC X(12) |
| 37–38 | 04 | RESP-CDE<br>The response code. | PIC X(2) |
| 39–41 | 04 | NETWK-ID<br>The network ID. This field contains data specific to the Visa DPS switch. | PIC X(3) |
| 42 | 04 | OFF-PREMISE<br>The off-premise indicator. This code indicates the transaction as occurring at either a remote or non-remote ATM terminal. Valid values are as follows:<br>Y = The transaction originated at an off-premise location.<br>␣ = The transaction did not originate at an off-premise location (where ␣ indicates a blank space). | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 43 | 04 | BILL-PAY<br>A code indicating whether or not the transaction is a billpay transaction. Valid values are as follows:<br>Y = Yes, the transaction is a bill payment transaction.<br>b = No, the transaction is not a bill payment transaction (where b indicates a blank space). | PIC X(1) |
| 44–46 | 04 | FEE-PGM-IND<br>The Fee Program Indicator. | PIC X(3) |
| 47–48 | 04 | E-COMM-GDS-IND<br>The Electronic Commerce Goods Indicator. | PIC X(2) |
| 49–58 | 04 | MRCH-VRFCN-VAL<br>The Merchant Verification Value. | PIC X(10) |
| 59–80 | 04 | RSK-DATA<br>The following fields contain information related to fraud prevention. | |
| 59–70 | 06 | FLCN | |
| 59 | 08 | SCR-SRC<br>The source of the risk score. | PIC X(1) |
| 60–63 | 08 | SCR-VAL<br>The value of the risk score. | PIC X(4) |
| 64 | 08 | RESP-CDE<br>The repsonse code issued based on the risk scoring. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 65–66 | 08 | RSN1<br>The first two-digit reason code. | PIC X(2) |
| 67–68 | 08 | RSN2<br>The second two-digit reason code. | PIC X(2) |
| 69–70 | 08 | RSN3<br>The third two-digit reason code. | PIC X(2) |
| 71–80 | 06 | VISA<br>The following fields contain information on the Visa fraud reason and condition coding. | |
| 71–72 | 08 | SCR<br>The risk score. | PIC X(2) |
| 73–74 | 08 | RSN-CDE<br>The reason code. | PIC X(2) |
| 75–76 | 08 | COND-CDE1<br>The first condition code. | PIC X(2) |
| 77–78 | 08 | COND-CDE2<br>The second condition code. | PIC X(2) |
| 79–80 | 08 | COND-CDE3<br>The third condition code. | PIC X(2) |
| 81–82 | 04 | CRD-LVL-RSLT-VAL<br>The Card Level Result Value. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 83–90 | 04 | VISA-WATCH-LIST<br>Visa watch list management data, carried in field 111.31 of the Visa message. | |
| 83–86 | 06 | MGMT-VALID-CDE | PIC X(4) |
| 87–89 | 06 | MGMT-RSLT-CDE | PIC X(3) |
| 90 | 06 | VMT-ACTVTY-CHK-RSLT | PIC X(1) |

## VisaNet

The following fields define the values in the B0 and B1 tokens that are used by the VisaNet interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–220 | | SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. The length includes the FIID and the data present in the data buffer. | TYPE BINARY 16 SIGNED |
| 3–6 | 02 | FIID<br>The FIID assigned to the interface. The value in this field is hard-coded. | PIC X(4) |
| 7–448 | 02 | BUF<br>The generic data, which has a variable length up to 442 characters. | PIC X(442) |
| 7–220 | 02 | VISA-SWI-TKN-DATA<br>VisaNet specific data used in the generic switch token. | REDEFINES BUF |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7–8 | 04 | VER-ID<br>The switch token version ID. | PIC X(2) |
| 9–12 | 04 | NTWK-ID-CDE<br>The network identification code. This field identifies the originating network or an interim originating network. | PIC X(4) |
| 13–23 | 04 | ISS-INST-ID-CDE<br>The issuing institution's routing and transit number or issuer identification number. This code identifies the issuer when the issuer cannot be determined from the account number in the message. | PIC X(11) |
| 24–27 | 04 | SETL-DAT<br>The transaction's settlement month and day (MM/DD). | PIC 9(4) |
| 28–38 | 04 | FWD-INST-CDE<br>The forwarding institution's identification code. This code identifies the message originator. | PIC X(11) |
| 39–42 | 04 | POS-ENTRY-MDE<br>The Visa value for point-of-service entry mode. This field contains two codes. The first code is two digits in length and indicates the method by which Track 2 data or the primary account number (PAN) was entered into the system. The second code is one digit in length and indicates the entry capabilities available at the point of service. | PIC 9(4) |
| 43–44 | 04 | RESP-CDE<br>The response code. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 45–50 | 04 | PMC-ID<br>The Proprietary Member Center (PMC) identifier. | PIC X(6) |
| 45–50 | 04 | USER-APR08-FLD | REDEFINES PMC-ID |
| 45–50 | 06 | DATA | PIC X(6) |
| 51–52 | 04 | MRCH-VOL-IND<br>The merchant volume indicator. This code is used in Visa's Volume-tiered Interchange Fee Program. | PIC X(2) |
| 53 | 04 | INTRNTL-SRV-ASMNT<br>A flag indicating whether the transaction is subject to an international service assessment. | PIC X(1) |
| 54–55 | 04 | CRD-RSLTS<br>A code indicating the participation program for the card involved in the transaction. | PIC X(2) |
| 56 | 04 | CCDR-IND<br>The Commercial Card Data Repository (CCDR) indicator. This code indicates whether or not CCDR data is included in the transaction. Valid values are as follows.<br>0 = Enhanced data is not included in the CCDR<br>1 = Enhanced data is included in the CCDR | PIC X(1) |
| 57–59 | 04 | FEE-PGM-IND<br>The fee program indicator. This code indicates to which fee program the message is assigned. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 60–63 | 04 | STIP-CDE | PIC X(4) |
| | | The Stand In Processing Code. This code indicates why SMS STIP responded for the issuer or why the SMS switch generated an advice. | |
| 64–67 | 04 | MSG-TYP | PIC 9(4) |
| | | The type of message sent to Visa (0100, 0110, 0200, etc.). | |
| 68 | 04 | DEF-BILL-IND | PIC X(1) |
| | | The deferred billing indicator. This code identifies transactions for which the billing of the merchandise occurred after the merchandise was delivered to the cardholder. | |
| 68 | 04 | USER-OCT02-FLD5 | REDEFINES DEF-BILL-IND |
| | | This field is replaced by the DEFERRED-BILLING-IND field located in the BASE24-pos Merchant token (token CI). | |
| 68 | 06 | DATA | PIC X(1) |
| | | This field ensures the operation of the redefine. | |
| 69–74 | 04 | PROC-CDE | PIC X(6) |
| | | The processing code sent to or received from Visa. | |
| 75 | 04 | REIMB-ATTR | PIC X(1) |
| | | The reimbursement attribute. This code identifies the applicable interchange reimbursement fee for the transaction. | |
| 76–77 | 04 | ECOMM-GDS-IND | PIC X(2) |
| | | The electronic commerce goods indicator. This code indicates the type of goods purchased in an Internet transaction. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 76–77 | 04 | USER-OCT02-FLD6 REDEFINES ECOMM-GDS-IND<br>This field replaces the E-COMM-GOODS-IND now located in the BASE24-pos Merchant token (token CI). | |
| 76–77 | 06 | DATA<br>This field ensures the operation of the redefine. | PIC X(2) |
| 78 | 04 | CAVV-RSLT-CDE<br>The cardholder authentication verification value. This code indicates the outcome of CAVV validation. | PIC X(1) |
| 78 | 04 | USER-OCT02-FLD7 REDEFINES CAVV-RSLT-CDE<br>This field replaces the CAVV-RSLT-CDE field now located in the BASE24-pos Release 5.1 token (token C0). | |
| 78 | 06 | DATA<br>This field ensures the operation of the redefine. | PIC X(1) |
| 79–88 | 04 | MRCH-VRFCN-VAL<br>The merchant verification value. This code is assigned by Visa and the acquirer to a specific merchant. | PIC X(10) |
| 89–94 | 04 | TRACE-NUM<br>The trace number. This number uniquely identifies a cardholder transaction. | PIC X(6) |
| 95–106 | 04 | REF-NUM<br>The transaction reference number. This number assists other key data elements in identifying and tracking all messages relating to a cardholder. | PIC X(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 107 | 04 | OFF-PREMISE | PIC X(1) |
| | | The off-premise indicator. This code indicates the transaction as occuring at either a remote or non-remote ATM terminal. Valid values are as follows:<br>Y = The transaction originated at an off-premise location.<br>␣ = The transaction did not originate at an off-premise location (where ␣ indicates a blank space). | |
| 108–111 | 04 | RSK | |
| | | This field contains a two-digit risk score and a two-digit reason code as supplied by Visa on request and advice transactions to the issuer. | |
| 108–109 | 06 | SCORE | PIC X(2) |
| | | The two-digit risk score. | |
| 110–111 | 06 | RSN-CDE | PIC X(2) |
| | | The two-digit reason code. | |
| 112–117 | 04 | COND | |
| | | This field contains up to three two-digit condition codes as supplied by Visa on request and advice transactions to the issuer. | |
| 112–113 | 06 | CDE1 | PIC X(2) |
| | | The first two-digit condition code. | |
| 114–115 | 06 | CER-NUM | PIC X(2) |
| | | A two-byte alphanumeric CER ID assigned to a significant CAMS event. Valid values are 0-9 and A-Z. A value of 00 indicates that no CER ID is assigned. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 116–117 | 06 | CDE3<br>The third two-digit condition code. | PIC X(2) |
| 118–123 | 04 | DEC-POSN-IND<br>The Visa currency precision indicator consists of three two-byte fields that indicate how many decimal places apply to the minor unit of currency. Participating acquirers set positions 1 and 2 to indicate how many decimal places the minor unit of currency contains for the transaction amount. Visa sets the remaining positions based on decimal places for the settlement amount and cardholder amount. | PIC X(6) |
| 124–138 | 04 | ISA-AMT<br>A code indicating the amount of assessment fees for international transactions. This code is valid only for the LAC region. The fees indicated by this code do not have any impact on the amounts sent to the Authorization (AUTH) or Router Authorization (RTAU) files. | |
| 124–138 | 06 | AMT-FEES<br>The following fields pertain to the Visa-specific International Service Assessment (ISA) program. | |
| 124–125 | 08 | FEE-TYP<br>The type of fee. This code is set to a default of 70 or left blank. | PIC 9(2) |
| 126–128 | 08 | CRNCY-CDE<br>The standard, three-digit ISO numeric currency code in which the fee amount is set. | PIC 9(3) |
| 129 | 08 | MINOR-UNIT<br>The number of implied decimal places in the amount field. | PIC 9(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 130 | 08 | AMT-SIGN<br>A code indicating whether the fee is negative or positive. Valid values are as follows.<br>C = Credit<br>D = Debit | PIC 9(1) |
| 131–138 | 08 | AMT<br>The amount of the fee. | PIC 9(8) |
| 139 | 04 | CHIP-TXN-IND<br>The chip transaction indicator. | PIC X(1) |
| 140 | 04 | RESP-SRC-RSN-CDE<br>Identifies the source of the of the field 39 response decision. | PIC X(1) |
| 141–142 | 04 | CHRGBCK-RGHTS-IND<br>Indicator used to notify the issuer of the CPS chargeback protection level. | PIC X(2) |
| 143–146 | 04 | MIS-CAS-CDE | PIC X(4) |
| 147 | 04 | PRTL-AUTH-IND<br>Partial authorization indicator. The value of 1 in this field indicates that the terminal will accept partial amount approvals. | PIC X(1) |
| 148-156 | 04 | TRAN-FEE<br>The transaction fee. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 148 | 06 | IND | PIC X(1) |
| 149-156 | 06 | AMT | PIC 9(8) |
| 157-159 | 04 | TRAN-CRNCY-CDE | PIC X(3) |
| | | A code indicating the currency code of the transaction, as received from the device or interchange. | |
| 160-167 | 04 | VISA-MONEY-XFER-DATA | |
| | | Bit map position 48, Usage 37 (Visa Money Transfer Data). | |
| 160-163 | 06 | WATCH-LIST-MGMT-VALID-CDE | PIC X(4) |
| 164-166 | 06 | WATCH-LIST-MGMT-RSLT-CDE | PIC X(3) |
| 167 | 06 | ACTVTY-CHK-RSLT | PIC X(1) |
| 168-169 | 04 | ORIG-RESP-CDE | PIC X(2) |
| | | Bit map position 44.11 (Original Response Code). Contains the value of field 39 in the original response message when a transaction is declined as a duplicate. | |
| 170-189 | 04 | ADDL-AMT | |
| | | Bit map position 54 (Additional amount). Contains Amount Type 95. | |
| 170-171 | 06 | ACCT-TYP | PIC X(2) |
| 172-173 | 06 | AMT-TYP | PIC X(2) |
| 174-176 | 06 | CRNCY-CDE | PIC X(3) |
| 177 | 06 | AMT-SIGN | PIC X(1) |
| 178-189 | 06 | AMT | PIC X(12) |
| 190-217 | 04 | ACCT-ID1 | PIC X(28) |
| | | Bit map position 102 (Account Identification 1). Contains a number identifying an account or customer relationship. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 218 | 04 | TERM-TYP | PIC X(1) |
| | | Bit map position 60.1 (Terminal Type). | |
| 219 | 04 | FEE-FUND-MSG-FUND-SRC-IND | PIC X(1) |
| | | Bit map position 48. Fee collections/funds disbursements Funds Source Indicator. | |
| 220 | 04 | SPEND-QUALIFIED-IND | PIC X(1) |
| | | Bit map position 62.25. The Spend Qualified indicator notifies the issuer or acquirer if the account has met the required spending threshold. | |

# Token B0 Switch Token (Acquirer) and Token B1 Switch Token (Issuer)—ASCII Format

The token ID for the Switch Token is B0 for Acquirer and B1 for Issuer. Both tokens use the same DDL structure; therefore these tokens are documented together.

Listed in this description is the data that is carried in the token. Because the data carried in the token varies by interchange, the information is documented for each interchange individually.

The fields in the ASCII format Switch token for acquirers (token ID B0) and issuers (token ID B1) are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

**Note:** The B0 and B1 tokens are specific to individual switch interfaces. Other BASE24 processes do not access the information in these tokens other than for transmission to the host or co-network and for logging to the TLF or PTLF.

## Interface FIIDs and Version IDs

Each interface has a unique FIID that identifies it to the BASE24 system. This value is hard-coded in the FIID field of the Switch token and is not checked against the value in the INTERFACE FIID field on screen 1 of the Interchange Configuration File (ICF) or the Enhanced Interchange Configuration File (ICFE). FIID values less than 4 characters in length are left-justified.

Also, each interface has a version ID assigned to it that identifies the current version of the interface. This value is carried in the VER or VER-ID field of the Switch token.

A summary table of the IDs for each interface B0 and B1 token format documented in this manual is provided with the binary format versions of these tokens.

## Alaska Option ISO

The following fields define the values in the B0 and B1 tokens that are used by the Alaska Option ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–40 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–40 | 02 | AOI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–14 | 04 | NETWRK-ID | PIC X(4) |
| 15 | 04 | ON-PREMISE | PIC X(1) |
| 16 | 04 | BILL-PAY | PIC X(1) |
| 17–26 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 27–29 | 04 | SWI-RESP-CDE | PIC X(3) |
| 30–31 | 04 | ACQ-FROM-ACCT | PIC X(2) |
| 32–33 | 04 | ACQ-TO-ACCT | PIC X(2) |
| 34–36 | 04 | FROM-ACCT-QUAL | PIC X(3) |
| 37–39 | 04 | TO-ACCT-QUAL | PIC X(3) |
| 40 | 04 | FULL-TRK2-DATA-PRSN | PIC X(1) |

## American Express CAPN ISO (AXCI)

The following fields define the values in the B0 and B1 tokens that are used by the American Express CAPN ISO (AXCI) interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–56 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–56 | 02 | AXCI-SWI-TKN | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–16 | 04 | APPRV-CDE | PIC X(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 17–19 | 04 | ACT-CDE | PIC X(3) |
| 20–44 | 04 | ADNL-RESP-DATA | PIC X(25) |
| 45–56 | 04 | PT-SVC-DATA-CDE | PIC X(12) |

## American Express Global Network (GNS)

The following fields define the values in the B0 and B1 tokens that are used by the American Express Global Network interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–58 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–58 | 02 | AEGN-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–22 | 04 | PT-SVC-DATA-CDE | PIC X(12) |
| 23–34 | 04 | RETRV-REF-NUM | PIC X(12) |
| 35–40 | 04 | APPRV-CDE | PIC X(6) |
| 41–43 | 04 | ACT-CDE | PIC 9(3) |
| 44–58 | 04 | AEGN-ACQ-REF-DATA | PIC X(15) |

## BankNet

The following fields define the values in the B0 and B1 tokens that are used by the BankNet interface (BNET).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–236 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9–236 | 02 | BNET-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–16 | 04 | LOCAL-TIM | PIC 9(6) |
| 17–20 | 04 | LOCAL-DAT | PIC 9(4) |
| 21–23 | 04 | ADVICE-RSN-CDE | PIC 9(3) |
| 24–26 | 04 | POS-ENTRY-MDE | PIC 9(3) |
| 27–28 | 04 | RESP-CDE | PIC 9(2) |
| 29 | 04 | CRD-VRFY-RESULT | PIC X(1) |
| 30 | 04 | VISA-XSTNG-DBT-IND | PIC X(1) |
| 31 | 04 | VISA-SVC-DVLPMT-IND | PIC X(1) |
| 32 | 04 | VISA-DEF-BILL-IND | PIC X(1) |
| 33–38 | 04 | PROC-CDE | PIC X(6) |
| 39–68 | 04 | ON-BEHALF | OCCURS 10 TIMES |
| | 06 | SVC | PIC X(2) |
| | 06 | RSLT-1 | PIC X(1) |
| 69–70 | 04 | MCHIP-PRO-IND | PIC X(2) |
| 71 | 04 | MCHIP-PROC-INFO | PIC X(1) |
| 72–80 | 04 | TRAN-FEE | PIC X(9) |
| 81–83 | 04 | E-COM-SEC-LVL-IND | PIC 9(3) |
| 84 | 04 | AVS-RESULT | PIC X(1) |
| 85 | 04 | ACCT-NUM-IND | PIC X(1) |
| 86–89 | 04 | ADVC-DETL-CDE | PIC 9(4) |
| 90–95 | 04 | AUTH-AGENT-ID-CDE | PIC X(6) |
| 96–98 | 04 | PMNT-TXN-TYP-IND | PIC X(3) |
| 99–148 | 04 | CHIP-BIT-ERR-RSLTS | OCCURS 10 TIMES |
| | 06 | CVR-TVR-ID | PIC X(1) |
| | 06 | BYTE-ID | PIC X(2) |
| | 06 | BIT-ID | PIC X(1) |
| | 06 | BIT-ERR | PIC X(1) |
| 149–150 | 04 | CRD-LVL-RSLT | PIC X(2) |
| 151–158 | 04 | MC-GTWY-TXN-IND | PIC X(8) |
| 159–164 | 04 | MC-ASGN-ID-SUB-FLD-32 | PIC X(6) |
| 165–183 | 04 | VPAN | PIC X(19) |
| 165–183 | 04 | PAYPASS-MAPPED-PAN | REDEFINES VPAN. |
| 165–183 | 04 | PAYPASS-PAN | REDEFINES VPAN. |
| 184 | 04 | RTE-IND | PIC X(1) |
| 185 | 04 | VISA-MKT-SPCFC-DATA | PIC X(1) |
| 186–196 | 04 | NTL-POS-DATA | |
| | 06 | COND-CDE | PIC 9(11) |
| 197–198 | 04 | PIN-SRV-CDE | PIC 9(2) |
| 199–200 | 04 | PIN-CAP-CDE | PIC 9(2) |
| 201–210 | 04 | AUTH-SYS-ADV-DAT-TIM | PIC 9(10) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 211–222 | 04 | FRAUD-DATA | |
| | 06 | SCORE | PIC X(3) |
| | 06 | RSN-CDE | PIC X(2) |
| | 06 | RULE-ADJ-SCORE | PIC X(3) |
| | 06 | RULE_RSN-CDE-1 | PIC X(2) |
| | 06 | RULE_RSN-CDE-2 | PIC X(2) |
| 222–224 | 04 | PMNT-INITIATION-CHAN | PIC X(2) |
| 225–236 | 04 | GRATUITY-AMT | PIC 9(12) |

## BIC ANSI

The BIC ANSI interface does not use the B0 and B1 token.

## BIC ISO

The following fields define the values in the B0 and B1 tokens that are used by the BIC ISO Interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–28 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–28 | 02 | BICI-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 21–26 | 04 | TRACE-NUM | PIC X(6) |
| 27–28 | 04 | RESP-CDE | PIC X(2) |

## Cash Station ISO

The following fields define the values in the B0 and B1 tokens that are used by the Cash Station ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–26 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–26 | 02 | CSSI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–16 | 04 | STAN | PIC X(6) |
| 17–19 | 04 | ACT-CDE | PIC X(3) |
| 20–25 | 04 | RSRVD-62-CSSI | PIC X(6) |
| 26 | 04 | OFF-PREMISE | PIC X(1) |

## Deluxe ISO

The following fields define the values in the B0 and B1 tokens that are used by the Deluxe ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–32 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–32 | 02 | DIGI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 21–26 | 04 | STAN | PIC X(6) |
| 27–28 | 04 | RESP-CDE | PIC X(2) |
| 29–31 | 04 | NETWK-ID | PIC X(3) |
| 32 | 04 | FILLER | PIC X(1) |

## DIAS

The following fields define the values in the B0 and B1 tokens that are used by the DIAS interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–40 | | TKN-DATA | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–40 | 02 | DIAS | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 21–26 | 04 | TRACE-NUM | PIC X(6) |
| 27–28 | 04 | RESP-CDE | PIC X(2) |
| 29–40 | 04 | RETRVL-REF-NUM | PIC X(12) |

## Discover ISO

The following fields define the values in the B0 and B1 tokens that are used by the Discover ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–80 | | TKN-DATA | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–80 | 02 | DSCV-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | XMIT-DAT-TIM | PIC X(10) |
| 21–26 | 04 | STAN | PIC X(6) |
| 27–28 | 04 | RESP-CDE | PIC X(2) |
| 29–41 | 04 | ADNL-RESP-DATA-DSCV | PIC X(13) |
| 42–56 | 04 | NETWK-REF-ID | PIC X(15) |
| 57–62 | 04 | PROC-CDE | PIC X(6) |
| 63–75 | 04 | POS-SRVC-DATA | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 63 | 06 | POS-TERM-ATTND-IND | PIC X(1) |
| 64 | 06 | PARTIAL-APPRV-IND | PIC X(1) |
| 65 | 06 | POS-TERM-LOC-IND | PIC X(1) |
| 66 | 06 | POS-CRDHLDR-PRSN-IND | PIC X(1) |
| 67 | 06 | POS-CRD-PRSN-IND | PIC X(1) |
| 68 | 06 | POS-CRD-CAPTR-CAP-IND | PIC X(1) |
| 69 | 06 | POS-TXN-STAT-IND | PIC X(1) |
| 70 | 06 | POS-TXN-SEC-IND | PIC X(1) |
| 71-72 | 06 | USER-FLD | PIC X(2) |
| 73 | 06 | POS-CRD-TERM-INPUT-CAP-IND | PIC X(1) |
| 74-75 | 06 | POS-AUTH-LIFE-CYC | PIC X(2) |
| 76-77 | 04 | TXN-QUAL | |
| 76 | 06 | TRK1-DATA-IND | PIC X(1) |
| 77 | 06 | TRK2-DATA-IND | PIC X(1) |
| 78-80 | 04 | PT-SVC-ENTRY-MDE | PIC X(3) |

## EPS-Net

The following fields define the values in the B0 and B1 tokens that are used by the EPS-Net interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–54 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–54 | 02 | EURO-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–16 | 04 | STAN | PIC 9(6) |
| 17–19 | 04 | POS-ENTRY-MODE | PIC 9(3) |
| 20–30 | 04 | FWD-INST-ID-CDE | PIC X(11) |
| 31–42 | 04 | RET-REF-NUM | PIC X(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 43–44 | 04 | RESP-CDE | PIC X(2) |
| 45–48 | 04 | EXP-DAT | PIC 9(4) |
| 49–54 | 04 | PROC-CDE | PIC 9(6) |

## FDR ISO

The FDR ISO interface does not use the B0 and B1 token.

## JCB ISO

The JCB ISO interface does not use the B0 and B1 token.

## LINK (LIS5)

The following fields define the values in the B0 and B1 tokens that are used by the LINK (LIS5) interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–372 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–372 | 02 | LINK-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–16 | 04 | PROC-CDE | PIC 9(6) |
| 17–22 | 04 | STAN | PIC 9(6) |
| 23–25 | 04 | POS-ENTRY-MDE | PIC 9(3) |
| 26–27 | 04 | POS-COND-CDE | PIC 9(2) |
| 28–38 | 04 | FWD-INST-ID-CDE | PIC X(11) |
| 39–40 | 04 | RESP-CDE | PIC X(2) |
| 41–42 | 04 | RVSL-RSN | PIC X(2) |
| 43 | 04 | CRD-DATA-INPUT-CAP | PIC X(1) |
| 44 | 04 | CRDHLDR-AUTH-CAP | PIC 9(1) |
| 45 | 04 | CRD-CAPTR-CAP | PIC 9(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 46 | 04 | OPERATING-ENVMT | PIC 9(1) |
| 47 | 04 | CRDHLDR-PRSN | PIC 9(1) |
| 48 | 04 | CRD-PRSN | PIC 9(1) |
| 49 | 04 | CRD-DATA-INPUT-MDE | PIC 9(1) |
| 50 | 04 | CRDHLDR-AUTH-METHOD | PIC 9(1) |
| 51 | 04 | CRDHLDR-AUTH-ENTY | PIC 9(1) |
| 52 | 04 | CRD-DATA-OUTPUT-CAP | PIC 9(1) |
| 53 | 04 | TERM-OUTPUT-CAP | PIC 9(1) |
| 54 | 04 | PIN-CAPTR-CAP | PIC X(1) |
| 55 | 04 | SOCL-DEPRIVATN-AREA | PIC X(1) |
| 56 | 04 | OUTSIDE-HOME-TERRITORY-IND | PIC X(1) |
| 57-62 | 04 | P61-RSRVD-1 | PIC X(6) |
| 63-72 | 04 | POST-CDE | PIC X(10) |
| 73-82 | 04 | P61-RSRVD-2 | PIC X(10) |
| 83-88 | 04 | ISS-TRACE-ID | PIC X(6) |
| 89-90 | 04 | PROD-TYP | PIC X(2) |
| 91-189 | 04 | BILATERAL-DISC-DATA | PIC X(99) |
| 190-229 | 04 | SENDER-PROXY | PIC X(40) |
| 230-269 | 04 | RECIPIENT-PROXY | PIC X(40) |
| 270-287 | 04 | TXN-REF | PIC X(18) |
| 288-298 | 04 | NSC-BIC | PIC X(11) |
| 299-332 | 04 | ACCT-IBAN | PIC X(34) |
| 333-372 | 04 | PMNT-REF | PIC X(40) |

## MAC MASM (MACI)

The following fields define the values in the B0 and B1 tokens that are used by the MAC MASM (MACI) interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–72 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–72 | 02 | MACI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–13 | 04 | TYP-QUAL | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 14–19 | 04 | TRACE-NUM | PIC X(6) |
| 20–21 | 04 | NETWK-ID | PIC X(2) |
| 22–23 | 04 | DENIAL-CDE | PIC X(2) |
| 24–29 | 04 | TERM-TIM | PIC X(6) |
| 30–44 | 04 | PRE-AUTH-KEY | PIC X(15) |
| 45 | 04 | ICHG-RATE-IND | PIC X(1) |
| 46–57 | 04 | MCCR-FEE-AMT | PIC X(12) |
| 58–72 | 04 | AIRLN-TCKT-NUM | PIC X(15) |

## MDS Cirrus ISO

The following fields define the values in the B0 and B1 tokens that are used by the MDS Cirrus ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–274 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–274 | 02 | MDS-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–12 | 04 | NTWK-ID-CDE | PIC X(2) |
| 13–24 | 04 | REF-NUM | PIC X(12) |
| 25–26 | 04 | RESP-CDE | PIC X(2) |
| 27–29 | 04 | POS-ENTRY-MODE | PIC 9(3) |
| 30 | 04 | TERM-LOC | PIC 9(1) |
| 31–39 | 04 | SWI-REF-NUM | PIC X(9) |
| 40–54 | 04 | AIRLINE-TCKT-NUM | PIC X(15) |
| 55–60 | 04 | PROC-CDE | PIC X(6) |
| 61–63 | 04 | SETL-SRVC-DATA | PIC X(3) |
| 64–67 | 04 | ADVC-DETL-CDE | PIC 9(4) |
| 68–73 | 04 | ON-BEHALF | OCCURS 2 TIMES |
| | 06 | SVC | PIC X(2) |
| | 06 | RSLT-1 | PIC X(1) |
| 74-85 | 04 | CRDHLDR-BILLING-AMT | PIC 9(12) |
| 86-97 | 04 | MCCR-AMT | PIC 9(12) |
| 98–105 | 04 | ICCR-AMT | PIC 9(8) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 106–111 | 04 | TIERED-MERCH-ID | PIC 9(6) |
| 112–114 | 04 | QCK-PYMNT-SRV-IND | PIC X(3) |
| 115–119 | 04 | GCMS-PROC-DAT-CYC | PIC X(5) |
| 120–124 | 04 | GCMS-ADNL-POS-DATA | PIC X(5) |
| 125–136 | 04 | GCMS-BUS-ACTVTY | PIC X(12) |
| 137–155 | 04 | GCMS-SETL-DATA | PIC X(19) |
| 156–157 | 04 | MCHIP-PRO-IND | PIC X(2) |
| 158 | 04 | MCHIP-PROC-INFO | PIC X(1) |
| 159-167 | 04 | ORIG-SWI-SERL-NUM | PIC X(9) |
| 168 | 04 | ACCT-CAT | PIC X(1) |
| 169 | 04 | PAYPASS-DEV-RQST-RESP | PIC X(1) |
| 170 | 04 | ICHG-RATE-IND | PIC X(1) |
| 171 | 04 | CVC2-PRG-IND | PIC X(1) |
| 172 | 04 | CHRGBCK-IND | PIC X(1) |
| 173-174 | 04 | PMNT-INITIATE-CHAN | PIC X(2) |
| 175 | 04 | PRMTN-CDE | PIC X(1) |
| 176-178 | 04 | USER-FLD-ACI | PIC X(3) |
| 179-187 | 04 | TRAN-FEE | PIC X(9) |
| 188 | 04 | ACCT-NUM-IND | PIC X(1) |
| 189-207 | 04 | VPAN | PIC X(19) |
| 189-207 | 04 | PAYPASS-MAPPED-PAN | REDEFINES VPAN. |
| 189-207 | 04 | PAYPASS-PAN | REDEFINES VPAN. |
| 208-257 | 04 | CHIP-BIT-ERR-RSLTS | OCCURS 10 TIMES |
| | 06 | CVR-TVR-ID | PIC X(1) |
| | 06 | BYTE-ID | PIC X(2) |
| | 06 | BIT-ID | PIC X(1) |
| | 06 | BIT-ERR | PIC X(1) |
| 258-260 | 04 | PROD-ID | PIC X(3) |
| 261-262 | 04 | BUS-APPL-ID | PIC X(2) |
| 263-274 | 04 | FRAUD-DATA | |
| | 06 | SCORE | PIC X(3) |
| | 06 | RSN-CDE | PIC X(2) |
| | 06 | RULE-ADJ-SCORE | PIC X(3) |
| | 06 | RULE-RSN-CDE-1 | PIC X(2) |
| | 06 | RULE-RSN-CDE-2 | PIC X(2) |
| 275-286 | 04 | GRATUITY-AMT | PIC 9(12) |

## Money Station (MONY)

The following fields define the values in the B0 and B1 tokens that are used by the MDS Cirrus ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–36 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–36 | 02 | MONY | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–16 | 04 | STAN | PIC X(6) |
| 17–18 | 04 | RESP-CDE | PIC X(2) |
| 19–22 | 04 | NETWK-ID | PIC X(4) |
| 23–28 | 04 | SWI-DAT | PIC X(6) |
| 29–34 | 04 | SWI-TIM | PIC X(6) |
| 35 | 04 | OFF-PREMISE | PIC X(1) |
| 36 | 04 | USER-FLD | PIC X(1) |

## MPS

The MPS interface does not use the B0 and B1 token.

## NBGC

The following fields define the values in the B0 and B1 tokens that are used by the NBGC interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–104 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9–104 | 02 | NBGC | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–14 | 04 | MSG-TYP | PIC 9(4) |
| 15–20 | 04 | PROC-CDE | PIC 9(6) |
| 21–26 | 04 | STAN | PIC 9(6) |
| 27–30 | 04 | EXP-DAT | PIC 9(4) |
| 31–33 | 04 | ACQ-CNTRY-CDE | PIC 9(3) |
| 34–36 | 04 | POS-ENTRY-MDE | PIC 9(3) |
| 37–38 | 04 | POS-COND-CDE | PIC 9(2) |
| 39–49 | 04 | FWD-INST-ID-CDE | PIC X(11) |
| 50–61 | 04 | RET-REF-NUM | PIC X(12) |
| 62–63 | 04 | RESP-CDE | PIC X(2) |
| 64–78 | 04 | CRD-ACPT-ID-CDE | PIC X(15) |
| | 04 | POS-DATA | |
| 79–89 | 06 | COND-CDE | PIC 9(11) |
| 90–91 | 06 | AUTH-LIFE-CYCLE | PIC 9(2) |
| 92–94 | 06 | CNTRY-CDE | PIC 9(3) |
| 95–104 | 06 | POSTAL-CDE | PIC X(10) |

## NPC ISO

The following fields define the values in the B0 and B1 tokens that are used by the NPC ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–46 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–46 | 02 | NPCI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–12 | 04 | AUTH-NETWORK-ID | PIC X(2) |
| 13–14 | 04 | AUTH-NETWORK-RESP | PIC X(2) |
| 15–39 | 04 | ADDL-RESP-DATA | PIC X(25) |
| 40–45 | 04 | STAN | PIC X(6) |
| 46 | 04 | USER-FLD | PIC X(1) |

# NYCE ISO

The following fields define the values in the B0 and B1 tokens that are used by the NYCE ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–106 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–106 | 02 | NYCI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 21–26 | 04 | STAN | PIC X(6) |
| 27–28 | 04 | RESP-CDE | PIC X(2) |
| 29–31 | 04 | NETWK-ID | PIC X(3) |
| 32 | 04 | OFF-PREMISE | PIC X(1) |
| 33 | 04 | BILL-PAY | PIC X(1) |
| 34–39 | 04 | PROC-CDE | PIC X(6) |
| 40 | 04 | BAL-RTRND-BY-HOST | PIC X(1) |
| 41–52 | 04 | DDA-BAL | PIC X(12) |
| 53–64 | 04 | SAV-BAL | PIC X(12) |
| 65–76 | 04 | CR-BAL | PIC X(12) |
| 77–88 | 04 | OTH-BAL | PIC X(12) |
| 89 | 04 | ISS-CAT | PIC X(1) |
| 91-92 | 04 | PT-SVC-ENTRY-MDE | PIC X(3) |
| 93-103 | 04 | NATL-PT-SVC-COND-CDE | |
| | 06 | TERM-CLASS | PIC X(3) |
| | 06 | PRSTN-DATA | PIC X(4) |
| | 06 | SEC-COND | PIC X(1) |
| | 06 | TERM-TYPE | PIC X(2) |
| | 06 | CRD-DATA-INPUT-CAP | PIC X(1) |
| 104-105 | 04 | PIN-DESCR-DATA | PIC X(2) |
| 106 | 04 | PIN-IND-DATA | PIC X(1) |

## Networks ISO

The following fields define the values in the B0 and B1 tokens that are used by the Networks ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–32 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–32 | 02 | NETI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 21–26 | 04 | STAN | PIC X(6) |
| 27–28 | 04 | RESP-CDE | PIC X(2) |
| 29–31 | 04 | NETWK-ID | PIC X(3) |
| 32 | 04 | FILLER | PIC X(1) |

## PLUS ISO

The following fields define the values in the B0 and B1 tokens that are used by the PLUS ISO interface.

| | | | |
|---|---|---|---|
| 1–176 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–176 | 02 | PISO-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–14 | 04 | NETWK-ID | PIC X(4) |
| 15–25 | 04 | FRWD-INST-ID | PIC 9(11) |
| 26–65 | 04 | PSI-REF-INFO | PIC X(40) |
| 66–76 | 04 | RCV-INST-ID | PIC 9(11) |
| 77–84 | 04 | TRAN-FEE | PIC 9(8) |
| 85–86 | 04 | RESP-CDE | PIC X(2) |
| 87–89 | 04 | ENTRY-MDE | PIC 9(3) |
| 90–95 | 04 | TRACE-NUM | PIC 9(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 96–98 | 04 | CRD-ISS-INST-CNTRY-CDE | PIC 9(3) |
| 99–101 | 04 | ACQ-INST-CNTRY-CDE | PIC 9(3) |
| 102 | 04 | DOC-INFO-IND | PIC X(1) |
| 103–108 | 04 | EXCPT-REF-NUM | PIC 9(6) |
| 109–158 | 04 | MBR-MSG-TXT | PIC X(50) |
| 159 | 04 | ISA-FEE | PIC X(1) |
| 160–162 | 04 | FEE-PGM | PIC X(3) |
| 163–174 | 04 | MCCR-FEE | PIC X(12) |
| 175–176 | 04 | CRD-LVL-PROD-ID-VAL | PIC X(2) |

## Pulse ISO

The following fields define the values in the B0 and B1 tokens that are used by the Pulse ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–162 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–162 | 02 | PULI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–22 | 04 | CHB-AMT | PIC X(12) |
| 23–32 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 33–40 | 04 | CHB-CONV-RATE | PIC X(8) |
| 41–46 | 04 | STAN | PIC X(6) |
| 47–48 | 04 | RESP-CDE | PIC X(2) |
| 49–51 | 04 | NETWK-ID | PIC X(3) |
| 52 | 04 | OFF-PREMISE | PIC X(1) |
| 53–152 | 04 | ISS-TRC-DATA | PIC X(100) |
| 153–162 | 04 | NATL-PT-SVC-COND-CDE | PIC X(10) |

## SPAN2

The following fields define the values in the B0 and B1 tokens that are used by the SPAN2 interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–96 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–96 | 02 | SPAN | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–16 | 04 | STAN | PIC X(6) |
| 17–28 | 04 | PT-SVC-DATA-CDE | PIC X(12) |
| 29–40 | 04 | RETRV-REF-NUM | PIC X(12) |
| 41–46 | 04 | APPRV-CDE | PIC X(6) |
| 47–49 | 04 | ACT-CDE | PIC X(3) |
| 50–72 | 04 | XPORT-DATA | PIC X(23) |
| 73–95 | 04 | RSRVD-62-SPAN | PIC X(23) |
| 96 | 04 | USR-FLD | PIC X(1) |

## Star ISO

The following fields define the values in the B0 and B1 tokens that are used by the Star ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–144 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–144 | 02 | STRI | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 21–26 | 04 | STAN | PIC X(6) |
| 27–28 | 04 | RESP-CDE | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 29–31 | 04 | NETWK-ID | PIC X(3) |
| 32 | 04 | BILL-PAY | PIC X(1) |
| 33 | 04 | OFF-PREMISE | PIC X(1) |
| 34–35 | 04 | COMPL-CNT | PIC X(2) |
| 36–41 | 04 | PROC-CDE | PIC X(6) |
| 42 | 04 | LAST-COMPL-IND | PIC X(1) |
| 43–57 | 04 | MRCH-ACQ-TCKT-NUM | PIC X(15) |
| 58–59 | 04 | ICHG-RATE-IND | PIC X(2) |
| 60–62 | 04 | PGM-RGSTR-ID | PIC X(3) |
| 63–74 | 04 | MCCR-AMT | PIC X(12) |
| 75 | 04 | MCCR-FEE-CR-DB-IND | PIC X(1) |
| 76–83 | 04 | MCCR-FEE-AMT | PIC X(8) |
| 84 | 04 | MCCR-FEE-RPLMT-CR-DB-IND | PIC X(1) |
| 85–92 | 04 | MCCR-FEE-RPLMT-AMT | PIC X(8) |
| 93–94 | 04 | CRD-LVL-RSLTS | PIC X(2) |
| 95 | 04 | CRDHLDR-ID-METHOD | PIC X(1) |
| 96–99 | 04 | MSG-RSN-CDE | PIC X(4) |
| 100 | 04 | RECUR-PMNT-IND | PIC X(1) |
| 101–103 | 04 | SVC-DVLP-FLD | PIC X(3) |
| 104 | 04 | CVC2-VALID-PRTCPT-IND | PIC X(1) |
| 105 | 04 | CHRGBCK-ELIGIBILITY-IND | PIC X(1) |
| 106–107 | 04 | FRAUD-SCORE-RSN-CDE | PIC X(2) |
| 108–109 | 04 | BUS-APPL-ID | PIC X(2) |
| 110 | 04 | AVS-RSLT-CDE | PIC X(1) |
| 111–118 | 04 | VISA-WATCH-LIST | |
| | 06 | MGMT-VALID-CDE | PIC X(4) |
| | 06 | MGMT-RSLTS-CDE | PIC X(3) |
| | 06 | VMT-ACTVTY-CHK-RSLT | PIC X(1) |
| 119 | 04 | REMT-PMNT-PGM-TYP-ID | PIC X(1) |
| 120-121 | 04 | PMNT-INIT-CHAN-DEV-TYP | PIC X(2) |
| 122 | 04 | MC-RATE-IND | PIC X(1) |
| 123-125 | 04 | STAR-ISSUER-IGI | PIC X(3) |
| 126-135 | 04 | NATL-PT-SVC-COND-CDE | PIC X(10) |
| 136–143 | 04 | TXN-DESCR | |
| | 06 | INTERLINK-ATTR | PIC X(1) |
| | 06 | ICHG-GRP-ID | PIC X(3) |
| | 06 | AGGR-IND | PIC X(1) |
| | 06 | AUTH-PGM | PIC X(1) |
| | 06 | TXN-SUBTYP | PIC X(1) |
| | 06 | PROD-ID | PIC X(1) |
| 144 | 04 | USER-FLD-ACI | PIC X(1) |

## SVS

The following fields define the values in the B0 and B1 tokens that are used by the SVS ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–46 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–46 | 02 | SVS | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–16 | 04 | SVS-TRAN-CDE | PIC X(6) |
| 17–28 | 04 | APPROVAL-AMT | PIC X(12) |
| 29–38 | 04 | XMIT-DAT-TIM | PIC X(10) |
| 39–44 | 04 | BAL-AMT | PIC X(6) |
| 45–46 | 04 | RESP-CDE | PIC X(2) |

## Shazam ISO

The following fields define the values in the B0 and B1 tokens that are used by the Shazam ISO interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–128 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–128 | 02 | SHZM | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 21–26 | 04 | STAN | PIC X(6) |
| 27–29 | 04 | FNCT-CDE | PIC X(3) |
| 30–32 | 04 | ACT-CDE | PIC X(3) |
| 33–35 | 04 | NETWORK-ID | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 36 | 04 | BILL-PAY | PIC X(1) |
| 37–52 | 04 | BP-ORG-NAME | PIC X(16) |
| 53–69 | 04 | BP-ACCT-NUM | PIC X(17) |
| 70 | 04 | TRAN-IND | PIC X(1) |
| 71–72 | 04 | RETRV-REF-NUM | PIC X(2) |
| 73–78 | 04 | KEY-SEQ-NUM | PIC X(6) |
| 79-127 | 04 | CRD-ACCPT-NAM-LOC | PIC X(49) |
| 128 | 04 | USER-FLD-ACI | PIC X(1) |

## ValueLink

The following fields define the values in the B0 and B1 tokens that are used by the ValueLink interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–46 | | SWI-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–46 | 02 | VLNK | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–18 | 04 | PREV-BAL | PIC 9(8) |
| 19–26 | 04 | NEW-BAL | PIC 9(8) |
| 27–30 | 04 | CRD-CLS | PIC 9(4) |
| 31–38 | 04 | CB-AMT | PIC 9(8) |
| 39–46 | 04 | SALE-AMT | PIC 9(8) |

# Visa Debit Processing Service (DPS)

The following fields define the values in the B0 and B1 tokens that are used by the Visa Debit Processing Service (DPS) interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–92 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–92 | 02 | VDPS | REDEFINES BUF |
| 9–10 | 04 | VER | PIC X(2) |
| 11–20 | 04 | SWI-TXN-DAT-TIM | PIC X(10) |
| 21–26 | 04 | STAN | PIC X(6) |
| 27–38 | 04 | RETRVL-REF-NUM | PIC X(12) |
| 39–40 | 04 | RESP-CDE | PIC X(2) |
| 41–43 | 04 | NETWK-ID | PIC X(3) |
| 44 | 04 | OFF-PREMISE | PIC X(1) |
| 45 | 04 | BILL-PAY | PIC X(1) |
| 46–48 | 04 | FEE-PGM-IND | PIC X(3) |
| 49–50 | 04 | E-COMM-GDS-IND | PIC X(2) |
| 51–60 | 04 | MRCH-VRFCN-VAL | PIC X(10) |
| 61–82 | 04 | RSK-DATA | |
| 61–72 | 06 | FLCN | |
| 61 | 08 | SCR-SRC | PIC (X) |
| 62–65 | 08 | SCR-VAL | PIC X(4) |
| 66 | 08 | RESP-CDE | PIC X(1) |
| 67–68 | 08 | RSN1 | PIC X(2) |
| 69–70 | 08 | RSN2 | PIC X(2) |
| 71–72 | 08 | RSN3 | PIC X(2) |
| 73–82 | 06 | VISA | |
| 73–74 | 08 | SCR | PIC X(2) |
| 75–76 | 08 | RSN-CDE | PIC X(2) |
| 77–78 | 08 | COND-CDE1 | PIC X(2) |
| 79–80 | 08 | COND-CDE2 | PIC X(2) |
| 81–82 | 08 | COND-CDE3 | PIC X(2) |
| 83–84 | 04 | CRD-LVL-RSLT-VAL | PIC X(2) |
| 85–92 | 04 | VISA-WATCH-LIST | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 85–88 | 06 | MGMT-VALID-CDE | PIC X(4) |
| 89–91 | 06 | MGMT-RSLT-CDE | PIC X(3) |
| 92 | 06 | VMT-ACTVTY-CHK-RSLT | PIC X(1) |

## VisaNet

The following fields define the values in the B0 and B1 tokens that are used by the VisaNet interface.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–222 | | SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–8 | 02 | FIID | PIC X(4) |
| 9–450 | 02 | BUF | PIC X(442) |
| 9–222 | 02 | VISA-SWI-TKN-DATA | REDEFINES BUF |
| 9–10 | 04 | VER-ID | PIC X(2) |
| 11–14 | 04 | NTWK-ID-CDE | PIC X(4) |
| 15–25 | 04 | ISS-INST-ID-CDE | PIC X(11) |
| 26–29 | 04 | SETL-DAT | PIC 9(4) |
| 30–40 | 04 | FWD-INST-CDE | PIC X(11) |
| 41–44 | 04 | POS-ENTRY-MDE | PIC 9 (4) |
| 45–46 | 04 | RESP-CDE | PIC X(2) |
| 47–52 | 04 | PMC-ID | PIC X(6) |
| 47–52 | 04 | USER-APR08-FLD | REDEFINES PMC-ID |
| 47–52 | 06 | DATA | PIC X(6) |
| 53–54 | 04 | MRCH-VOL-IND | PIC X(2) |
| 55 | 04 | INTERNTL-SRV-ASMNT | PIC X(1) |
| 56–57 | 04 | CRD-RSLTS | PIC X(2) |
| 58 | 04 | CCDR-IND | PIC X(1) |
| 59–61 | 04 | FEE-PGM-IND | PIC X(3) |
| 62–65 | 04 | STIP-CDE | PIC X(4) |
| 66–69 | 04 | MSG-TYP | PIC 9(4) |
| 70 | 04 | DEF-BILL-IND | PIC X(1) |
| 70 | 04 | USER-OCT02-FLD5 | REDEFINES DEF-BILL-IND |
| 70 | 06 | DATA | PIC X(1) |
| 71–76 | 04 | PROC-CDE | PIC X(6) |
| 77 | 04 | REIMB-ATTR | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 78–79 | 04 | ECOMM-GDS-IND | PIC X(2) |
| 78–79 | 04 | USER-OCT02-FLD6 | REDEFINES ECOMM-GDS-IND |
| 78–79 | 06 | DATA | PIC X(2) |
| 80 | 04 | CAVV-RSLT-CDE | PIC X(1) |
| 80 | 04 | USER-OCT02-FLD7 | REDEFINES CAVV-RSLT-CDE |
| 80 | 06 | DATA | PIC X(1) |
| 81–90 | 04 | MRCH-VRFCN-VAL | PIC X(10) |
| 91–96 | 04 | TRACE-NUM | PIC X(6) |
| 97–108 | 04 | REF-NUM | PIC X(12) |
| 109 | 04 | OFF-PREMISE | PIC X(1) |
| 110–113 | 04 | RSK | |
| 110–111 | 06 | SCORE | PIC X(2) |
| 112–113 | 06 | RSN-CDE | PIC X(2) |
| 114–119 | 04 | COND | |
| 114–115 | 06 | CDE1 | PIC X(2) |
| 116–117 | 06 | CER-NUM | PIC X(2) |
| 118–119 | 06 | CDE3 | PIC X(2) |
| 120–125 | 04 | DEC-POSN-IND | PIC X(6) |
| 126–140 | 04 | ISA-AMT | |
| 126–140 | 06 | AMT-FEES | |
| 126–127 | 08 | FEE-TYP | PIC 9(2) |
| 128–130 | 08 | CRNCY-CDE | PIC 9(3) |
| 131 | 08 | MINOR-UNIT | PIC 9(1) |
| 132 | 08 | AMT-SIGN | PIC 9(1) |
| 133–140 | 08 | AMT | PIC 9(8) |
| 141 | 04 | CHIP-TXN-IND | PIC X(1) |
| 142 | 04 | RESP-SRC-RSN-CDE | PIC X(1) |
| 143–144 | 04 | CHRGBCK-RGHTS-IND | PIC X(2) |
| 145–148 | 04 | MIS-CAS-CDE | PIC X(4) |
| 149 | 04 | PRTL-AUTH-IND | PIC X(1) |
| 150–158 | 04 | TRAN-FEE | |
| 150 | 06 | IND | PIC X(1) |
| 151–158 | 06 | AMT | PIC 9(8) |
| 159–161 | 04 | TRAN-CRNCY-CDE | PIC X(3) |
| 162-170 | 04 | VISA-MONEY-XFER-DATA | |
| 162-165 | 06 | WATCH-LIST-MGMT-VALID-CDE | PIC X(4) |
| 166-168 | 06 | WATCH-LIST-MGMT-RSLT-CDE | PIC X(3) |
| 169 | 06 | ACTIVTY-CHK-RSLT | PIC X(1) |
| 170-171 | 04 | ORIG-RESP-CDE | PIC X(2) |
| 172-191 | 04 | ADDL-AMT | |
| 172-173 | 06 | ACCT-TYP | PIC X(2) |
| 174-175 | 06 | AMT-TYP | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 176-178 | 06 | CRNCY-CDE | PIC X(3) |
| 179 | 06 | AMT-SIGN | PIC X(1) |
| 180-191 | 06 | AMT | PIC X(12) |
| 192-219 | 04 | ACCT-ID1 | PIC X(28) |
| 220 | 04 | TERM-TYP | PIC X(1) |
| 221 | 04 | FEE-FUND-MSG-FUND-SRC-IND | PIC X(1) |
| 222 | 04 | SPEND-QUALIFIED-IND | PIC X(1) |

# Token B2 EMV Request Data Token—Binary Format

The EMV Request Data token contains the thirteen minimum request data elements required for inclusion in request messages, as defined by EMV. The Device Handler process or the Interchange Interface process creates this token and adds it to the transaction message before sending it to the Authorization process.

For more information about the EMV data elements refer to the MasterCard M/Chip or the Visa Smart Debit Credit (VSDC) documentation sets or the EMVCo specification.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–80 | | EMV-RQST-TKN | |
| 1–2 | 02 | BIT-MAP | TYPE BINARY 16 SIGNED |

Indicates whether data in each of the remaining fields in the token is present or absent. The token itself is a fixed format structure, so the absence of a data item means that the appropriate field is present but that its contents are undefined.

Note that the positions of the bits within the bit map follow the ISO 8583 convention (i.e., the highest order bit represents the first field in the token, following the BIT-MAP field).

| Bit Map Position | Field Name | EMV Tag |
|---|---|---|
| 1 | USER-FLD1 | N/A |
| 2 | CRYPTO-INFO-DATA | 9F27 |
| 3 | TVR | 95 |
| 4 | ARQC | 9F26 |
| 5 | AMT-AUTH | 9F02 |
| 6 | AMT-OTHER | 9F03 |
| 7 | AIP | 82 |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| Bit Map Position | Field Name | EMV Tag |
|---|---|---|
| 8 | ATC | 9F36 |
| 9 | TERM-CNTRY-CDE | 9F1A |
| 10 | TRAN-CRNCY-CDE | 5F2A |
| 11 | TRAN-DAT | 9A |
| 12 | TRAN-TYPE | 9C |
| 13 | UNPREDICT-NUM | 9F37 |
| 16 | ISS-APPL-DATA | 9F10 |

3–4 02 USER-FLD1 PIC X(2)

Must contain binary zeros.

5 02 CRYPTO-INFO-DATA PIC X(1)

The type of cryptogram and the actions to be performed by the terminal. Valid values are shown in the table below.

In the EMV specifications, definitions that include bit positions indicate that bit 8 is the leftmost bit.

**Caution:** In TAL programming, the highest order bit is the zero bit.

| EMV Defined Bit Position | Description |
|---|---|
| 8–7 | Type of cryptogram. Valid values are as follows:<br>00 = AAC<br>01 = TC<br>10 = ARQC<br>11 = AAR |
| 6 | Reserved for future use |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 5 | Reserved for future use |
| 4 | Advice required flag. Valid values are as follows:<br>0 = Advice is not required.<br>1 = Advice is required. |
| 3–1 | The reason, advice, or referral code. Valid values are as follows:<br>000 = No information given<br>001 = Service not allowed<br>010 = PIN try limit exceeded<br>011 = Issuer authentication failed |

6–10 02 TVR PIC X(5)

The terminal verification results. This field indicates the status of the different functions as seen from the terminal. Valid values are shown in the tables below. The default for all bit settings is a value of 0.

In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.

Bit positions not listed are reserved for future use.

**Caution:** In TAL programming, the highest order bit is the zero bit.

## Byte 1

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Offline data authentication flag. Valid values are as follows:<br>0 = Offline data authentication was performed.<br>1 = Offline data authentication was not performed. |
| 7 | Offline static data authentication flag. Valid values are as follows:<br>0 = Offline static data authentication passed.<br>1 = Offline static data authentication failed. |
| 6 | Integrated circuit card (ICC) data flag. Valid values are as follows:<br>0 = ICC data is present.<br>1 = ICC data is missing. |
| 5 | Card on exception file flag. Valid values are as follows:<br>0 = Card does not appear on terminal exception file.<br>1 = Card appears on terminal exception file. |
| 4 | Offline dynamic data authentication flag. Valid values are as follows:<br>0 = Offline dynamic data authentication passed.<br>1 = Offline dynamic data authentication failed. |

### Byte 2

| EMV Defined Bit Position | Description |
|---|---|
| 8 | ICC and terminal version flag. Valid values are as follows:<br>0 = The ICC and the terminal have the same application versions.<br>1 = The ICC and the terminal have different application versions. |
| 7 | Expired application flag. Valid values are as follows:<br>0 = The application has not expired.<br>1 = The application expired. |
| 6 | Application effective flag. Valid values are as follows:<br>0 = The application is effective.<br>1 = The application is not yet effective. |
| 5 | Requested service flag. Valid values are as follows:<br>0 = The requested service is allowed for the card product.<br>1 = The requested service is not allowed for the card product. |
| 4 | New card flag. Valid values are as follows:<br>0 = The transaction was not initiated with a new card.<br>1 = The transaction was initiated with a new card. |

### Byte 3

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Cardholder verification flag. Valid values are as follows:<br>0 = Cardholder verification was successful.<br>1 = Cardholder verification was not successful. |
| 7 | Unrecognized cardholder verification method (CVM) flag. Valid values are as follows:<br>0 = The CVM was recognized.<br>1 = The CVM was not recognized. |
| 6 | PIN tries flag. Valid values are as follows:<br>0 = The PIN try limit was not exceeded.<br>1 = The PIN try limit was exceeded. |
| 5 | PIN required/PIN pad not available condition. Valid values are as follows:<br>0 = PIN entry is not required or the PIN pad is present and operable.<br>1 = PIN entry is required and the PIN pad is not present or inoperable. |
| 4 | PIN required/PIN not entered condition. Valid values are as follows:<br>0 = PIN entry is not required or the PIN pad is not present or the PIN was entered.<br>1 = PIN entry is required, PIN pad is present, PIN not entered. |
| 3 | Online PIN flag. Valid values are as follows:<br>0 = Online PIN not entered.<br>1 = Online PIN entered. |

**Byte 4**

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Floor limit flag. Valid values are as follows:<br>0 = The transaction amount does not exceed the floor limit.<br>1 = The transaction amount exceeds the floor limit. |
| 7 | Lower consecutive offline limit flag. Valid values are as follows:<br>0 = The lower consecutive offline limit was not exceeded.<br>1 = The lower consecutive offline limit was exceeded. |
| 6 | Upper consecutive offline limit flag. Valid values are as follows:<br>0 = The upper consecutive offline limit was not exceeded.<br>1 = The upper consecutive offline limit was exceeded. |
| 5 | Random selection flag. Valid values are as follows:<br>0 = The transaction was not selected at random for online processing.<br>1 = The transaction was selected at random for online processing. |
| 4 | Merchant forced online flag. Valid values are as follows:<br>0 = The merchant did not force the transaction online.<br>1 = The merchant forced the transaction online. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

**Byte 5**

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Transaction certificate data object list (TDOL) status. Valid values are as follows:<br>0 = The default TDOL was not used.<br>1 = The default TDOL was used. |
| 7 | Issuer authentication flag. Valid values are as follows:<br>0 = Issuer authentication was successful.<br>1 = Issuer authentication was not successful. |
| 6 | Script processing before final GENERATE AC command flag. Valid values are as follows:<br>0 = Script processing did not fail before final GENERATE AC command.<br>1 = Script processing failed before final GENERATE AC command. |
| 5 | Script processing after final GENERATE AC flag. Valid values are as follows:<br>0 = Script processing did not fail after final GENERATE AC command.<br>1 = Script processing failed after final GENERATE AC command. |

11–18 02 ARQC PIC X(8)

The authorization request cryptogram. The cryptogram returned by the ICC in response to the GENERATE AC command.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 19–24 | 02 | AMT-AUTH | PIC X(6) |

The authorized amount of the transaction (excluding adjustments). Data in this field is right-justified, zero-filled packed data (i.e., binary coded decimal).

| | | | |
|---|---|---|---|
| 25–30 | 02 | AMT-OTHER | PIC X(6) |

A secondary amount associated with the transaction, representing a cash-back amount. Data in this field is right-justified, zero-filled packed data (i.e., binary coded decimal).

| | | | |
|---|---|---|---|
| 31–32 | 02 | AIP | PIC X(2) |

The application interchange profile. This field indicates the capabilities of the card to support specific functions in the application. Valid values are shown in the tables below.

In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.

Bit positions not listed are reserved for future use.

**Caution:** In TAL programming, the highest order bit is the zero bit.

**Byte 1**

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Initiate flag. Valid values are as follows:<br>0 = Do not initiate.<br>1 = Initiate. |
| 7 | Offline static data authentication support flag. Valid values are as follows:<br>0 = Offline static data authentication is not supported.<br>1 = Offline static data authentication is supported. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 6 | Offline dynamic data authentication support flag. Valid values are as follows:<br>0 = Offline dynamic data authentication is not supported.<br>1 = Offline dynamic data authentication is supported. |
| 5 | Cardholder verification support flag. Valid values are as follows:<br>0 = Cardholder verification is not supported.<br>1 = Cardholder verification is supported. |
| 4 | Terminal risk management support flag. Valid values are as follows:<br>0 = Terminal risk management will not be performed.<br>1 = Terminal risk management will be performed. |
| 3 | Issuer authentication support flag. Valid values are as follows:<br>0 = Issuer authentication is not supported<br>1 = Issuer authentication is supported. |

**Byte 2**

All bits in byte 2 are reserved for future use.

33–34 02 ATC PIC X(2)

The application transaction counter. The application on the chip maintains and increments this counter.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 35–36 | 02 | TERM-CNTRY-CDE | PIC X(2) |
| | | A code indicating the country of the terminal, according to the ISO 3166 standard, ***Codes for the Representation of Names of Countries.*** Data in this field is right-justified, zero-filled packed data (i.e., binary coded decimal). | |
| 37–38 | 02 | TRAN-CRNCY-CDE | PIC X(2) |
| | | A code indicating the currency code of the transaction, as received from the device or interchange, according to the ISO 4217 standard, ***Codes for the Representation of Currencies and Funds.*** Data in this field is right-justified, zero-filled packed data (i.e., binary coded decimal). | |
| 39–41 | 02 | TRAN-DAT | PIC X(3) |
| | | The local date (in YYMMDD format) that the transaction was authorized. Data in this field is stored as packed data (i.e., binary coded decimal). | |
| 42 | 02 | TRAN-TYPE | PIC X(1) |
| | | A code indicating the type of financial transaction, represented by the first two digits of the processing code from the 1987 ISO 8583 standard, ***Bank Card Originated Messages—Interchange Message Specifications—Content for Financial Transactions.*** Data in this field is stored as packed data (i.e., binary coded decimal). | |
| 43–46 | 02 | UNPREDICT-NUM | PIC X(4) |
| | | An unpredictable number used to provide variability and uniqueness to the generation of a cryptogram. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 47–48 | 02 | ISS-APPL-DATA-LGTH<br>Indicates the length of the issuer application data in the following field. The ASCII and binary versions of the token must contain the same value in this field. The ASCII version of the token must contain the decimal (not hexadecimal) representation of the length value. | TYPE BINARY 16 SIGNED |
| 49–80 | 02 | ISS-APPL-DATA<br>The proprietary issuer application data for transmission to the issuer in an online transaction. The data is left-justified and padded to the right with binary zeroes.<br><br>BASE24 currently supports the following definitions for issuer application data. For more information on these fields, refer to DDL documentation or the individual card scheme documentation. | PIC X(32) |
| | 02 | AEGN-APPL-DATA<br>The American Express Global Network (AEGN) definition of the issuer application data (IAD). | REDEFINES ISS-APPL-DATA |
| 49 | 04 | LGTH<br>Length of the binary representation of the following data. The ASCII and binary versions of the token must contain the same value in this field. | PIC X(1) |
| 50 | 04 | DERIV-KEY-INDEX<br>The derivation key index. This value identifies to the issuer the derivation key required to derive the card's unique keys to be used to perform on-line card and issuer authentication. The derivation key index is not used by the card. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 51 | 04 | CRYPTO-VER-NUM | PIC X(1) |

The cryptogram version number. This value indicates the version of the TC/AAC/ARQC algorithm used by the application. Values are assigned by card schemes. Currently the only supported value is 10.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 52–55 | 04 | CRD-VRFY-RSLTS | PIC X(4) |

The card verification results. The contents of this field indicate the exception conditions that occurred during card risk management, as shown below.

In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.

Bit positions not listed are reserved for future use.

**Caution:** In TAL programming, the highest order bit is the zero bit.

**Byte 1**

Length Indicator

**Byte 2**

| EMV Defined Bit Position | Description |
|---|---|
| 8–7 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in second GENERATE AC<br>01 = TC returned in second GENERATE AC<br>10 = Second GENERATE AC not requested<br>11 = Reserved for future use |

| EMV Defined Bit Position | Description |
|---|---|
| 6–5 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in first GENERATE AC<br>01 = TC returned in first GENERATE AC<br>10 = ARQC returned in first GENERATE AC<br>11 = Reserved for future use |
| 4 | Issuer authentication failure flag. Valid values are as follows:<br>0 = Issuer authentication did not fail.<br>1 = Issuer authentication failed. |
| 3 | Off-line PIN verification performed. Valid values are as follows:<br>0 = Off-line PIN verification was not performed.<br>1 = Off-line PIN verification was performed. |
| 2 | Off-line PIN verification failed. Valid values are as follows:<br>0 = Off-line PIN verification did not fail.<br>1 = Off-line PIN verification failed. |
| 1 | Unable to go on-line. Valid values are as follows:<br>0 = Able to go on-line.<br>1 = Unable to go on-line |

### Byte 3

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Last on-line transaction not completed. Valid values are as follows:<br>0 = Last on-line transaction completed.<br>1 = Last on-line transaction did not complete. |
| 7 | PIN try limit exceeded. Valid values are as follows:<br>0 = PIN try limit was not exceeded.<br>1 = PIN try limit exceeded. |
| 6 | Exceeded velocity checking counters. Valid values are as follows:<br>0 = Velocity checking counters were not exceeded.<br>1 = Velocity checking counters were exceeded. |
| 5 | New card flag. Valid values are as follows:<br>0 = New card not used to initiate the transaction.<br>1 = New card used to initiate the transaction. |
| 4 | Issuer authentication failure on last on-line transaction. Valid values are as follows:<br>0 = Issuer authentication did not fail on last on-line transaction.<br>1 = Issuer authentication failed on last on-line transaction. |

| EMV Defined Bit Position | Description |
|---|---|
| 3 | Issuer authentication not performed after on-line authorization. Valid values are as follows:<br>0 = Issuer authentication performed after on-line authorization.<br>1 = Issuer authentication not performed after on-line authorization. |
| 2 | Application blocked by card because PIN try limit exceeded. Valid values are as follows:<br>0 = Application not blocked by card because PIN try limit exceeded.<br>1 = Application blocked by card because PIN try limit exceeded. |
| 1 | Static data authentication failed on last transaction and transaction declined off-line. Valid values are as follows:<br>0 = Static data authentication did not fail on the last transaction and transaction was not declined off-line.<br>1 = Static data authentication failed on the last transaction and transaction was declined off-line. |

**Byte 4**

| EMV Defined Bit Position | Description |
|---|---|
| 8–5 | Number of issuer script commands containing secure messaging processed on last transaction. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 4 | Issuer script processing failed on last transaction.<br>0 = Issuer script processing did not fail on last transaction.<br>1 = Issuer script processing failed on last transaction. |
| 3 | Dynamic data authentication failed on last transaction and transaction declined offline. Valid values are as follows:<br>0 = Dynamic data authentication did not fail on the last transaction and transaction was not declined off-line.<br>1 = Dynamic data authentication failed on the last transaction and transaction was declined off-line. |
| 2 | Dynamic data authentication performed. Valid values are as follows:<br>0 = Dynamic data authentication was not performed.<br>1 = Dynamic data authentication was performed. |
| 1 | Reserved for future use. |

56–80 04 INFO PIC X(25)

This field contains the issuer discretionary data.

---

02 CCD-A-APPL-DATA REDEFINES ISS-APPL-DATA

Contains Format A of the EMV Common Core Definition of the issuer application data.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 49 | 04 | LGTH<br>Length of the binary representation of the following data. The ASCII and binary versions of the token must contain the same value in this field. | PIC X |
| 50 | 04 | COMMON-CORE-ID<br>The first four bits of the Common Core IAD Format Code and the second four bits of the Common Core Cryptogram Version. Valid value is A5. | PIC X |
| 51 | 04 | DERIV-KEY-INDEX<br>The derivation key index. This value identifies to the issuer the derivation key required to derive the card's unique DEA keys to be used to perform on-line card and issuer authentication. The derivation key index is not used by the card. | PIC X |
| 52–56 | 04 | CRD-VRFY-RSLTS<br>The card verification results. The contents of this field indicate the exception conditions that occurred during card risk management, as shown below.<br>In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.<br>**Caution:** In TAL programming, the highest order bit is the zero bit. | PIC X(5) |

### Byte 1

| EMV Defined Bit Position | Description |
|---|---|
| 8–7 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in second GENERATE AC<br>01 = TC returned in second GENERATE AC<br>10 = Second GENERATE AC not requested<br>11 = Reserved for future use |
| 6–5 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in first GENERATE AC<br>01 = TC returned in first GENERATE AC<br>10 = ARQC returned in first GENERATE AC<br>11 = Reserved for future use |
| 4 | CDA performed. Valid values are as follows:<br>0 = CDA was not performed.<br>1 = CDA was performed. |
| 3 | Offline DDA performed. Valid values are as follows:<br>0 = Offline DDA was not performed.<br>1 = Offline DDA was performed. |
| 2 | Issuer authentication not performed. Valid values are as follows:<br>0 = Issuer authentication was performed.<br>1 = Issuer authentication was not performed. |

| EMV Defined Bit Position | Description |
|---|---|
| 1 | Issuer authentication failed. Valid values are as follows:<br>0 = Issuer authentication did not fail.<br>1 = Issuer authentication failed. |

**Byte 2**

| EMV Defined Bit Position | Description |
|---|---|
| 8–5 | Right nibble of PIN Try Counter. |
| 4 | Offline PIN verification performed. Valid values are as follows:<br>0 = Offline PIN verification was not performed.<br>1 = Offline PIN verification was performed. |
| 3 | Offline PIN verification performed and PIN not successfully verified. Valid values are as follows:<br>0 = Offline PIN verification performed and PIN was successfully verified.<br>1 = Offline PIN verification performed and PIN was not successfully verified. |
| 2 | PIN try limit exceeded. Valid values are as follows:<br>0 = PIN try limit was not exceeded.<br>1 = PIN try limit was exceeded. |
| 1 | Last online transaction not completed. Valid values are as follows:<br>0 = Last online transaction completed.<br>1 = Last online transaction was not completed. |

### Byte 3

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Lower offline transaction count limit exceeded. Valid values are as follows:<br>0 = The lower offline transaction count limit was not exceeded.<br>1 = The lower offline transaction count limit was exceeded. |
| 7 | Upper offline transaction count limit exceeded. Valid values are as follows:<br>0 = The upper offline transaction count limit was not exceeded.<br>1 = The upper offline transaction count limit was exceeded. |
| 6 | Lower cumulative offline amount limit exceeded. Valid values are as follows:<br>0 = The lower cumulative offline amount limit was not exceeded.<br>1 = The lower cumulative offline amount limit was exceeded. |
| 5 | Upper cumulative offline amount limit exceeded. Valid values are as follows:<br>0 = The upper cumulative offline amount limit was not exceeded.<br>1 = The upper cumulative offline amount limit was exceeded. |
| 4 | Issuer-discretionary bit 1. |
| 3 | Issuer-discretionary bit 2. |
| 2 | Issuer-discretionary bit 3. |
| 1 | Issuer-discretionary bit 4. |

### Byte 4

| EMV Defined Bit Position | Description |
|---|---|
| 8–5 | Right nibble of Script Counter. |
| 4 | Issuer script processing failed. Valid values are as follows:<br>0 = Issuer script processing did not fail.<br>1 = Issuer script processing failed. |
| 3 | Offline data authentication failed on previous transaction. Valid values are as follows:<br>0 = Offline data authentication did not fail on previous transaction.<br>1 = Offline data authentication failed on previous transaction. |
| 2 | Go online on next transaction was set. Valid values are as follows:<br>0 = Go online on next transaction was not set.<br>1 = Go online on next transaction was set. |
| 1 | Unable to go online. Valid values are as follows:<br>0 = The transaction was able to go online.<br>1 = The transaction was not able to go online. |

### Byte 5

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Reserved for future use. |
| 7 | Reserved for future use. |
| 6 | Reserved for future use. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 5 | Reserved for future use. |
| 4 | Reserved for future use. |
| 3 | Reserved for future use. |
| 2 | Reserved for future use. |
| 1 | Reserved for future use. |

57–64 04 COUNTERS PIC X(8)

The contents of this field are at the discretion of the payment system.

65 04 ISS-DISCR-DATA-LGTH PIC X

The length of the binary representation of the data that follows. The ASCII and binary versions of the token must contain the same value in this field.

66–80 04 ISS-DISCR-DATA PIC X(15)

This field contains the issuer discretionary data.

---

02 MCHIP4-APPL-DATA REDEFINES ISS-APPL-DATA

Contains the MasterCard/Europay M/Chip 4 definition of the issuer application data.

49 04 DERIV-KEY-INDEX PIC X(1)

The derivation key index. This value identifies to the issuer the derivation key required to derive the card's unique DEA keys to be used to perform on-line card and issuer authentication. The derivation key index is not used by the card.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 50 | 04 | CRYPTO-VER-NUM | PIC X(1) |

The cryptogram version number. This value indicates the version of the TC/AAC/ARQC algorithm used by the application. Currently the supported values are 10, 11, 12, 13, 14, and 15.

| | | | |
|---|---|---|---|
| 51–56 | 04 | CRD-VRFY-RSLTS | PIC X(6) |

The card verification results. The contents of this field indicate the exception conditions that occurred during card risk management, as shown below.

In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.

**Caution:** In TAL programming, the highest order bit is the zero bit.

**Byte 1**

| EMV Defined Bit Position | Description |
|---|---|
| 8–7 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in second GENERATE AC<br>01 = TC returned in second GENERATE AC<br>10 = Second GENERATE AC not requested<br>11 = Reserved for future use |

| EMV Defined Bit Position | Description |
|---|---|
| 6–5 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in first GENERATE AC<br>01 = TC returned in first GENERATE AC<br>10 = ARQC returned in first GENERATE AC<br>11 = Reserved for future use |
| 4 | Reserved for future use. |
| 3 | Offline PIN verification flag. Valid values are as follows:<br>0 = Offline PIN verification was not successful.<br>1 = Offline PIN verification was successful. |
| 2 | Offine encrypted PIN verification flag. Valid values are as follows:<br>0 = Offline encrypted PIN verification was not successful.<br>1 = Offline encrypted PIN verification was successful. |
| 1 | Offline PIN verification successful. Valid values are as follows:<br>0 = Offline PIN verification was not successful.<br>1 = Offline PIN verification was successful. |

## Byte 2

| EMV Defined Bit Position | Description |
|---|---|
| 8 | DDA returned. Valid values are as follows:<br>0 = DDA was not returned.<br>1 = DDA was returned. |
| 7 | Combined DDA/AC generation returned in first GENERATE AC. Valid values are as follows:<br>0 = The combined DDA/AC generation was not returned in the first GENERATE AC.<br>1 = The combined DDA/AC generation was returned in the first GENERATE AC. |
| 6 | Combined DDA/AC generation returned in second GENERATE AC. Valid values are as follows:<br>0 = The combined DDA/AC generation was not returned in the second GENERATE AC.<br>1 = The combined DDA/AC generation was returned in the second GENERATE AC. |
| 5 | Issuer authentication performed. Valid values are as follows:<br>0 = Issuer authentication was not performed.<br>1 = Issuer authentication was performed. |

| EMV Defined Bit Position | Description |
|---|---|
| 4 | Card risk management skipped on CAT3. Valid values are as follows:<br>0 = Card risk management was not skipped on CAT3.<br>1 = Card risk management was skipped on CAT3. |
| 3 | Reserved for future use. |
| 2 | Reserved for future use. |
| 1 | Reserved for future use. |

### Byte 3

| EMV Defined Bit Position | Description |
|---|---|
| 8–5 | Right nibble of Script Counter. |
| 4–1 | Right nibble of PIN Try Counter. |

### Byte 4
### Current transaction

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Reserved for future use. |
| 7 | Unable to go online. Valid values are as follows:<br>0 = The transaction was able to go online.<br>1 = The transaction was not able to go online. |

| EMV Defined Bit Position | Description |
|---|---|
| 6 | Offline PIN verification not performed. Valid values are as follows:<br>0 = Offline PIN verification was performed.<br>1 = Offline PIN verification was not performed. |
| 5 | Offline PIN verification failed. Valid values are as follows:<br>0 = Offline PIN verification did not fail.<br>1 = Offline PIN verification failed. |
| 4 | PTL exceeded. Valid values are as follows:<br>0 = PTL was not exceeded.<br>1 = PTL was exceeded. |
| 3 | International transaction. Valid values are as follows:<br>0 = The current transaction is not an international transaction.<br>1 = The current transaction is an international transaction. |
| 2 | Domestic transaction. Valid values are as follows:<br>0 = The current transaction is not a domestic transaction.<br>1 = The current transaction is a domestic transaction. |
| 1 | Terminal erroneously considers offline PIN OK. Valid values are as follows:<br>0 = Terminal does not erroneously consider offline PIN OK.<br>1 = Terminal erroneously considers offline PIN OK. |

## Byte 5
## Current plus last online transaction

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Lower consecutive offline limit exceeded. Valid values are as follows:<br>0 = The lower consecutive offline limit was not exceeded.<br>1 = The lower consecutive offline limit was exceeded. |
| 7 | Upper consecutive offline limit exceeded. Valid values are as follows:<br>0 = The upper consecutive offline limit was not exceeded.<br>1 = The upper consecutive offline limit was exceeded. |
| 6 | Lower cumulative offline limit exceeded. Valid values are as follows:<br>0 = The lower cumulative offline limit was not exceeded.<br>1 = The lower cumulative offline limit was exceeded. |
| 5 | Upper cumulative offline limit exceeded. Valid values are as follows:<br>0 = The upper cumulative offline limit was not exceeded.<br>1 = The upper cumulative offline limit was exceeded. |
| 4 | Go online on next transaction was set. Valid values are as follows:<br>0 = Go online on next transaction was not set.<br>1 = Go online on next transaction was set. |

| EMV Defined Bit Position | Description |
|---|---|
| 3 | Issuer authentication failed. Valid values are as follows:<br>0 = Issuer authentication did not fail.<br>1 = Issuer authentication failed. |
| 2 | Script received. Valid values are as follows:<br>0 = The script was not received.<br>1 = The script was received. |
| 1 | Script failed. Valid values are as follows:<br>0 = The script did not fail.<br>1 = The script failed. |

### Byte 6
### Current transaction

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Reserved for future use. |
| 7 | Reserved for future use. |
| 6 | Reserved for future use. |
| 5 | Reserved for future use. |
| 4 | Reserved for future use. |
| 3 | Reserved for future use. |
| 2 | Match found in additional check table. Valid values are as follows:<br>0 = No match not found in additional check table.<br>1 = Match found in additional check table. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 1 | No match found in additional check table. Valid values are as follows:<br>0 = Match found in additional check table.<br>1 = No match found in additional check table. |

57–58 04 DAC PIC X(2)

The Dynamic Authentication Code, or two leftmost bytes of the ICC Dynamic Number. This value can be used to prove that the terminal correctly performed static or dynamic data authentication.

59–66 04 CNTR PIC X(8)

This field contains plain text or encrypted counter information.

67–80 04 INFO PIC X(14)

This field contains the issuer discretionary data.

---

02 MCPA-APPL-DATA REDEFINES ISS-APPL-DATA

The MasterCard/Europay (MCPA) M/Chip 2.1 definition of the issuer application data.

49 04 DERIV-KEY-INDEX PIC X(1)

The derivation key index. This value identifies to the issuer the derivation key required to derive the card's unique DEA keys to be used to perform on-line card and issuer authentication. The derivation key index is not used by the card.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 50 | 04 | CRYPTO-VER-NUM | PIC X(1) |

The cryptogram version number. This value indicates the version of the TC/AAC/ARQC algorithm used by the application. Currently the only supported value is 0*x*, where *x* represents any hexadecimal digit.

| | | | |
|---|---|---|---|
| 51–54 | 04 | CRD-VRFY-RSLTS | PIC X(4) |

The card verification results. The contents of this field indicate the exception conditions that occurred during card risk management, as shown below.

In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.

**Caution:** In TAL programming, the highest order bit is the zero bit.

**Byte 1**

Length Indicator

**Byte 2**

| EMV Defined Bit Position | Description |
|---|---|
| 8–7 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in second GENERATE AC<br>01 = TC returned in second GENERATE AC<br>10 = Second GENERATE AC not requested<br>11 = Reserved for future use |

| EMV Defined Bit Position | Description |
|---|---|
| 6–5 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in first GENERATE AC<br>01 = TC returned in first GENERATE AC<br>10 = ARQC returned in first GENERATE AC<br>11 = Reserved for future use |
| 4 | Issuer authentication failure flag. Valid values are as follows:<br>0 = Issuer authentication did not fail.<br>1 = Issuer authentication failed. |
| 3 | Off-line PIN verification performed. Valid values are as follows:<br>0 = Off-line PIN verification was not performed.<br>1 = Off-line PIN verification was performed. |
| 2 | Off-line PIN verification failed. Valid values are as follows:<br>0 = Off-line PIN verification did not fail.<br>1 = Off-line PIN verification failed. |
| 1 | Unable to go on-line. Valid values are as follows:<br>0 = Able to go on-line.<br>1 = Unable to go on-line |

### Byte 3

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Last on-line transaction not completed. Valid values are as follows:<br>0 = Last on-line transaction completed.<br>1 = Last on-line transaction did not complete. |
| 7 | PIN try limit exceeded. Valid values are as follows:<br>0 = PIN try limit was not exceeded.<br>1 = PIN try limit exceeded. |
| 6 | Exceeded velocity checking counters. Valid values are as follows:<br>0 = Velocity checking counters were not exceeded.<br>1 = Velocity checking counters were exceeded. |
| 5 | New card flag. Valid values are as follows:<br>0 = New card not used to initiate the transaction.<br>1 = New card used to initiate the transaction. |
| 4 | Issuer authentication failure on last on-line transaction. Valid values are as follows:<br>0 = Issuer authentication did not fail on last on-line transaction.<br>1 = Issuer authentication failed on last on-line transaction. |

| EMV Defined Bit Position | Description |
|---|---|
| 3 | Issuer authentication not performed after on-line authorization. Valid values are as follows:<br>0 = Issuer authentication performed after on-line authorization.<br>1 = Issuer authentication not performed after on-line authorization. |
| 2 | Application blocked by card because PIN try limit exceeded. Valid values are as follows:<br>0 = Application not blocked by card because PIN try limit exceeded.<br>1 = Application blocked by card because PIN try limit exceeded. |
| 1 | Static data authentication failed on last transaction and transaction declined off-line. Valid values are as follows:<br>0 = Static data authentication did not fail on the last transaction and transaction was not declined off-line.<br>1 = Static data authentication failed on the last transaction and transaction was declined off-line. |

**Byte 4**

| EMV Defined Bit Position | Description |
|---|---|
| 8–6 | Number of issuer script commands containing secure messaging processed on last transaction. |
| 5 | Reserved for future use. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 4 | Issuer script processing failed on last transaction. Valid values are as follows:<br>0 = Issuer script processing did not fail on last transaction.<br>1 = Issuer script processing failed on last transaction. |
| 3 | Lower consecutive offline limit exceeded. Valid values are as follows:<br>0 = The lower consecutive offline limit was not exceeded.<br>1 = The lower consecutive offline limit was exceeded. |
| 2 | Upper consecutive offline limit exceeded. Valid values are as follows:<br>0 = The upper consecutive offline limit was not exceeded.<br>1 = The upper consecutive offline limit was exceeded. |
| 1 | Reserved for future use. |

55–56 04 DAC PIC X(2)

The Dynamic Authentication Code, or two leftmost bytes of the ICC Dynamic Number. This value can be used to prove that the terminal correctly performed static or dynamic data authentication.

57–80 04 INFO PIC X(24)

This field contains the issuer discretionary data.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | VISA-APPL-DATA REDEFINES ISS-APPL-DATA<br>The Visa/UKIS definition of the issuer application data. | |
| 49 | 04 | LGTH<br>Length of the binary representation of the following data. The ASCII and binary versions of the token must contain the same value in this field. | PIC X(1) |
| 50 | 04 | DERIV-KEY-INDEX<br>The derivation key index. This value identifies to the issuer the derivation key required to derive the card's unique DEA keys to be used to perform on-line card and issuer authentication. The derivation key index is not used by the card. | PIC X(1) |
| 51 | 04 | CRYPTO-VER-NUM<br>The cryptogram version number. This value indicates the version of the TC/AAC/ARQC algorithm used by the application. Values are assigned by card schemes. Valid values are as follows:<br>0A = Decimal 10<br>0E = Decimal 14<br>11 = Decimal 17 | PIC X(1) |
| 52–55 | 04 | CRD-VRFY-RSLTS<br>The card verification results. The contents of this field indicate the exception conditions that occurred during card risk management, as shown below.<br>In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.<br>Bit positions not listed are reserved for future use.<br>**Caution:** In TAL programming, the highest order bit is the zero bit. | PIC X(4) |

### Byte 1

Length Indicator

### Byte 2

| EMV Defined Bit Position | Description |
|---|---|
| 8–7 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in second GENERATE AC<br>01 = TC returned in second GENERATE AC<br>10 = Second GENERATE AC not requested<br>11 = Reserved for future use |
| 6–5 | Type of cryptogram. Valid values are as follows:<br>00 = AAC returned in first GENERATE AC<br>01 = TC returned in first GENERATE AC<br>10 = ARQC returned in first GENERATE AC<br>11 = Reserved for future use |
| 4 | Issuer authentication failure flag. Valid values are as follows:<br>0 = Issuer authentication did not fail.<br>1 = Issuer authentication failed. |
| 3 | Off-line PIN verification performed. Valid values are as follows:<br>0 = Off-line PIN verification was not performed.<br>1 = Off-line PIN verification was performed. |

| EMV Defined Bit Position | Description |
|---|---|
| 2 | Off-line PIN verification failed. Valid values are as follows:<br>0 = Off-line PIN verification did not fail.<br>1 = Off-line PIN verification failed. |
| 1 | Unable to go on-line. Valid values are as follows:<br>0 = Able to go on-line.<br>1 = Unable to go on-line |

**Byte 3**

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Last on-line transaction not completed. Valid values are as follows:<br>0 = Last on-line transaction completed.<br>1 = Last on-line transaction did not complete. |
| 7 | PIN try limit exceeded. Valid values are as follows:<br>0 = PIN try limit was not exceeded.<br>1 = PIN try limit exceeded. |
| 6 | Exceeded velocity checking counters. Valid values are as follows:<br>0 = Velocity checking counters were not exceeded.<br>1 = Velocity checking counters were exceeded. |

| EMV Defined Bit Position | Description |
|---|---|
| 5 | New card flag. Valid values are as follows:<br>0 = New card not used to initiate the transaction.<br>1 = New card used to initiate the transaction. |
| 4 | Issuer authentication failure on last on-line transaction. Valid values are as follows:<br>0 = Issuer authentication did not fail on last on-line transaction.<br>1 = Issuer authentication failed on last on-line transaction. |
| 3 | Issuer authentication not performed after on-line authorization. Valid values are as follows:<br>0 = Issuer authentication performed after on-line authorization.<br>1 = Issuer authentication not performed after on-line authorization. |
| 2 | Application blocked by card because PIN try limit exceeded. Valid values are as follows:<br>0 = Application not blocked by card because PIN try limit exceeded.<br>1 = Application blocked by card because PIN try limit exceeded. |

| EMV Defined Bit Position | Description |
|---|---|
| 1 | Static data authentication failed on last transaction and transaction declined off-line. Valid values are as follows:<br>0 = Static data authentication did not fail on the last transaction and transaction was not declined off-line.<br>1 = Static data authentication failed on the last transaction and transaction was declined off-line. |

**Byte 4**

| EMV Defined Bit Position | Description |
|---|---|
| 8–5 | Number of issuer script commands containing secure messaging processed on last transaction. |
| 4 | Issuer script processing failed on last transaction.<br>0 = Issuer script processing did not fail on last transaction.<br>1 = Issuer script processing failed on last transaction. |
| 3 | Dynamic data authentication failed on last transaction and transaction declined offline. Valid values are as follows:<br>0 = Dynamic data authentication did not fail on the last transaction and transaction was not declined off-line.<br>1 = Dynamic data authentication failed on the last transaction and transaction was declined off-line. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 2 | Dynamic data authentication performed. Valid values are as follows:<br>0 = Dynamic data authentication was not performed.<br>1 = Dynamic data authentication was performed. |
| 1 | Reserved for future use. |

| | | | |
|---|---|---|---|
| 56–80 | 04 | INFO | PIC X(25) |

This field contains the issuer discretionary data.

# Token B2 EMV Request Data Token—ASCII Format

The fields in the ASCII format EMV Request Data token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–158 | | EMV-RQST-TKNX | |
| 1–4 | 02 | BIT-MAP | PIC X(4) |
| 5–8 | 02 | USER-FLD1 | PIC X(4) |
| 9–10 | 02 | CRYPTO-INFO-DATA | PIC X(2) |
| 11–20 | 02 | TVR | PIC X(10) |
| 21–36 | 02 | ARQC | PIC X(16) |
| 37–48 | 02 | AMT-AUTH | PIC X(12) |
| 49–60 | 02 | AMT-OTHER | PIC X(12) |
| 61–64 | 02 | AIP | PIC X(4) |
| 65–68 | 02 | ATC | PIC X(4) |
| 69–71 | 02 | TERM-CNTRY-CDE | PIC X(3) |
| 72–74 | 02 | TRAN-CRNCY-CDE | PIC X(3) |
| 75–80 | 02 | TRAN-DAT | PIC X(6) |
| 81–82 | 02 | TRAN-TYPE | PIC X(2) |
| 83–90 | 02 | UNPREDICT-NUM | PIC X(8) |
| 91–94 | 02 | ISS-APPL-DATA-LGTH | PIC X(4) |
| 95–158 | 02 | ISS-APPL-DATA | PIC X(64) |
| | 02 | AEGN-APPL-DATA | REDEFINES ISS-APPL-DATA |
| 95 | 04 | LGTH | PIC X |
| 96 | 04 | DERIV-KEY-INDEX | PIC X |
| 97 | 04 | CRYPTO-VER-NUM | PIC X |
| 98–101 | 04 | CRD-VRFY-RSLTS | PIC X(4) |
| 102–158 | 04 | INFO | PIC X(57) |
| | 02 | CCD-A-APPL-DATA | REDEFINES ISS-APPL-DATA |
| 95–96 | 04 | LGTH | PIC X(2) |
| 97–98 | 04 | COMMON-CORE-ID | PIC X(2) |
| 99–100 | 04 | DERIV-KEY-INDEX | PIC X(2) |
| 101–110 | 04 | CRD-VRFY-RSLTS | PIC X(10) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 111–126 | 04 | COUNTERS | PIC X(16) |
| 127–128 | 04 | ISS-DISCR-DATA-LGTH | PIC X(2) |
| 129–158 | 04 | ISS-DISCR-DATA | PIC X(30) |
| | 02 | MCHIP4-APPL-DATA | REDEFINES ISS-APPL-DATA |
| 95–96 | 04 | DERIV-KEY-INDEX | PIC X(2) |
| 97–98 | 04 | CRYPTO-VER-NUM | PIC X(2) |
| 99–110 | 04 | CRD-VRFY-RSLTS | PIC X(12) |
| 111–114 | 04 | DAC | PIC X(4) |
| 115–130 | 04 | CNTR | PIC X(16) |
| 131–158 | 04 | INFO | PIC X(28) |
| | 02 | MCPA-APPL-DATA | REDEFINES ISS-APPL-DATA |
| 95–96 | 04 | DERIV-KEY-INDEX | PIC X(2) |
| 97–98 | 04 | CRYPTO-VER-NUM | PIC X(2) |
| 99–106 | 04 | CRD-VRFY-RSLTS | PIC X(8) |
| 107–110 | 04 | DAC | PIC X(4) |
| 111–158 | 04 | INFO | PIC X(48) |
| | 02 | VISA-APPL-DATA | REDEFINES ISS-APPL-DATA |
| 95–96 | 04 | LGTH | PIC X(2) |
| 97–98 | 04 | DERIV-KEY-INDEX | PIC X(2) |
| 99–100 | 04 | CRYPTO-VER-NUM | PIC X(2) |
| 101–108 | 04 | CRD-VRFY-RSLTS | PIC X(8) |
| 109–158 | 04 | INFO | PIC X(50) |

# Token B3 EMV Discretionary Data Token—Binary Format

The EMV Discretionary Request Data token consists of EMV-related data that is not required for authorization. However, each data element is supported by more than one EMV-compliant interface and, therefore, can be mapped between interfaces by BASE24.

For more information about the EMV data elements refer to the MasterCard M/Chip or the Visa Smart Debit Credit (VSDC) documentation sets or the EMVCo specification.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–44 | | EMV-DISCR-TKN | |
| 1–2 | 02 | BIT-MAP | TYPE BINARY 16 SIGNED |

Indicates whether data in each of the remaining fields in the token is present or absent. The token itself is a fixed format structure, so the absence of a data item means that the appropriate field is present but that its contents are undefined.

Note that the positions of the bits within the bit map follow the ISO 8583 convention (i.e., the highest order bit represents the first field in the token, following the BIT-MAP field). There are 16 bits in the BIT-MAP field, but only 8 fields (excluding the BIT-MAP field) in the token; therefore the lowest order 8 bits in the BIT-MAP field are reserved for future use.

| Bit Map Position | Field Name | EMV Tag |
|---|---|---|
| 1 | TERM-SER-NUM | 9F1E |
| 2 | EMV-TERM-CAP | 9F33 |
| 3 | USER-FLD1 | n/a |
| 4 | USER-FLD2 | n/a |
| 5 | EMV-TERM-TYPE | 9F35 |
| 6 | APPL-VER-NUM | 9F09 |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| Bit Map Position | Field Name | EMV Tag |
|---|---|---|
| 7 | CVM-RSLTS | 9F34 |
| 8 | This field will contain one of the following data elements:<br>DF-NAME<br>APPLICATION ID | <br><br><br>84<br>4F |

3–10 02 TERM-SERL-NUM PIC X(8)

The interface device (IFD) number, a unique and permanent serial number assigned to the terminal by the manufacturer.

11–14 02 EMV-TERM-CAP PIC X(4)

The card data input, cardholder verification method (CVM), and security capabilities of the terminal. Valid values are shown in the tables below. The default for all bit settings is a value of 0.

In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.

Bit positions not listed are reserved for future use.

**Caution:** In TAL programming, the highest order bit is the zero bit.

### Byte 1 (Card Data Input Capability)

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Manual key entry capability. Valid values are as follows:<br>0 = The terminal does not support manual key entry to input the card data.<br>1 = The terminal supports manual key entry to input the card data. |

| EMV Defined Bit Position | Description |
|---|---|
| 7 | Magnetic stripe capability. Valid values are as follows:<br>0 = The terminal does not support data capture from the magnetic stripe on the card.<br>1 = The terminal supports data capture from the magnetic stripe on the card. |
| 6 | IC with contacts capability. Valid values are as follows:<br>0 = The terminal does not support data capture from the integrated chip card.<br>1 = The terminal supports data capture from the integrated chip card. |

### Byte 2 (CVM Capability)

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Plaintext PIN for integrated chip card (ICC) verification capability. Valid values are as follows:<br>0 = The terminal does not use plaintext PIN for ICC verification for CVM.<br>1 = The terminal uses plaintext PIN for ICC verification for CVM. |
| 7 | Enciphered PIN for online verification capability. Valid values are as follows:<br>0 = The terminal does not use enciphered PIN for online verification for CVM.<br>1 = The terminal uses enciphered PIN for online verification for CVM. |

| EMV Defined Bit Position | Description |
|---|---|
| 6 | Signature (paper) capability. Valid values are as follows:<br>0 = The terminal does not use signature (paper) verification for CVM.<br>1 = The terminal uses signature (paper) verification for CVM. |
| 5 | Enciphered PIN for offline verification capability. Valid values are as follows:<br>0 = Enciphered PIN for offline verification was not used for CVM by the terminal.<br>1 = Enciphered PIN for offline verification was used for CVM by the terminal. |

### Byte 3 (Security Capability)

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Static data authentication capability. Valid values are as follows:<br>0 = Static data authentication security is not used by this terminal.<br>1 = Static data authentication security is used by this terminal.. |
| 7 | Dynamic data authentication capability. Valid values are as follows:<br>0 = Dynamic data authentication security is not used by this terminal.<br>1 = Dynamic data authentication security is used by this terminal.. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 6 | Card capture capability. Valid values are as follows:<br>0 = The terminal does not have card capture capability.<br>1 = The terminal does have card capture capability.. |

**Byte 4**

USER-FLD1-EMV-TERM-CAP

This field is used to ensure word alignment.

15–16 02 USER-FLD1 PIC X(2)

Must contain binary zeroes.

17–20 02 USER-FLD2 PIC X(4)

Must contain binary zeroes.

21 02 EMV-TERM-TYPE PIC X(1)

The EMV terminal type, indicating the environment of the terminal, its communications capability, and its operational control, as shown in the table below.

| | Operational Control | | |
|---|---|---|---|
| **Environment** | **Financial Institution** | **Merchant** | **Cardholder** |
| Attended Terminal | | | |
| Online only | 11 | 21 | N/A |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| | Operational Control | | |
|---|---|---|---|
| **Environment** | **Financial Institution** | **Merchant** | **Cardholder** |
| Offline with online capability | 12 | 22 | N/A |
| Offline only | 13 | 23 | N/A |
| Unattended Terminal | | | |
| Online only | 14 | 24 | 34 |
| Offline with online capability | 15 | 25 | 35 |
| Offline only | 16 | 26 | 36 |

22–23 02 APPL-VER-NUM PIC X(2)

The version number assigned by the payment system for the terminal application.

24–26 02 CVM-RSLTS PIC X(3)

The results of the last cardholder verification method (CVM) performed. Valid values are shown in the tables below. The default for all bit settings is a value of 0.

In the EMV specifications, definitions that include bit positions indicate that bit position 8 is the leftmost bit.

Bit positions not listed are reserved for future use.

**Caution:** In TAL programming, the highest order bit is the zero bit.

### Byte 1 (CVM Performed)

| EMV Defined Bit Position | Description |
|---|---|
| 7 | 0 = Fail cardholder verification if this cardholder verification method (CVM) is unsuccessful<br>1 = Apply succeeding card verification rule (CVR) if this CVM is unsuccessful |
| 6–1 | 000000= Fail CVM processing<br>000001= Plaintext PIN verification performed by ICC<br>000010= Enciphered PIN verified online<br>000011= Plaintext PIN verification performed by ICC and signature (paper)<br>000100= Enciphered PIN verification performed by ICC<br>000101= Enciphered PIN verification performed by ICC and signature (paper)<br>0*xxxxx* = Values in the range 000110–011101 reserved for future use by the EMV specification<br>011110= Signature (paper)<br>011111= No CVM required<br>10*xxxx* = Values in the range 100000–101111 reserved for use by the individual payment systems<br>11*xxxx* = Values in the range 110000–111110 reserved for future use by the issuer<br>111111= Not available for use |

### Byte 2 (CVM Condition)

| Value | Description |
|---|---|
| 00 | Always |

| Value | Description |
|---|---|
| 01 | If cash or cashback |
| 02 | If not cash or cashback |
| 03 | If terminal supports the CVM |
| 04 | Reserved for future use |
| 05 | Reserved for future use |
| 06 | If transaction is in the application currency and is under *x* value |
| 07 | If transaction is in the application currency and is over *x* value |
| 08 | If transaction is in the application currency and is under *y* value |
| 09 | If transaction is in the application currency and is over *y* value |
| 0A–7F | Reserved for future use |
| 80–FF | Reserved for future use by individual payment systems |

### Byte 3 (CVM Result)

Result of the last CVM performed, as known by the terminal. Valid values are as follows:

| Value | Description |
|---|---|
| 0 | Unknown (for example, for signature) |
| 1 | Failed (for example, for offline PIN) |
| 2 | Successful (for example, for offline PIN) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 27–28 | 02 | DF-NAME-LGTH | TYPE BINARY 16 SIGNED |
| | | The length of the dedicated file name or application identifier in the following field. The ASCII and binary versions of the token must contain the same value in this field. The ASCII version of the token must contain the decimal (not hexadecimal) representation of the length value. | |
| 29–44 | 02 | DF-NAME | PIC X(16) |
| | | The name of the dedicated file (as described in ISO/IEC 7816-4) or application identifier (as described in ISO/IEC 7816-5). The data is left-justified and padded to the right with binary zeroes. | |

# Token B3 EMV Discretionary Data Token—ASCII Format

The fields in the ASCII format EMV Discretionary Data token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–80 | | EMV-DISCR-TKNX | |
| 1–4 | 02 | BIT-MAP | PIC X(4) |
| 5–12 | 02 | TERM-SERL-NUM | PIC X(8) |
| 13–20 | 02 | EMV-TERM-CAP | PIC X(8) |
| 21–24 | 02 | USER-FLD1 | PIC X(4) |
| 25–32 | 02 | USER-FLD2 | PIC X(8) |
| 33–34 | 02 | EMV-TERM-TYPE | PIC X(2) |
| 35–38 | 02 | APPL-VER-NUM | PIC X(4) |
| 39–44 | 02 | CVM-RSLTS | PIC X(6) |
| 45–48 | 02 | DF-NAME-LGTH | PIC X(4) |
| 49–80 | 02 | DF-NAME | PIC X(32) |

# Token B4 EMV Status Token—Binary Format

The EMV Status token holds data identifying the status of a transaction. Device Handler and Interchange Interface processes create this token and add it to the STM before sending it to the Authorization process. The acquiring endpoint adds the token when the transaction originates from an EMV-capable terminal, regardless of whether or not the data relates to an EMV transaction.

For more information about the EMV data elements refer to the MasterCard M/Chip or the Visa Smart Debit Credit (VSDC) documentation sets or the EMVCo specification.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | EMV-STAT-TKN | |
| 1–3 | 02 | PT-SRV-ENTRY-MDE | PIC X(3) |
| | | The point-of-service entry mode. This field indicates the manner in which the card details were entered at the device and the PIN entry capability of the terminal. | |
| 4 | 02 | TERM-ENTRY-CAP | PIC X(1) |
| | | The capability of the terminal. This field is set by the acquiring process. Valid values are as follows:<br>0 = Unknown<br>1 = Reserved for future use<br>2 = Magnetic stripe read capability<br>3-4 Reserved for future use<br>5 = ICC contact read capability<br>8 = Contactless read capability, but no ICC contact read capability | |
| 5 | 02 | LAST-EMV-STAT | PIC X(1) |
| | | Indicates whether the card used to initiate a magnetic stripe transaction is a chip card. Valid values are as follows:<br>0 = Not a chip card<br>1 = A chip card | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 6 | 02 | DATA-SUSPECT | PIC X(1) |
| | | Indicates whether the card authentication method (CAM) data is reliable. This flag is set by the acquiring process. Valid values are as follows:<br>0 = CAM data assumed correct<br>1 = CAM data is unreliable | |
| 7–8 | 02 | APPL-PAN-SEQ-NUM | PIC X(2) |
| | | The application PAN sequence number (EMV Tag 5F34). This field identifies and differentiates cards with the same PAN. This field contains spaces if the card does not include an application PAN sequence number. | |
| 9–14 | 02 | DEV-INFO | PIC X(6) |
| | | The device information field. This field contains device-specific data. | |
| 9–14 | 02 | CAM-FLAGS | REDEFINES DEV-INFO |
| | | Identifies conditions encountered at the terminal. Valid values are shown in the tables below. The default for all bit settings is a value of 0. This field is specific to ATM transactions.<br>This field is specific to an NCR terminal and is defined by NCR in the ***NCR NDC+ CAM 2 Functional Specification.***<br>The two bytes (16 flags) of CAM data defined in the NCR specification are converted to four bytes of ASCII hexadecimal data in the native message for transmission from the ATM. Each of the two bytes is split into four 4-bit units. Each 4-bit unit is represented in the low order four bits of each of the 4 bytes in the native message. The four bytes in the native message are moved directly into the first four bytes of this token field.<br>Bit positions not listed are reserved for future use. | |

### Byte 1

Byte 1 as defined by NCR is moved into bytes 1 and 2 of this token field.

| NCR Defined Bit Position | Description |
|---|---|
| 4 | Application data retrieval indicator. Valid values are as follows:<br>0 = Application data retrieval successful.<br>1 = Application data retrieval failed. |
| 3 | Get processing options indicator. Valid values are as follows:<br>0 = Get processing options successful.<br>1 = Get processing options failed. |
| 2 | Application selection indicator. Valid values are as follows:<br>0 = Application selection successful.<br>1 = Application selection failed. |

### Byte 2

Byte 2 as defined by NCR is moved into bytes 3 and 4 of this token field.

| NCR Defined Bit Position | Description |
|---|---|
| 8 | Processing options data object list (PDOL) data flag: Valid values are as follows:<br>0 = PDOL data valid.<br>1 = PDOL data invalid. |
| 7 | Card risk management data object list (CDOL1) data flag. Valid values are as follows:<br>0 = CDOL1 data valid.<br>1 = CDOL1 data invalid. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| NCR Defined Bit Position | Description |
|---|---|
| 6 | GENERATE AC command flag. Valid values are as follows:<br>0 = GENERATE AC successful.<br>1 = GENERATE AC failed. |
| 4 | Card authentication method (CAM) processing flag. Valid values are as follows:<br>0 = CAM processing not yet successful.<br>1 = CAM processing previously successful. |
| 3 | Easy entry processing flag. Valid values are as follows:<br>0 = Easy entry processing initiated.<br>1 = Easy entry processing not initiated. |
| 2 | CAM processing initiated flag. Valid values are as follows:<br>0 = CAM processing initiated.<br>1 = CAM processing not initiated. |

Bytes 5 and 6 of this token field are reserved for future use.

| | | | |
|---|---|---|---|
| 9–14 | 02 | CVM-RSLTS | REDEFINES DEV-INFO |

The results of the last cardholder verification method (CVM) performed. Valid values are shown in the tables below. The default for all bit settings is a value of 0. This field is specific to POS transactions.

This field is defined as 24 bits (three bytes) by EMV, but is converted to six ASCII bytes, each containing one hexadecimal character representing four bits when included in the EMV Status token.

Bit positions not listed are reserved for future use.

### Byte 1 (CVM Performed)

| Bit Position | Description |
|---|---|
| 7 | 0 = Fail cardholder verification if this cardholder verification method (CVM) is unsuccessful<br>1 = Apply succeeding card verification rule (CVR) if this CVM is unsuccessful |
| 6–1 | 000000 = Fail CVM processing<br>000001 = Plaintext PIN verification performed by ICC<br>000010 = Enciphered PIN verified online<br>000011 = Plaintext PIN verification performed by ICC and signature (paper)<br>000100 = Enciphered PIN verification performed by ICC<br>000101 = Enciphered PIN verification performed by ICC and signature (paper)<br>0xxxxx = Values in the range 000110–011101 reserved for future use by the EMV specification<br>011110 = Signature (paper)<br>011111 = No CVM required<br>10xxxx = Values in the range 100000–101111 reserved for use by the individual payment systems<br>11xxxx = Values in the range 110000–111110 reserved for future use by the issuer<br>111111 = Not available for use |

### Byte 2 (CVM Condition)

| Value | Description |
|---|---|
| 00 | Always |
| 01 | If cash or cashback |
| 02 | If not cash or cashback |
| 03 | If terminal supports the CVM |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| Value | Description |
|---|---|
| 04 | Reserved for future use |
| 05 | Reserved for future use |
| 06 | If transaction is in the application currency and is under X value |
| 07 | If transaction is in the application currency and is over X value |
| 08 | If transaction is in the application currency and is under Y value |
| 09 | If transaction is in the application currency and is over Y value |
| 0A–7F | Reserved for future use |
| 80–FF | Reserved for future use by individual payment systems |

**Byte 3 (CVM Result)**

| Value | Description |
|---|---|
| 0 | Unknown (for example, for signature) |
| 1 | Failed (for example, for offline PIN) |
| 2 | Successful (for example, for offline PIN) |

| | | | |
|---|---|---|---|
| 9–14 | 02 | ICHG-DEF | REDEFINES DEV-INFO |

The interchange definition. This token is used by the VisaNet Interface only.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9–10 | 04 | APPRVD-RC | PIC X(2) |

In some authorization requests received via an interchange interface, this field contains the Authorization Response Code (ARC) required for Authorization Response Cryptogram (ARPC) generation.

| | | | |
|---|---|---|---|
| 11–14 | 04 | UNUSED | PIC X(4) |

This field is reserved for future use.

| | | | |
|---|---|---|---|
| 15–18 | 02 | RSN-ONL-CDE | PIC X(4) |

The message reason code specifies why a transaction is to be authorized online (rather than being completed locally), or why a transaction has been completed locally (rather than being authorized online). Values are defined in the ***ISO 8583 (1993) Standard***. Refer to the ***ACI Standard POS Device Message Specifications Manual*** for additional information on the Standard POS Device Handler.

In a request message, the valid values are as follows:

| Value | Description |
|---|---|
| 1500 | ICC application, common data file unable to process |
| 1501 | ICC application, application data file unable to process |
| 1502 | ICC random selection |
| 1503 | Terminal random selection |
| 1504 | Terminal not able to process ICC |
| 1505 | Online forced by ICC (CDF or ADF) |
| 1506 | Online forced by card acceptor |
| 1507 | Online forced by CAD to be updated |
| 1508 | Online forced by terminal |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| Value | Description |
|---|---|
| 1509 | Online forced by issuer |
| 1510 | Over floor limit |
| 1511 | Merchant suspicious |

In an advice message that the terminal previously has attempted to send to the acquirer as a request message, this field contains the same value as in the original request message.

In an advice message that the terminal previously has not attempted to send to the acquirer as a request message, the valid values are as follows:

| Value | Description |
|---|---|
| 1004 | Terminal processed |
| 1005 | ICC processed |
| 1006 | Under floor limit |
| 1007 | Stand-in processing at the acquirer's option |

19 02 ARQC-VRFY PIC X(1)

The result of the authorization request cryptogram (ARQC) verification. Valid values are as follows:

0 = ARQC was not verified.
1 = ARQC was checked by an acquiring system or switch but failed verification.
2 = ARQC was checked by an acquiring system or switch and passed verification.
3 = ARQC was checked by BASE24 or an issuer system but failed verification.
4 = ARQC was checked by BASE24 or an issuer system and passed verification.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | 5-7 = Reserved for future use.<br>8 = ARQC was not verified, but other EMV processing is required (e.g., ATC check).<br>9 = ARQC was not verified; transaction was downgraded to magnetic stripe (instead of chip). | |
| 20 | 02 | ISO-RC-IND | PIC X(1) |
| | | The ISO 8583 (1987) Response Code Indicator. This field indicates whether the ISO response code sent to the interchange should be used in generating the Authorization Response Cryptogram (ARPC), or if the ISO response code received from the interchange should be returned to the terminal as the Authorization Response Code. Valid values are as follows:<br>b = No information available (where b indicates a blank space)<br>0 = Do not use interchange response code<br>For EMV transactions where BASE24 is the issuer:<br>1 = Use supplied response code in ARPC generation for approved transactions<br>For EMV transactions where BASE24 is the acquirer:<br>9 = Use interchange response code as ARC sent to terminal | |

# Token B4 EMV Status Token—ASCII Format

The fields in the ASCII format EMV Status token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | EMV-STAT-TKNX | |
| 1–3 | 02 | PT-SRV-ENTRY-MDE | PIC X(3) |
| 4 | 02 | TERM-ENTRY-CAP | PIC X(1) |
| 5 | 02 | LAST-EMV-STAT | PIC X(1) |
| 6 | 02 | DATA-SUSPECT | PIC X(1) |
| 7–8 | 02 | APPL-PAN-SEQ-NUM | PIC X(2) |
| 9–14 | 02 | DEV-INFO | PIC X(6) |
| 9–14 | 02 | CAM-FLAGS | REDEFINES DEV-INFO |
| 9–14 | 02 | CVM-RSLTS | REDEFINES DEV-INFO |
| 9–14 | 02 | ICHG-DEF | REDEFINES DEV-INFO |
| 9–10 | 04 | APPRVD-RC | PIC X(2) |
| 11–14 | 04 | UNUSED | PIC X(4) |
| 15–18 | 02 | RSN-ONL-CDE | PIC X(4) |
| 19 | 02 | ARQC-VRFY | PIC X(1) |
| 20 | 02 | ISO-RC-IND | PIC X(1) |

# Token B5 EMV Response Data Token—Binary Format

The EMV Response Data token contains the response cryptogram, data required to generate the response cryptogram, and flags used to identify the scripts to be returned to the acquirer. If authorization is performed on BASE24, the BASE24 Authorization process creates this token. If the transaction is routed to an interchange for authorization, the BASE24 Interchange Interface process creates the token.

All tokens passed in an external message to the BASE24 ISO Host Interface process or the BASE24 BIC Interchange Interface process are sent in ASCII format. When the tokens are processed by the interface, some of the fields may be converted to binary. If this is the case, any unused fields or parts of fields must contain ASCII zeroes rather than spaces.

For more information about the EMV data elements refer to the MasterCard M/Chip or the Visa Smart Debit Credit (VSDC) documentation sets or the EMVCo specification.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | EMV-RESP-TKN | |
| 1–2 | 02 | ISS-AUTH-DATA-LGTH | TYPE BINARY 16 SIGNED |
| | | The length of the binary representation of the data in the following field. The ASCII and binary versions of the token must contain the same value in this field. The ASCII version of the token must contain the decimal (not hexadecimal) representation of the length value. | |
| | 02 | EMV-ISS-AUTH-DATA | PIC X(16) |
| | | The data is left-justified and padded to the right with binary zeroes. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | ISS-AUTH-DATA REDEFINES EMV-ISS-AUTH-DATA<br>Issuer authentication data (EMV Tag 91) sent to the ICC for online issuer authentication. | |
| 3–10 | 04 | ARPC<br>The authorization response cryptogram computed by the card application for online issuer authentication. | PIC X(8) |
| 11–18 | 04 | ADDL-DATA<br>Additional issuer authentication data used in the algorithm to compute the authorization response cryptogram.<br><br>BASE24 currently supports the following definitions for additional issuer data. For more information on these fields, refer to DDL documentation or the individual card scheme documentation. | PIC X(8) |
| 11–18 | 04 | VISA-ADDL-DATA REDEFINES ADDL-DATA<br>The Visa/UKIS definition of the additional issuer authentication data. | |
| 11–12 | 06 | ISS-RESP-CDE<br>The issuer response code used when generating the ARPC. This field is sent to and from Visa in EBCDIC. The Visa Interchange interface converts it to ASCII when placing it in the token.<br><br>The BASE24-atm Authorization process and BASE24-pos Device Handler/Router/Authorization module load this field in the ASCII fomat when creating the token, and log this field to the transaction log files in the binary format without using any conversion utilities. A response code of 00 is logged as 00. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 13–18 | 06 | INFO<br>Issuer discretionary data. | PIC X(6) |
| 11–18 | 04 | MCPA-ADDL-DATA<br>The M/Chip 2.1 definition of the additional issuer authentication data. | REDEFINES ADDL-DATA |
| 11–12 | 06 | ISS-AUTH-RESP-CDE<br>The ISO issuer authorization response code used in generating the authorization response cryptogram (ARPC). | PIC X(2) |
| 13–18 | 06 | INFO<br>Issuer discretionary data. | PIC X(6) |
| 11–18 | 04 | MCHIP4-ADDL-DATA<br>The M/Chip 4 definition of the additional issuer authentication data. | REDEFINES ADDL-DATA |
| 11–12 | 06 | ARPC-RESP-CDE<br>The issuer authorization response code used in generating the authorization response cryptogram. Where the bit value is not specified, a bit setting of 1 indicates the meaning specified, and a bit setting of 0 indicates the opposite.<br>In EMV specifications, definitions which include bit positions indicate that bit position 8 is the leftmost bit. In the tokens, this position is stored in bit position 0 (leftmost bit).<br>This field is binary.<br>Valid values are shown in the tables below.<br>**Caution:** In TAL programming, the highest order bit is the zero bit. | PIC X(2) |

**Byte 1**

| EMV Defined Bit Position | Description |
|---|---|
| 8–5 | Reserved for future use |
| 4–1 | PIN try counters |

**Byte 2**

| EMV Defined Bit Position | Description |
|---|---|
| 8–6 | Reserved for future use |
| 5 | Issuer approves online transaction. Valid values are as follows:<br>0 = Issuer does not approve online transaction.<br>1 = Issuer approves online transaction. |
| 4 | Update PIN try counter. Valid values are as follows:<br>0 = Do not update the PIN try counter.<br>1 = Update the PIN try counter. |
| 3 | Set go online on next transaction. Valid values are as follows:<br>0 = Reset go online on next transaction<br>1 = Set go online on next transaction |
| 2–1 | Update counters. Valid values are as follows:<br>00 = Do not update offline counters<br>01 = Set counters to upper offline limits<br>10 = Reset counters to zero<br>11 = Add transaction to counter |

13–18 06 INFO PIC X(6)

Issuer discretionary data.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–18 | 02 | CCD-A-AUTH-DATA<br>The CCD definition of the issuer authentication data. | REDEFINES EMV-ISS-AUTH-DATA |
| 3–6 | 04 | ARPC<br>The application response cryptogram. The value in this field is computed by the on-line issuer to allow the card to perform issuer authentication. This field is binary. | PIC X(4) |
| 7–10 | 04 | CRD-STAT-UPDT<br>Card status update. This value allows the issuer to provide additional information to the card, and is used in the algorithm to compute the ARPC. Where the bit value is not specified, a bit setting of 1 indicates the meaning specified, and a bit setting of 0 indicates the opposite.<br>In EMV specifications, definitions which include bit positions indicate that bit position 8 is the leftmost bit. In the tokens, this position is stored in bit position 0 (leftmost bit).<br>This field is binary.<br>Valid values are shown in the tables below.<br>**Caution:** In TAL programming, the highest order bit is the zero bit. | PIC X(4) |

**Byte 1**

| EMV Defined Bit Position | Description |
|---|---|
| 8–5 | Reserved for future use |
| 4–1 | PIN try counters |

### Byte 2

| EMV Defined Bit Position | Description |
|---|---|
| 8 | Issuer approves online transaction. Valid values are as follows:<br>0 = Issuer does not approve online transaction.<br>1 = Issuer approves online transaction. |
| 7 | Send card block script.<br>0 = Do not send card block script.<br>1 = Send card block script. |
| 6 | Send application block script.<br>0 = Do not send application block script.<br>1 = Send application block script. |
| 5 | Update PIN try counter. Valid values are as follows:<br>0 = Do not update the PIN try counter.<br>1 = Update the PIN try counter. |
| 4 | Set go online on next transaction. Valid values are as follows:<br>0 = Reset go online on next transaction<br>1 = Set go online on next transaction |
| 3 | Card status update creation flag. Valid values are as follows:<br>0 = The card status update was not created by proxy for the issuer.<br>1 = The card status update was created by proxy for the issuer. |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

| EMV Defined Bit Position | Description |
|---|---|
| 2–1 | Update counters. Valid values are as follows:<br>00 = Do not update online counters<br>01 = Reset online counters to zero<br>10 = Set counters to upper offline limits<br>11 = Add transaction to counter |

### Byte 3

Byte 3 is reserved for future use.

### Byte 4

Byte 4 is issuer discretionary.

11–18 04 ADDL-DATA PIC X(8)

Additional issuer authentication data. This value may be used in the algorithm to compute the ARPC, but it is not used by BASE24. This field is binary.

19 02 SEND-CRD-BLK PIC X(1)

A code indicating whether a card block script is to be generated by the Authorization process and sent to the ICC. Valid values are as follows:

C = Send a PIN change script
N = No, do not send a card block script
U = Send a PIN unblock script
Y = Yes, send a card block script

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 20 | 02 | SEND-PUT-DATA | PIC X(1) |

A code indicating whether a put data script is to be generated by the Authorization process and sent to the ICC. Valid values are as follows:

Y = Yes, send a put data script
N = No, do not send a put data script

# Token B5 EMV Response Data Token—ASCII Format

The fields in the ASCII format EMV Response Data token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–38 | | EMV-RESP-TKNX | |
| 1–4 | 02 | ISS-AUTH-DATA-LGTH | PIC X(4) |
| | 02 | EMV-ISS-AUTH-DATA | PIC X(32) |
| | 02 | ISS-AUTH-DATA REDEFINES EMV-ISS-AUTH-DATA | |
| 5–20 | 04 | ARPC | PIC X(16) |
| 21–36 | 04 | ADDL-DATA | PIC X(16) |
| 21–36 | 04 | VISA-ADDL-DATA | REDEFINES ADDL-DATA |
| 21–24 | 06 | ISS-RESP-CDE | PIC X(4) |
| 25–36 | 06 | INFO | PIC X(12) |
| 21–36 | 04 | MCPA-ADDL-DATA | REDEFINES ADDL-DATA |
| 21–24 | 06 | ISS-AUTH-RESP-CDE | PIC X(4) |
| 25–36 | 06 | INFO | PIC X(12) |
| 21–36 | 04 | MCHIP4-ADDL-DATA | REDEFINES ADDL-DATA |
| 21–24 | 06 | ARPC-RESP-CDE | PIC X(4) |
| 25–36 | 06 | INFO | PIC X(12) |
| 21–36 | 02 | CCD-A-AUTH-DATA | REDEFINES EMV-ISS-AUTH-DATA |
| 5–12 | 04 | ARPC | PIC X(8) |
| 13–20 | 04 | CRD-STAT-UPDT | PIC X(8) |
| 21–36 | 04 | ADDL-DATA | PIC X(16) |
| 37 | 02 | SEND-CRD-BLK | PIC X(1) |
| 38 | 02 | SEND-PUT-DATA | PIC X(1) |

# Token B6 EMV Script Data Token—Binary Format

The EMV Script Data token holds EMV script data. The issuer process creates this token. In the context of EMV transactions, the issuer process can be an Interchange Interface process if the issuer is external to BASE24, or the Authorization process if BASE24 is configured for offline or online/offline authorization. The token is added to the STM before returning the message to the acquiring process. This token is present only if the transaction response contains script data.

For more information about the EMV data elements refer to the MasterCard M/Chip or the Visa Smart Debit Credit (VSDC) documentation sets or the EMVCo specification.

**Note:** The EMV Script Data token is variable length. The values given are the maximum lengths.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–130 | | EMV-SCRIPT-TKN | |
| 1–2 | 02 | ISS-SCRIPT-DATA-LGTH<br>The length of the binary representation of the data in the following field. The ASCII and binary versions of the token must contain the same value in this field. The ASCII version of the token must contain the decimal (not hexadecimal) representation of the length value. | TYPE BINARY 16 SIGNED |
| 3–130 | 02 | ISS-SCRIPT-DATA<br>The Issuer Script Templates (EMV Tag 71 and/or 72) sent to the terminal for processing by the card application. Each template may contain a script ID and one or more script commands. If generated by BASE24, this field includes a single Issuer Script Template, containing only one script command. The data is left-justified and padded to the right with binary zeroes. | PIC X(128) |

# Token B6 EMV Script Data Token—ASCII Format

The fields in the ASCII format EMV Script Data token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–260 | | EMV-SCRIPT-TKNX | |
| 1–4 | 02 | ISS-SCRIPT-DATA-LGTH | PIC X(4) |
| 5–260 | 02 | ISS-SCRIPT-DATA | PIC X(256) |

# Token B7 TLF Token—Binary Format

The TLF token contains the transaction log file name, the relative byte address (RBA), and alternate key data of the record written to the transaction log file. This data is used by the Device Handler and Authorization processes to build a reversal message when the last transaction data is no longer known by the Device Handler process. The fields in the TLF token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–86 | | TLF-BASE-TKN | |
| 1–8 | 02 | RBA-64BIT | TYPE BINARY 64 SIGNED |
| | | The relative byte address (RBA) of the record in a transaction log file where the transaction was logged. | |
| 1–8 | 02 | RBA | TYPE BINARY 32 SIGNED<br>OCCURS 2<br>REDEFINES RBA-64BIT |
| 9–43 | 02 | TLF-NAM | PIC X(35) |
| | | The name of the Transaction Log File where the transaction was logged. | |
| 44 | 02 | TKN-RETRV-OPT | PIC X(1) |
| | | The token data retrieval option. When the Device Handler process is processing a reversal, this field indicates where the token data should be retrieved. Valid values are as follows:<br>0 = Tokens are not included in the reversal message.<br>1 = Token data is retrieved from the BASE24-atm Terminal Data Dynamic File—scratch pad (ATDD2) or BASE24-pos Terminal Data Dynamic File—scratch pad (PTDD2) and appended to the reversal message.<br>2 = Token data is retrieved from the Transaction Log File (TLF) and appended to the reversal message. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 45–74 | 02 | KEY-DATA<br>The alternate key data of the record logged to a transaction log file. | PIC X(30) |
| 45–74 | 02 | ATM-KEY<br>The alternate key data used by the BASE24-atm product. | REDEFINES KEY-DATA |
| 45–74 | 04 | CRD<br>The values in the following fields identify the card issuer and cardholder associated with the transaction. | |
| 45–48 | 06 | LN<br>The logical network associated with the card issuer. | PIC X(4) |
| 49–52 | 06 | FIID<br>The FIID of the card issuer. | PIC X(4) |
| 53–71 | 06 | PAN<br>The cardholder's primary account number (PAN) for card-initiated transactions. | PIC X(19) |
| 72–74 | 06 | MBR-NUM<br>The member number associated with the card used in the transaction. | PIC X(3) |
| 45–74 | 02 | POS-KEY<br>The alternate key data used by the BASE24-pos product. | REDEFINES KEY-DATA |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 45–74 | 04 | CRD<br>The values in the following fields identify the card issuer and cardholder associated with the transaction. | |
| 45–48 | 06 | LN<br>The logical network associated with the card issuer. | PIC X(4) |
| 49–52 | 06 | FIID<br>The FIID of the card issuer. | PIC X(4) |
| 53–74 | 06 | CARD<br>The values in the following fields identify the card used in the transaction. | |
| 53–71 | 08 | CRD-NUM<br>The card number identifying the card used in the transaction. | PIC X(19) |
| 72–74 | 08 | MBR-NUM<br>The member number associated with the card used in the transaction. | PIC X(3) |
| 75–76 | 02 | FILE-FRMT<br>A flag indicating whether the transaction log file is a Format 1 or Format 2 file. Valid values are as follows:<br>1 = Format 1 file<br>2 = Format 2 file | TYPE BINARY 16 SIGNED |
| 77–86 | 02 | USER-FLD1<br>This field is not used. | PIC X(10) |

# Token B7 TLF Token—ASCII Format

The fields in the ASCII format of the TLF token are shown below. Please refer to the binary definition of the TLF token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–100 | | TLF-BASE-TKNX | |
| 1–19 | 02 | RBA | PIC X(19) |
| 20–54 | 02 | TLF-NAM | PIC X(35) |
| 55 | 02 | TKN-RETRV-OPT | PIC X(1) |
| 56–85 | 02 | KEY-DATA | PIC X(30) |
| 56–85 | | ATM-KEY | REDEFINES KEY-DATA |
| 56–85 | 04 | CRD | |
| 56–59 | 06 | LN | PIC X(4) |
| 60–63 | 06 | FIID | PIC X(4) |
| 64–82 | 06 | PAN | PIC X(19) |
| 83–85 | 06 | MBR-NUM | PIC X(3) |
| 56–85 | 02 | POS-KEY | REDEFINES KEY-DATA |
| 56–85 | 04 | CRD | |
| 56–59 | 06 | LlN | PIC X(4) |
| 60–63 | 06 | FIID | PIC X(4) |
| 64–85 | 06 | CARD | |
| 64–82 | 08 | CRD-NUM | PIC X(19) |
| 83–85 | 08 | MBR-NUM | PIC X(3) |
| 86–90 | 02 | FILE-FRMT | PIC X(5) |
| 91–100 | 02 | USER-FLD1 | PIC X(10) |

# Token B8 Transaction Profile Token

The Transaction Profile token contains transaction profile information from the Acquirer Processing Code File (APCF) or from the Issuer Processing Code File (IPCF). This data is used for transactions allowed checking. Transaction acquirers add APCF data to the token while transaction issuers add IPCF data to the token. The fields in the Transaction Profile token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–34 | | TXN-PRFL-TKN | |
| 1–16 | 02 | ACQ-TXN-PRFL<br>The acquirer transaction profile from the APCF. | PIC X(16) |
| 17–32 | 02 | ISS-TXN-PRFL<br>The issuer transaction profile from the IPCF. | PIC X(16) |
| 33 | 02 | DISCRD-RVSL-FLG<br>The discard reversal flag from the IPCF. | PIC X(1) |
| 34 | 02 | APCF-RTE<br>Indicates if the transaction was routed to the process specified in the Acquirer Processing Code File (APCF). Valid values are as follows:<br>Y = Yes, the transaction was routed to the APCF authorization destination.<br>N = No, the transaction was not routed to the APCF authorization destination. | PIC X(1) |

# Token B9 Transaction Description Token

The Transaction Description token contains the processing code description from the Acquirer Processing Code File (APCF) or from the Issuer Processing Code File (IPCF). This data is not used in transaction processing and is informational only. Transaction acquirers add APCF data to the token while transaction issuers add IPCF data to the token. The fields in the Transaction Description token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–60 | | TXN-DESCR-TKN | |
| 1–30 | 02 | ACQ-DESCR-TAG<br>The acquirer processing code description tag from the Acquirer Code Processing File (APCF). | PIC X(30) |
| 31–60 | 02 | ISS-DESCR-TAG<br>The issuer processing code description tag from the Issuer Code Processing File (IPCF). | PIC X(30) |

# Token BA Acquirer Routing Token

The Acquirer Routing token contains the authorization destination and the authorization destination response logging option values from the Acquirer Processing Code File (APCF). This data is added to the token by transaction acquirers. The fields in the Acquirer Routing token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–18 | | ACQ-RTE-TKN | |
| 1–16 | 02 | AUTH-DEST<br>The name of the authorization destination to which transactions from the acquiring endpoint are to be sent. | PIC X(16) |
| 17 | 02 | LOG-AUTH-DEST-RESP<br>A field indicating whether responses for transactions sent to the process specified in the AUTH-DEST field are to be logged by the BASE24 system. Valid values are as follows:<br>Y = Yes, log the response.<br>N = No, do not log the response (default). | PIC X(1) |
| 18 | 02 | USER-FLD1<br>This field is not used. | PIC X(1) |

# Token BB Pre-Pay Generic Receipt Token—Binary Format

The Pre-Pay Generic Receipt token contains the generic receipt message. It is populated by the Transaction Context Manager process using the telco's Mobile Operator File (MOF) record. The fields in the Pre-Pay Generic Receipt token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–210 | | PRE-PAY-GENRC-RCPT-TKN | |
| 1–6 | 02 | MSG-TS | TYPE BINARY 16 OCCURS 3 TIMES |
| | | Contains the timestamp when the generic receipt message was last changed. | |
| 7 | 02 | DFLT-RCPT-MSG-FLG | PIC X |
| | | A flag indicating whether the default receipt message is used in conjunction with the telco operator message contained in the PRE-PAY-RCPT-TKN. | |
| 8 | 02 | USER-FLD-ACI | PIC X |
| | | Reserved for future use. | |
| 9–10 | 02 | LGTH | TYPE BINARY 16 |
| | | Length of the data in the MSG field. | |
| 11–210 | 02 | MSG | PIC X(200) |
| | | The generic data containing the telco specific message. This field can contain up to 200 characters. | |

# Token BB Pre-Pay Generic Receipt Token—ASCII Format

The fields in the ASCII format Pre-Pay Generic Receipt token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–220 | | PRE-PAY-GENRC-RCPT-TKNX | |
| 1–15 | 02 | MSG-TS | PIC X(5)<br>OCCURS 3 TIMES |
| 16 | 02 | DFLT-RCPT-MSG-FLG | PIC X |
| 17 | 02 | USER-FLD-ACI | PIC X |
| 18–20 | 02 | LGTH | PIC X(3) |
| 21–220 | 02 | MSG | PIC X(200) |

# Token BC TSS Index Token

The TSS Index token provides indexes of keys used by the Transaction Security Services (TSS) process. Transaction acquirers use it to indicate that only the specified keys are used to attempt PIN, MAC, or data verification instead of repeating an attempt with more than one key. The fields in the TSS Index token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–6 | | TSS-IDX-TKN | |
| 1 | 02 | PIN-KEY-IDX<br>The index of the PIN encryption key. | PIC X |
| 2 | 02 | MAC-KEY-IDX<br>The index of the MAC encryption key. | PIC X |
| 3 | 02 | DATA-KEY-IDX<br>The index of the data encryption key. | PIC X |
| 4–6 | 02 | USER-FLD1<br>This field is not used. | PIC X(3) |

# Token BD Multiple Currency Token—Binary Format

The Multiple Currency token contains information about the different currencies used during a transaction. The token ID is BD. The default token is created with five currency occurrences. You can change this value by modifying the CUSTCNST file. This token is added by BASE24 currency conversion utilities when the first currency conversion is performed for a transaction. It is updated whenever subsequent currency conversions are performed.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–192 | | MULT-CRNCY-TKN | |
| 1–8 | 02 | TXN-AMT-1 | TYPE BINARY 64 |
| | | The transaction amount in the transaction currency. | |
| 9–16 | 02 | TXN-AMT-2 | TYPE BINARY 64 |
| | | The additional amount in the transaction currency. | |
| 17–19 | 02 | TXN-CRNCY-CDE | [CRNCY-CDE] PIC 9(3) |
| | | The ISO numeric currency code of the transaction currency. | |
| 20 | 02 | USER-FLD1 | PIC 9 |
| 21–30 | 02 | CRNCY-FLG | PIC X(10) |
| | | Indicators specifying the position in the CRNCY array of the data related to the various currencies involved in the transaction. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

Each character position (specified from left to right) is used as an indicator for a different currency type, and is referenced by a literal indicating the offset into the field. Valid position values are as follows:

| Position | Currency |
|---|---|
| 1 | Terminal currency |
| 2 | Reserved for future use |
| 3 | Acquirer Institution currency |
| 4 | Logical Network currency |
| 5 | Reserved for future use |
| 6 | Issuer Institution currency |
| 7 | *From* Account currency |
| 8 | *To* Account currency |
| 9 | Reserved for future use |
| 10 | Base currency |

Valid values for each byte are as follows:

0–9 = The offset into the array of the appropriate data
S = The currency is the same as the Saved Currency in the DCC Processing token.
T = The currency is the same as the transaction currency
b = The currency is not required for the transaction (where b indicates a blank space)

31–32 02 NUM-CRNCY TYPE BINARY 16

The number of entries in the following currency code array.

33–192 02 CRNCY OCCURS CUST-MAX-CRNCY-ENTRIES-L TIMES

The array containing the data related to the various currencies involved in the transaction. Each occurrence is 32 positions in length.

04 AMT-1 TYPE BINARY 64

The transaction amount in the currency specified by the CRNCY-CDE field.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | AMT-2 | TYPE BINARY 64 |
| | | The additional amount in the currency specified by the CRNCY-CDE field. | |
| | 04 | CRNCY-CDE | PIC 9(3) |
| | | The ISO numeric currency code of the amounts in this array entry. | |
| | 04 | CONV-RATE | PIC 9(8) |
| | | The derived exchange rate used to convert the amounts from the transaction currency to the currency specified by the CRNCY-CDE field (for information only). | |
| | | This field is in ISO format (i.e., the leftmost digit specifies the number of decimal places, and positions 2–8 provide the rate.) For example, 69972552 = 9.972522. | |
| | 04 | CONV-DAT | PIC 9(4) |
| | | The date on which the currency conversion was performed, in MMDD format. | |
| | 04 | USER-FLD2 | PIC X(1) |

# Token BD Multiple Currency Token—ASCII Format

The fields in the ASCII format Multiple Currency token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–328 | | MULT-CRNCY-TKNX | |
| 1–19 | 02 | TXN-AMT-1 | PIC X(19) |
| 20–38 | 02 | TXN-AMT-2 | PIC X(19) |
| 39–41 | 02 | TXN-CRNCY-CDE | PIC 9(3) |
| 42 | 02 | USER-FLD1 | PIC X(1) |
| 43–52 | 02 | CRNCY-FLG | PIC X(10) |
| 53–57 | 02 | NUM-CRNCY | PIC 9(5) |
| 58 | 02 | USER-FLD3 | PIC X(1) |
| 59–328 | 02 | CRNCY | OCCURS CUST-MAX-CRNCY-ENTRIES-L TIMES |
| | 04 | AMT-1 | PIC X(19) |
| | 04 | AMT-2 | PIC X(19) |
| | 04 | CRNCY-CDE | PIC 9(3) |
| | 04 | CONV-RATE | PIC 9(8) |
| | 04 | CONV-DAT | PIC 9(4) |
| | 04 | USER-FLD2 | PIC X(1) |

# Token BE Original Currency Release 6.0 Token—Binary Format

The Original Currency Release 6.0 token contains information about the original transaction currency. This token is added when a transaction request or transaction advice is received that contains amount data in two different currencies. Typically, this occurs when an international interchange, such as VisaNet or Banknet, converts amounts from the transaction currency (used by the acquirer) to the cardholder billing currency (used by the issuer). BASE24 uses the amounts in the cardholder billing currency for authorization processing, and the interchange interface adds the information in the transaction currency to the Original Currency Release 6.0 token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–40 | | ORIG-CRNCY-60-TKN | |
| 1–8 | 02 | AMT-1<br>The transaction amount in the Original currency. | TYPE BINARY 64 SIGNED |
| 9–16 | 02 | AMT-2<br>The additional amount in the Original currency. | TYPE BINARY 64 SIGNED |
| 17–19 | 02 | CRNCY-CDE<br>The currency code of the Original currency. | PIC 9(3) |
| 20–27 | 02 | CONV-RATE<br>The rate used to convert the original amount to the currency received by BASE24.<br>This field is in ISO format (i.e., the leftmost digit specifies the number of decimal places, and positions 2–8 specify the rate). For example, 69972522 equals 9.972522. | PIC 9(8) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 28–31 | 02 | CONV-DAT<br>The date on which the currency conversion was performed. | PIC 9(4) |
| 32 | 02 | CONV-IND<br>An indicator specifying the entity that performed the currency conversion. Values:<br>b = No information available (where b indicates a blank space).<br>0 = Interchange<br>1 = Terminal (dynamic currency conversion) | PIC X |
| 33–40 | 02 | USER-FLD1 | PIC X(8) |

# Token BE Original Currency Release 6.0 Token—ASCII Format

The fields in the ASCII format Original Currency Release 6.0 token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–62 | | ORIG-CRNCY-60-TKNX | |
| 1–19 | 02 | AMT-1 | PIC X(19) |
| 20–38 | 02 | AMT-2 | PIC X(19) |
| 39–41 | 02 | CRNCY-CDE | PIC 9(3) |
| 42–49 | 02 | CONV-RATE | PIC 9(8) |
| 50–53 | 02 | CONV-DAT | PIC 9(4) |
| 54 | 02 | CONV-IND | PIC X |
| 55–62 | 02 | USER-FLD1 | PIC X(8) |

# Token BF Pre-Pay Receipt Token—Binary Format

The Pre-Pay Receipt token contains the information used on the receipt for a Pre-Pay transaction. The fields in the Pre-Pay Receipt token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–262 | | PRE-PAY-RCPT-TKN | |
| 1–2 | 02 | LGTH<br>Length of the token buffer. | TYPE BINARY 16 |
| 3–6 | 02 | FIID<br>The FIID assigned to the Mobile Operator Interface. The value in this field matches the value in the ICF. | PIC 9( 4) |
| 7–262 | 02 | BUF<br>Field used for any marketing message. | PIC X(256) |
| 7–262 | 02 | PRE-PAY-RCPT | REDEFINES BUF |
| 7–206 | 04 | OPER-MSG<br>Marketing message as supplied by the Mobile Operator. | PIC X(200) |
| 207–262 | 04 | USER-FLD<br>Reserved for future use. | PIC X(56) |

# Token BF Pre-Pay Receipt Token—ASCII Format

The fields in the ASCII format of the Pre-Pay Receipt token are shown below. Refer to the binary definition of the Pre-Pay Receipt token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–264 | | PRE-PAY-RCPT-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD | PIC X(1) |
| 5–8 | 02 | FIID | TYPE * |
| 9–264 | 02 | BUF | PIC X(256) |
| 9–264 | 02 | PRE-PAY-RCPT | REDEFINES BUF |
| 9–208 | 04 | OPER-MSG | PIC X(200) |
| 209–264 | 04 | USER-FLD | PIC X(56) |

# Token BG Track 3 Token

The Track 3 token contains Track 3 data. The fields in the Track 3 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–108 | | TRK3-TKN | |
| 1–107 | 02 | TRK3<br>This field contains the Track 3 data. | PIC X(107) |
| 108 | 02 | USER-FLD1<br>This field ensures word alignment. | PIC X |

# Token BH Reversal Date and Time Token

The Reversal Date and Time token contains the actual date and time that a reversal was initiated at a POS terminal. This data is required by interchanges in some countries (e.g., Italy and Germany) for all reversals. The token also contains a reversal code field, which can be used when additional reversal code values are required but not permitted in the standard internal message (e.g., in the Netherlands). The fields in the Reversal Date and Time token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–28 | | RVSL-DAT-TIM-TKN | |
| 1–8 | 02 | TXN-DAT<br>The transaction date in YYYYMMDD format. | PIC X(8) |
| 9–16 | 02 | TXN-TIM<br>The transaction time in hhmmsstt format. | [TIM] |
| 9–10 | 04 | HH | PIC X(2) |
| 11–12 | 04 | MM | PIC X(2) |
| 13–14 | 04 | SS | PIC X(2) |
| 15–16 | 04 | TT | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 17-18 | 02 | RVSL-CDE | PIC 9(2) |

A code that indicates the reason for the reversal. This field is used only for a reversal (0420). Valid values are as follows:

- 06 = Transaction was declined for an unspecified reason.
- 09 = Transaction was reversed due to an unknown transaction status.
- 17 = Transaction was cancelled by the customer.
- 32 = Transaction was partially completed.
- 68 = Time out. Transaction was reversed by the acquirer because a response was not received or was received too late.
- 81 = Money not taken. Transaction was reversed because the customer took their card, but did not take their money.
- 82 = Card not taken, card retained. Transaction was reversed because the customer did not take their card.
- 85 = Card not returned. Transaction was reversed because the card could not be returned due to equipment failure.
- 86 = Transaction failed, card returned.

| | | | |
|---|---|---|---|
| 19–28 | 02 | USER-FLD1 | PIC X(10) |

This field ensures word alignment.

# Token BI Pre-Pay Top-Up Token

The fields in the Pre-Pay Top-Up token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–202 | | PRE-PAY-TOP-UP-TKN | |
| 1–40 | 02 | TRK2<br>This field carries Track 2 data captured by the acquiring interface (e.g., SPDH). The data in this field is formatted into a Track 2 buffer. | PIC X(40) |
| 1 | 04 | RTE-METHOD<br>Code indicating the method used by the acquiring interface to identify the Mobile Operator (Service Provider). Also known as the entry ID as listed in the tables above. Valid values are as follows:<br>M = PAN for phone top-up card issued by mobile operator (key entered)<br>; = PAN for phone top-up card issued by mobile operator (swiped)<br>I = Mobile Operator ID (IIN)<br>P = Registered customer phone number<br>L = IIN and registered customer phone number | PIC (X) |
| 2–40 | 04 | TRK2-DATA<br>Buffer containing the rest of the track 2 data. The buffer is formatted based on one of the options listed above. Utilities or services are available to interrogate this data. | PIC X(39) |
| 41–42 | 02 | TXN-CDE<br>The internal BASE24 transaction code for the pre-pay top-up transaction. This is established by the acquiring interface (e.g., SPDH). | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 43–57 | 02 | PHN-NUM | PIC X(15) |
| | | The phone number to receive the top-up. This is established by the Transaction Content Manager (TCM) process or entered by the cardholder. It may be extracted from the TUP-TRACK2 field or retrieved from the Mobile Customer File (MCF) by TCM. | |
| 58 | 02 | PMNT-METHOD | PIC X |
| | | Code that indicates the method of payment for the top-up by the customer. This field is established by the acquiring interface. Valid values are as follows:<br>N = No card type (cash)<br>D = Debit card<br>C = Credit card<br>U = Unknown | |
| 59–73 | 02 | REF | PIC X(15) |
| | | Reference data (e.g., Auth Code or Tran ID) issued by Mobile Operator in top-up and refund top-up responses, which may be required for refunds, confirmations, or reversals (ORIG-REF). The Mobile Phone Operator Interface process establishes this field. The operator should return it in an approved top-up response. | |
| 74–88 | 02 | ORIG-REF | PIC X(15) |
| | | Reference data from the REF field in the original message for a refund or reversal transaction. This field is established by the acquiring interface. | |
| 89–90 | 02 | CRD-TYP | PIC X(2) |
| | | The card type that the Device Handler process uses to update the BASE24-pos Terminal Data File and BASE24-atm Terminal Data File totals. This field is updated by the Transaction Context Manager process using information from the Mobile Operator File (MOF). | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 91–106 | 02 | OPER-NAM-RCPT | PIC X(16) |
| | | The mobile operator name retrieved from the Mobile Operator File (MOF). This field will appear on receipts. This field is updated by the Transaction Context Manager process. | |
| 107–110 | 02 | OPER-ID | PIC X(4) |
| | | The mobile operator ID used in retrieving a record in the Mobile Operator File (MOF). This field is updated by the Transaction Context Manager process. | |
| 111–128 | 02 | SVC-BAL | PIC X(18) |
| | | The amount on the phone after top-up. The Mobile Phone Operator Interface process establishes this field. The operator should return it in an approved top-up response. | |
| 129–146 | 02 | OTHER-BAL | PIC X(18) |
| | | Value indicating a balance. For example, the value in this field could be the amount of time on the phone after a top-up transaction (e.g., hhhhmmss). The Mobile Phone Operator Interface process establishes this field. The operator should return it in an approved top-up response. | |
| 147–162 | 02 | ACTVN-CDE | PIC X(16) |
| | | A code supplied to the customer to activate a top-up. This field is supplied to the customer in the clear and is logged to the BASE24-pos Transaction Log File as spaces. The Mobile Phone Operator Interface process establishes this field. The operator should return it in an approved top-up response. | |
| 163–177 | 02 | APPRV-CDE | PIC X(15) |
| | | Approval code as supplied in responses from the mobile operator. The Mobile Phone Operator Interface process establishes this field. The operator should return it in an approved top-up response. For declined transactions, this field may contain the response code returned from the operator. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 178–182 | 02 | RESP-CDE | PIC X(5) |
| | | Response code returned from mobile operator. This code is the operator's actual response code. Alternatively, this field may contain a generic response code agreed upon between the mobile operator and the service provider. This code is converted by the Mobile Phone Operator Interface process into an internal BASE24 response code that is carried in the (P)STM response code field. This field is also used when the Transaction Context Manager process has authorized a transaction. | |
| 183–201 | 02 | OPER-RTLR-ID | PIC X(19) |
| | | The mobile operator retailer ID used for clearing against the mobile operator within the Card Management System. | |
| 202 | 02 | RVSL-CDE | PIC (X) |
| | | A code indicating the reason for a reversal. Valid values are as follows:<br>␣ = Not a reversal (where ␣ indicates a blank space)<br>S = Reason code specified in (P)STM<br>A = Mobile Operator denial | |

# Token BJ EMV Issuer Script Results Token

The EMV Issuer Script Results token holds information about the processing of EMV Script data. This token is created by the acquirer interface process (e.g., Device Handler or Interchange Interface) or sent by the acquirer. It contains information about the results of EMV Script processing.

For more information about the EMV data elements refer to the MasterCard M/Chip or the Visa Smart Debit Credit (VSDC) documentation sets or the EMVCo specification.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–82 | | EMV-ISS-SCRIPT-RSLTS-TKN | |
| 1 | 02 | NUM-ISS-SCRIPT-RSLTS<br>The number of completed issuer script results contained within the token. | PIC X(1) |
| 2 | 02 | USER-FLD1<br>For future use within the token. | PIC X(1) |
| 3–82 | 02 | ISS-SCRIPT-RSLTS-DATA | OCCURS 0 TO 8 TIMES DEPENDING ON NUM-ISS-SCRIPT-RSLTS |
| | 04 | ISS-SCRIPT-PROC-RSLT<br>A code indicating the result of the script processing. Valid values are as follows:<br>0 = Script not performed<br>1 = Script processing failed<br>2 = Script processing successful<br>9 = Script processing unknown | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | ISS-SCRIPT-SEQ | PIC X(1) |

The details of the Script Sequence in the processing. Valid values are as follows:

| | | |
|---|---|---|
| 0 | = | Script sequence not specified, script not performed, all commands successful. |
| 1–9, A–E | = | Sequence number from 1–14 for failed command. |
| F | = | Sequence number if 15 or over for failed command. |

| | | |
|---|---|---|
| 04 | ISS-SCRIPT-ID | PIC X(8) |

The issuer script identifier.

# Token BK Multiple Logical Network Token—Binary Format

The Multiple Logical Network token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–8 | | MULT-LN-TKN | |
| 1–4 | 02 | ACQ-LN<br>Contains the acquirer's logical network ID when the transaction is routed to another logical network after locating a CPF record in the Sproute File. This field is updated by the Authorization process or the Router/Authorization module. | PIC X(4) |
| 5 | 02 | SITE-IND<br>Used for transactions processed in a dual site system. The value p indicates a transaction that originated on a remote BASE24 system. | PIC X |
| 6–8 | 02 | USER-FLD1<br>Reserved for future use. | PIC X(3) |

# Token BK Multiple Logical Network Token—ASCII Format

The fields in the ASCII format of the Multiple Logical Network token are shown below. Refer to the binary definition of the Multiple Logical Network token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–8 | | MULT-LN-TKNX | |
| 1–4 | 02 | ACQ-LN | PIC X(4) |
| 5 | 02 | SITE-IND | PIC X(1) |
| 6–8 | 02 | USER-FLD1 | PIC X(3) |

# Token BL Virtual Primary Account Number Token

The Virtual Primary Account Number token is used to carry the virtual primary account number and its associated expiration date. The fields in the Virtual Primary Account Number token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–34 | | PSEUDO-CRD-NUM-TKN | |
| 1–19 | 02 | PSEUDO-CRD-NUM<br>The virtual primary account number. | PIC X(19) |
| 20–23 | 02 | EXP-DAT<br>The expiration date of the authentication transaction associated with the virtual primary account number. | PIC X(4) |
| 24–34 | 02 | USER-FLD<br>This field is not used. | PIC X(11) |

# Token BM Transaction Subtype Token

Transaction Subtypes are four-character values used to indicate that the processing associated with a particular transaction code should be altered. The Transaction Subtype token identifies the type of transaction being processed. The fields in the Transaction Subtype token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–36 | | TXN-SUBTYP-TKN | |
| 1–4 | 02 | TXN-SUBTYP | PIC X(4) |

A subtype identifier to further describe this transaction. All alphanumeric characters are valid in this field. Values P000 through RZZZ are provided for user-defined transaction subtypes. Subtypes in the ranges 0000 through OZZZ and S000 through WZZZ are reserved for use by BASE24 products. Subtypes in the ranges X000 through ZZZZ are reserved for use by distributors. For the various BASE24 products, Transaction Subtypes, their codes and descriptions, follow below:

**BASE24 Base**

| Subtype | Transaction Subtype Description |
|---|---|
| B000 | Payment from Third Party |
| B001 | Payment to Third Party |
| BBT0 | BCGI Top-Up |

**BASE24-atm**

| Subtype | Transaction Subtype Description |
|---|---|
| ABC0 | Bulk Check transaction. |
| ABL0 | Electronic Bill Payment Payee List |
| ABP0 | Electronic Bill Payment |
| ACR0 | Enhanced Card Review |

| | |
|---|---|
| AER0 | Exchange Rate Notification |
| AIS0 | IFX Interim Statement |
| AMA0 | Multiple Account w/Balances Inquiry |
| API0 | Preferred Transaction Inquiry |
| APS0 | Preferred Transaction Set-up |
| APT0 | Preferred Transaction |
| APU0 | Passbook Update |

**BASE24-pos**

| Subtype | Transaction Subtype Description |
|---|---|
| C000 | Account Funding Transaction |
| C001 | Healthcare/Transit Auto-Substantiation |
| C002 | Healthcare Eligibility Inquiry |
| C003 | Dormancy Transaction |
| C004 | Escheatment Transaction |
| C005 | Payment Transaction |
| C006 | Original Credit Transaction |
| C007 | Loyalty/Sweepstakes/Extras Transactions |
| C008 | Quasi-cash Transaction |
| C009 | Account Status Inquiry |
| CI00 | Canadian Idebit |

5–10 02 ACQ-PROC-CDE PIC X(6)

The acquirer's external processing code. The first two characters of the processing code indicate the type of transaction, the next two characters specify the *from* account for the transaction, and the last two characters specify the *to* account for the transaction.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 11–16 | 02 | ISS-PROC-CDE | PIC X(6) |
| | | The issuer's external processing code. The first two characters of the processing code indicate the type of transaction, the next two characters specify the *from* account for the transaction, and the last two characters specify the *to* account for the transaction. | |
| 17–36 | 02 | USER-FLD1 | PIC X(20) |
| | | This field is not used. | |

# Token BN Data Encryption Key Token

The fields in the Data Encryption Key token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | DATA-ENCRYPTION-KEY-TKN | |
| 1 | 02 | FLD-TYP<br>The type of data being encrypted. Currently, the only valid value is 1 (balance). | PIC X |
| 2 | 02 | USER-FLD | PIC X |
| 3–4 | 02 | ENCRYPT-TYP<br>The type of data encryption to be performed. Valid values are as follows:<br>03 = Electronic Code Block (ECB)<br>04 = Cipher Block Chaining method 16 byte binary (CBC)<br>14 = Cipher Block Chaining method 16 byte unpacked hex (CBC)<br>24 = Cipher Block Chaining method 32 byte character hex (CBC) | PIC X(2) |
| 5–20 | 02 | ENCRYPT-KEY<br>The security module key locator value used by TSS to extract the encryption key from the TSS database for use by the Interac switch. | PIC X(16) |

# Token BO Encrypted Balance Token—Binary Format

The Encrypted Balance token can be either binary or ASCII format, depending on the value of the DATA-TYP field included in the message. The fields in the Encrypted Balance token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–34 | | ENCRYPTED-BAL-TKN | |
| 1 | 02 | DATA-TYP<br>The type of encrypted data contained in the following fields. Valid values are as follows:<br>B = Binary<br>U = Unpacked ASCII hexadecimal | PIC X |
| 2 | 02 | USER-FLD | PIC X |
| 3–18 | 02 | BAL-1<br>The first encrypted balance. This value can contain the available balance. If only one balance is sent, this field will be populated. | PIC X(16) |
| 19–34 | 02 | BAL-2<br>The second encrypted balance. This value can contain the ledger balance. If only one balance is sent, this field is not populated. | PIC X(16) |

# Token BO Encrypted Balance Token—ASCII Format

The fields in the ASCII format of the Encrypted Balance token are shown below. Refer to the binary definition of the Encrypted Balance token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–66 | | ENCRYPTED-BAL-TKN | |
| 1 | 02 | DATA-TYP | PIC X |
| 2 | 02 | USER-FLD1 | PIC X |
| 3–34 | 02 | BAL-1 | PIC X(32) |
| 35–66 | 02 | BAL-2 | PIC X(32) |

# Token BP Person-to-Person Transaction Token

The fields in the Person-to-Person Transaction token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–382 | | P2P-TXN-TKN | |
| 1 | 02 | INTENDED-USE-FLAG | PIC X |

The intended use flag identifies the intended use of the Person-to-Person transaction. Valid values are as follows:

0 = Movement of funds between two accounts owned by a single consumer but housed at different institutions (Interbank Transfer)
1 = Movement of funds between accounts owned by two separate consumers (Interbank Transfer)
2 = Payer-initiated payment
3 = Payee-initiated payment
4 = Account Verification Inquiry
5 = Generic Account Balance Inquiry

| | | | |
|---|---|---|---|
| 2 | 02 | BUS-USE-FLAG | PIC X |

The business use flag identifies the business use of the Person-to-Person transaction. Valid values are as follows:

0 = Auction (account to account auction sale)
1 = Business (business to business)
2 = Person (account to account, non-auction sale)
3 = Utility (e.g., cable, electricity, natural gas, telephone)
4 = Brokerage
5 = Government
6 = Credit card issuer (e.g., Amex, Visa MC issuer)
7 = Various consumer loans (e.g., mortgages, installment loans)
8 = Insurance premiums
9 = Private label cards (e.g., JC Penney, Sears)
A = Rent
B = Collections
C = Education

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | D = Account authentication<br>E = Funding<br>F = Charity giving<br>b = Other | |
| 3–17 | 02 | TRC-NUM<br>The unique trace number assigned by the acquirer. This number is left-justified and space-filled. | PIC X(15) |
| 18–77 | 02 | THIRD-PARTY-ID<br>The identity of the third party involved in the Person-to-Person transaction. The acquirer of the transaction populates the identification of the receiver/sender/payer/payee. This field is left-justified and space-filled. | PIC X(60) |
| 78–102 | 02 | THIRD-PARTY-ACCT-NUM<br>The billing account number for the third party involved in the Person-to-Person transaction. | PIC X(25) |
| 103–127 | 02 | THIRD-PARTY-PHN-NUM<br>A code identifying the third party's phone or pager number. This field is left-justified and space-filled. | PIC X(25) |
| 128–152 | 02 | THIRD-PARTY-INVC-NUM<br>The invoice number associated with the third party. This field is left-justified and space-filled. | PIC X(25) |
| 153–212 | 02 | THIRD-PARTY-EMAIL-ADDR<br>The third party's e-mail address. This field is left-justified and space-filled. | PIC X(60) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 213–262 | 02 | THIRD-PARTY-DESCR | PIC X(50) |
| | | A description of the third party involved in the Person-to-Person transaction. This field is left-justified and space-filled. | |
| 263–322 | 02 | RECIPIENT-EMAIL-ADDR | PIC X(60) |
| | | The recipient's e-mail address. This field is left-justified and space-filled. | |
| 323–362 | 02 | RECIPIENT-NUM | PIC X(40) |
| | | The recipient's phone/pager/fax number. This field is left-justified and space-filled. | |
| 363–382 | 02 | USER-FLD1 | PIC X(20) |
| | | This is a short, free-form message from the sender to the receiver used in interbank transfer transactions. This field is left-justified and space-filled. | |

# Token BQ Completion Required Token

The Completion Required token is used to pass the COMPL-REQ field in the STM/PSTM to and from the host. The token is initially added by HISO (ATM or POS).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–2 | | COMPL-REQ-TKN | |
| 1 | 02 | COMPL-REQ<br>The token is only added on 0210 responses where the compl-req field is set to 2. Valid values are as follows:<br>2 = Host not available. The Authorization process generated a completion regardless of the completion required in the IPCF. | PIC 9 |
| 2 | 02 | USER-FLD1<br>This field is reserved for future use. | PIC X |

# Token BR Split Transaction Routing Token

The Split Transaction Routing token carries data that allows BASE24 to route multiple transaction requests related to a single cardholder request. This token also allows BASE24 to identify and merge the multiple responses received into a single response destined for the cardholder. The fields in the Split Transaction Routing token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–158 | | SPLIT-TXN-RTE-TKN | |
| 1–2 | 02 | TXN-RESP-IND | PIC X(2) |

The transaction response indicator identifies the type of request to which a response received is related. Valid values are as follows:

PF = Primary; Funds Authorization
SF = Secondary; Funds Authorization
TF = Tertiary; Funds Authorization (future use)
PS = Primary; Subtype Authorization
SS = Secondary; Subtype Authorization
TS = Tertiary; Subtype Authorization (future use)

| | | | |
|---|---|---|---|
| 3–4 | 02 | TXN-STAT | PIC X(2) |

Transaction status indicates to BASE24 the status of a secondary authorization. The status can indicate that secondary authorization is not required, secondary authorization is complete, or it can indicate that a response was received from the secondary authorizer but further information must be obtained before authorization can proceed. Valid values are as follows:

C1 = Cardholder cancelled the transaction, amount unavailable
C2 = Cardholder cancelled the transaction, telecommunication provider unrecognized
C3 = Cardholder cancelled the transaction, tax notification

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

PC = Primary authorizer requires cardholder confirmation. The BASE24-atm Transaction Content Manager (TCM) sets and uses this value to recognize a request was generated by the Device Handler (DH) after an account selection response message (type 9906) or a surcharge notify response message (type 9901) was sent to the DH. TCM then forwards any subsequent request back to funds authorizer.

SC = Secondary authorizer requires cardholder confirmation (TCM formats and sends a 9909 message to the DH. TCM subsequently forwards a new DH-generated 0200 request to the funds authorizer).

SD = Secondary authorizer denied transaction (TCM sends the 0210 response to the DH).

SF = Secondary authorization complete (final response from secondary authorizer received; TCM solicits funds authorization).

SI = Secondary authorization incomplete (TCM formats and sends 9909 to the DH; TCM subsequently forwards a new 0200 request generated by the DH to the secondary authorizer).

**Note**: Values beginning with T are reserved for future use and are to be associated with Tertiary authorization.

5–20 02 ORIG-PRO-NAME [SYM-NAME] PIC X(16)

The symbolic name of the BASE24-atm process that received the transaction. For transactions that involve split transaction routing, this field is set by the acquiring endpoint (e.g., the Device Handler process). This field is required when the BASE24-atm Transaction Context Manager (TCM) process resides between the Device Handler process and the Authorization process.

21–36 02 FUNDS-AUTH-DEST [SYM-NAME] PIC X(16)

The name of the destination associated with the funds authorization request. The data for this field is obtained from the Split Transaction Routing File (STRF). This field is

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | required when the BASE24-atm Transaction Context Manager process resides between the Device Handler process and the Authorization process. | |
| 37 | 02 | FUNDS-AUTH-STAT | PIC X |
| | | The value in this field, in combination with the values in the RTE-HRCHY field, the TXN-RESP-IND field, and the TXN-STAT field, enables the BASE24-atm Transaction Context Manager process to determine the next routing decision. Valid values are as follows:<br>0 = Information request complete<br>1 = Transaction complete<br>2 = Transaction declined<br>3 = Transaction reversed<br>4 = Transaction reversed, need acknowledgement | |
| 38–39 | 02 | RTE-HRCHY | PIC X(2) |
| | | The routing hierarchy obtained from the Split Transaction Routing File (STRF). The routing hierarchy supplies the BASE24-atm Transaction Context Manager with information necessary for routing decisions. | |
| 40–55 | 02 | SCND-SVC-DEST [SYM-NAME] | PIC X(16) |
| | | The name of the destination associated with the secondary service request. The data for this field is obtained from the Split Transaction Routing File (STRF). | |
| 56 | 02 | SCND-SVC-STAT | PIC X |
| | | The value in this field, in combination with the values in the RTE-HRCHY field, the TXN-RESP-IND field, and the TXN-STAT field, enables the BASE24-atm Transaction Context Manager process to determine the next routing decision. Valid values are as follows:<br>0 = Information request complete<br>1 = Transaction complete | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | 2 = Transaction declined<br>3 = Transaction reversed<br>4 = Transaction reversed, need acknowledgement | |
| 57–72 | 02 | TERTIARY-SVC-DEST [SYM-NAME]<br>The name of the destination associated with the tertiary service request. The data for this field is obtained from the Split Transaction Routing File (STRF). This field is reserved for future use. | PIC X(16) |
| 73–107 | 02 | OFFLINE-AUTH-FNAME [FNAME]<br>The filename of the Inventory File associated with the request. The data for this field is obtained from the Split Transaction Routing File (STRF) and is blank-filled if not available. | PIC X(35) |
| 108 | 02 | ORIG-FUNDS-RESPONDER<br>The STM.RTE.RESPONDER from the response message received from the FUNDS-AUTH-DEST. This value is used to restore original data to the FUNDS-AUTH-DEST if required for a subsequent reversal. | PIC X |
| 109–111 | 02 | ORIG-FUNDS-RESP-CDE<br>The STM.RQST.RESP from the response message received from the FUNDS-AUTH-DEST. This field is used to restore original data to the FUNDS-AUTH-DEST if required for a subsequent reversal. | PIC X(3) |
| 112–158 | 02 | USER-FLD-ACI<br>Reserved for future use. | PIC X(47) |

# Token BS Pre-Pay Switch Token—Binary Format

The Pre-Pay Switch token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–450 | | PRE-PAY-SWI-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data, including the FIID and the data present in the data buffer. | TYPE BINARY 16 |
| 3–450 | 02 | BUF<br>The generic data variable length up to 448 characters. | PIC X(448) |
| 3–30 | | BCGI-TKN<br>The following fields contain data specific to the BCGI Interface. | REDEFINES BUF |
| 3 | 02 | ACCT-TYP<br>Identifies the BCGI account ID type used in the XML message. Valid values are as follows:<br>0 = Phone<br>1 = Card (for future use)<br>2 = IMSI (for future use)<br>3 = Min (for future use)<br>4 = Carrier | PIC X |
| 4–5 | 02 | RESP-IND<br>Identifies the type of response the Telco Interface most recently received from the telco host. The type of response allows the interface to determine the next action when a | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | subsequent request is received. Valid values vary according to the telco host. The valid values for the BCGI host are as follows:<br>C1 = Cash replenishment inquiry response.<br>C2 = Cash replenishment inquiry response, available amounts present.<br>C3 = Cash replenishment inquiry response, taxes present.<br>P1 = PIN support inquiry response, carrier list present.<br>R1 = Replenish account response. | |
| 6–10 | 02 | ERR-CDE<br>Identifies the error returned when the transaction fails. This error code is used to map the telco specific error into a BASE24 internal error. | PIC X(5) |
| 11–19 | 02 | REF-NUM [ID-NUM]<br>The internal card identifier. | PIC 9(11) |
| 20–29 | 02 | POSTAL-CDE<br>Identifies the postal code for the terminal. | PIC X(10) |
| 30 | 02 | USER-FLD-ACI<br>Reserved for future use. | PIC X(1) |

# Token BS Pre-Pay Switch Token—ASCII Format

The fields in the ASCII format of the Pre-Pay Switch token are shown below. Refer to the binary definition of the Pre-Pay Switch token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–452 | | PRE-PAY-SWI-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X |
| 5–452 | 02 | BUF | PIC X(448) |
| 5–32 | | BCGI-TKN | REDEFINES BUF |
| 5 | 02 | ACCT-TYP | PIC X |
| 6–7 | 02 | RESP-IND | PIC X(2) |
| 8–12 | 02 | ERR-CDE | PIC X(5) |
| 13–21 | 02 | REF-NUM [ID-NUM] | PIC 9(11) |
| 22–31 | 02 | POSTAL-CDE | PIC X(10) |
| 32 | 02 | USER-FLD-ACI | PIC X(1) |

# Token BT Pre-Pay Response Token—Binary Format

The Pre-Pay Response token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–50 | | PRE-PAY-RESP-TKN | |
| 1–8 | 02 | TAX-AMT<br>The tax amount to be applied to the top-up transaction. This field is available for printing on receipt and can be viewed using TLF perusal. | TYPE BINARY 64 |
| 9 | 02 | RESP-IND<br>An indicator specifying the pre-pay notification process required by the Device Handler process. This value is provided by the Telco Interface. It is received by the Device Handler process in a 9909 message. Valid values are as follows:<br>␣ = Information request complete (where ␣ indicates a blank space).<br>1 = The Device Handler process presents a tax notification to the cardholder.<br>2 = The Device Handler process presents a telecommunication provider list to the cardholder and after the carrier is selected, the Device Handler process presents an available amounts list to the cardholder.<br>3 = The Device Handler process presents an available amounts list to the cardholder. | PIC X |
| 10–14 | 02 | CARRIER-CDE<br>The telecommunication provider code used by the mobile network to identify a specific carrier. | PIC X(5) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 15–34 | 02 | CARRIER-NAM | PIC X(20) |
| | | The telecommunication provider name supplied to the customer. This field is left justified and filled with spaces. This field is available for printing on receipts. | |
| 35–49 | 02 | CARRIER-PHN-NUM | PIC X(15) |
| | | The telecommunication provider's phone number which is supplied to the customer. This field is left justified and filled with spaces. This field is available for printing on receipts. | |
| 50 | 02 | USER-FLD-ACI | PIC X(1) |
| | | Reserved for future use. | |

# Token BT Pre-Pay Response Token—ASCII Format

The fields in the ASCII format of the Pre-Pay Response token are shown below. Refer to the binary definition of the Pre-Pay Response token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–62 | | PRE-PAY-RESP-TKNX | |
| 1–19 | 02 | TAX-AMT | PIC X(19) |
| 20 | 02 | RESP-IND | PIC X |
| 21–25 | 02 | CARRIER-CDE | PIC X(5) |
| 26–45 | 02 | CARRIER-NAM | PIC X(20) |
| 46–60 | 02 | CARRIER-PHN-NUM | PIC X(15) |
| 61 | 02 | USER-FLD-ACI | PIC X(1) |
| 62 | 02 | USER-FLD1 | PIC X(1) |

# Token BU Pre-Pay Selection Token—Binary Format

The content of the Pre-Pay Selection token varies based upon the type of response recevied from the host. The Pre-Pay Selection token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–614 | | PRE-PAY-SELCT-TKN | |
| 1–2 | 02 | FRMT-CDE<br>Identifies the format of the token contents. Valid values are as follows:<br>00 = No format specified<br>01 = Amount list<br>02 = Telecommunication provider list | PIC X(2) |
| 3–4 | 02 | LGTH<br>Identifies the length of the information field. | TYPE BINARY |
| 5–614 | 02 | INFO | PIC X(610) |
| 5–614 | 02 | AMT-LIST<br>Available amounts returned in a response from the host. This situation occurs when the amount originally requested was unavailable. | REDEFINES INFO |
| 5–6 | 04 | NUM-AMT<br>The number of entries in the AMT fields. | TYPE BINARY |
| 7–54 | 04 | AMT<br>The array of top-up amounts available to the cardholder. | TYPE BINARY 64<br>OCCURS 6 TIMES |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 55–614 | 04 | RESERVED | PIC X(560) |
| 5–614 | 02 | CARRIER-LIST<br>Available telecommunication providers and associated amounts returned in a response from the host. This situation occurs when the cardholder's home telecommunication provider cannot be determined. | REDEFINES INFO |
| 5–6 | 04 | NUM-CARRIER<br>The number of telecommunication provider entries in the carrier table. | TYPE BINARY UNSIGNED |
| 7–614 | 04 | CARRIER | TYPE CARRIER-ENTRY<br>OCCURS 0 TO 8 TIMES DEPENDING ON NUM -CARRIER |
| | 06 | CDE<br>The telecommunication provider code assigned by the telco. | PIC X(5) |
| | 06 | USER-FLD-ACI<br>Reserved for future use. | PIC X |
| | 06 | NAM<br>The telecommunication provider name associated with the carrier code. This field is left justified and filled with spaces. | PIC X(20) |
| | 06 | NUM-AMT<br>The number of entries in the AMT field. | TYPE BINARY |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 08 | AMT | TYPE BINARY 64<br>OCCURS 0 TO 6 TIMES<br>DEPENDING ON NUM-AMT |
| | | The array of top-up amounts available to the cardholder for a specific telecommunication provider. | |

# Token BU Pre-Pay Selection Token—ASCII Format

The fields in the ASCII format of the Pre-Pay Selection token are shown below. Refer to the binary definition of the Pre-Pay Selection token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–1228 | | PRE-PAY-SELCT-TKNX | |
| 1–2 | 02 | FRMT-CDE | PIC X(2) |
| 3–6 | 02 | LGTH | PIC X(4) |
| 7–1228 | 02 | INFO | PIC X(1222) |
| 7–1228 | 02 | AMT-LIST | REDEFINES INFO |
| 7–11 | 04 | NUM-AMT | PIC 9(5) |
| 12 | 04 | USER-FLD1 | PIC X |
| 13–114 | 04 | AMOUNT | OCCURS 6 TIMES |
| 115–133 | 06 | AMT | PIC X(19) |
| 132 | 06 | USER-FLD2 | PIC X |
| 133–1228 | 04 | RESERVED | PIC X(1096) |
| 7–1228 | 02 | CARRIER-LIST | REDEFINES INFO |
| 7–11 | 04 | NUM-CARRIER | PIC 9(5) |
| 12 | 04 | USER-FLD3 | PIC X |
| 13–1228 | 04 | CARRIER | TYPE CARRIER-ENTRYX OCCURS 0 TO 8 TIMES DEPENDING ON NUM-CARRIER |
| | 06 | CDE | PIC X(5) |
| | 06 | USER-FLD-ACI | PIC X |
| | 06 | NAM | PIC X(20) |
| | 06 | NUM-AMT | PIC 9(5) |
| | 06 | USER-FLD4 | PIC X |
| | 06 | AMOUNT | OCCURS 0 TO 6 TIMES DEPENDING ON NUM-AMT |
| | 08 | AMT | PIC X(19) |
| | 08 | USER-FLD5 | PIC X |

# Token BV Pre-Pay Voucher Receipt Token—Binary Format

The Pre-Pay Voucher Receipt token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–152 | | PRE-PAY-VCHR-RCPT-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. | TYPE BINARY 16 |
| 3–152 | 02 | BUF<br>The generic data. This field is variable length up to 150 characters. | PIC X(150) |
| 3–152 | 02 | VCHR-FRMT1<br>Voucher Format 1 represents the first pre-pay receipt format supported in BASE24 for voucher transactions. | REDEFINES BUF |
| 3–13 | 04 | BTCH-ID [ID-NUM]<br>A value that identifies the order of the card in the batch load. This field is available for printing on receipts. | PIC 9(11) |
| 14–24 | 04 | CNTL-NUM [ID-NUM]<br>The internal card identifier. This field is available for printing on receipts. | PIC 9(11) |
| 25–28 | 04 | EXP-DAYS<br>The number of days the account expiration date is extended after this replenishment. This field is available for printing on receipts. | PIC X(4) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 29–47 | 04 | NUM<br>Prepaid card identification number used for replenishment of an account not enabled with real-time recharge. This field is available for printing on receipts. | PIC X(19) |
| 48–53 | 04 | SHELF-DAT<br>The date, in YYMMDD format, after which the card used in this transaction can no longer be used. This field is available for printing on receipts. | [DAT] |
| 48–49 | 06 | YY | PIC X(2) |
| 50–51 | 06 | MM | PIC X(2) |
| 52–53 | 06 | DD | PIC X(2) |
| 54–152 | 04 | USER-FLD-ACI<br>Reserved for future use. | PIC X(99) |

# Token BV Pre-Pay Voucher Receipt Token—ASCII Format

The fields in the ASCII format of the Pre-Pay Voucher Receipt token are shown below. Refer to the binary definition of the Pre-Pay Voucher Receipt token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–154 | | PRE-PAY-VCHR-RCPT-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X |
| 5–154 | 02 | BUF | PIC X(150) |
| 5–154 | 02 | VCHR-FRMT1 | REDEFINES BUF |
| 5–15 | 04 | BTCH-ID [ID-NUM] | PIC 9(11) |
| 16–26 | 04 | CNTL-NUM [ID-NUM] | PIC 9(11) |
| 27–30 | 04 | EXP-DAYS | PIC X(4) |
| 31–49 | 04 | NUM | PIC X(19) |
| 50–55 | 04 | SHELF-DAT | [DAT] |
| 50–51 | 06 | YY | PIC X(2) |
| 52–53 | 06 | MM | PIC X(2) |
| 54–55 | 06 | DD | PIC X(2) |
| 56–154 | 04 | USER-FLD-ACI | PIC X(99) |

# Token BW Pre-Pay Online Receipt Token—Binary Format

The Pre-Pay Online Receipt token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–152 | | PRE-PAY-ONL-RCPT-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. | TYPE BINARY 16 |
| 3–152 | 02 | BUF<br>The generic data variable length up to 150 characters. | PIC X(150) |
| 3–152 | 02 | ONL-FRMT1<br>Online Format 1 represents the first pre-pay receipt format supported in BASE24. | REDEFINES BUF |
| 3–8 | 04 | EXP-DAT<br>The expiration date (YYMMDD). Available for printing on receipts. | TYPE DAT |
| 9–27 | 04 | NUM<br>Prepaid card identification number used for replenishment. This field may not be present for every transaction. | PIC X(19) |
| 28–152 | 04 | USER-FLD-ACI<br>Reserved for future use. | PIC X(125) |

# Token BW Pre-Pay Online Receipt Token—ASCII Format

The fields in the ASCII format of the Pre-Pay Online Receipt token are shown below. Refer to the binary definition of the Pre-Pay Online Receipt token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–154 | | PRE-PAY-ONL-RCPT-TKNX | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4 | 02 | USER-FLD1 | PIC X |
| 5–154 | 02 | BUF | PIC X(150) |
| 5–154 | 02 | ONL-FRMT1 | REDEFINES BUF |
| 5–10 | 04 | EXP-DAT | TYPE DAT |
| 11–29 | 04 | NUM | PIC X(19) |
| 30–154 | 04 | USER-FLD-ACI | PIC X(125) |

# Token BX Pre-Pay Original Data Token

The Pre-Pay Original Data token contains data from an original transaction that can be modified during pre-pay processing. This data is restored to the appropriate fields in the internal message before returning a response to the transaction originator. The fields in the Pre-Pay Original Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–66 | | PRE-PAY-ORIG-DATA-TKN | |
| 1–40 | 02 | TRK2<br>Contains the original track 2 data. | PIC X(40) |
| 41–42 | 02 | TXN-CDE<br>Contains the original transaction code. | PIC X(2) |
| 43–46 | 02 | CRD-FIID<br>Contains the original card-owning FIID. | PIC X(4) |
| 47-48 | 02 | SRV<br>For POS, contains the original PSTM.RTE.SRV. | PIC X(2) |
| 49–66 | 02 | USER-FLD-ACI<br>Reserved for future use. | PIC X(18) |

# Token BY Switch Common Data Token

The Switch Common Data token contains additional fields that can be accessed by other processes. This enhancement adds the System Trace Audit Number (STAN), Retrieval Reference Number, and Debit Sharing/Network ID Code in the response message for the BASE24-pos Standard POS Device Handler to improve PIN-Debit reconciliation. The fields of the Switch Common Data token are detailed below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–60 | | SWI-CMN-DATA-TKN | |
| 1–6 | 02 | STAN<br>System Trace Audit Number. | PIC X(6) |
| 7–18 | 02 | RETRVL-REF-NUM<br>Retrieval Reference Number. This field will be left-justified and space-filled. | PIC X(12) |
| 19–22 | 02 | NETWK-ID<br>Network ID. This field can contain between two and four characters. | PIC X(4) |
| 23–25 | 02 | RESP-CDE<br>External Response Code. This field is left justified and space filled. It can contain two or three characters. | PIC X(3) |
| 26 | 02 | ISA-IND<br>International Service Assessment (ISA) Indicator. This indicator notes whether or not a transaction is eligible for an ISA fee or not. Valid values for this indicator are as follows:<br>0 or ␣ = Not ISA fee eligible (where ␣ indicates a blank space).<br>1 = ISA fee eligible (debit). | PIC X |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | C = Single currency ISA assessed.<br>S = Multi-currency ISA assessed.<br>R = ISA fee eligible (credit). | |
| 27–34 | 02 | ISA-AMT<br>International Service Assessment (ISA) Amount. | PIC X(8) |
| 35–37 | 02 | FEE-PGM-IND<br>Fee Program Indicator (FPI). This field assists endpoints in determining interchange fees. | PIC X(3) |
| 38 | 02 | ON-OFF-PREM-IND<br>ON/OFF Premise Indicator. Valid values for this indicator are as follows:<br>0 = On premise<br>1 = Off premise | PIC X |
| 39 | 02 | CROSS-BORDER-TXN-IND<br>Cross-border Transaction Indicator. Valid values are as follows:<br>Y = Qualifies as a cross-border transaction.<br>N = Does not qualify as a cross-border transaction. | PIC X |
| 40 | 02 | CROSS-BORDER-CRNCY-IND<br>Cross-border Currency Indicator. Valid values are as follows:<br>X = Transaction does not qualify as a cross-border transaction.<br>Y = Transaction was submitted in local currency of merchant.<br>N = Transaction was not submitted in local currency of merchant. | PIC X |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 41–52 | 02 | CRNCY-CONV-ASSESS-AMT | PIC X(12) |

The Currency Conversion Assessment amount (MCCR).

| | | | |
|---|---|---|---|
| 53–54 | 02 | CRD-LVL-PROD-ID-VAL | PIC X(2) |

The Card Level Product ID value. Valid values are as follows:

A = Visa Traditional Non-Rewards
AX = American Express
B = Visa Traditional Rewards/MasterCard Enhanced (Consumer)
C = Visa Signature/MasterCard Consumer World
D = Visa Signature Preferred/MasterCard Consumer World Elite
DI = Discover
G = Visa Business/MasterCard Business World
G1 = Visa Signature Business
G2 = Visa Business Check Card
H = Visa Check Card/MasterCard Business World Elite
I = Visa Commerce/MasterCard Corporate World
J = MasterCard Corporate World Elite
J1 = Visa General Prepaid
J2 = Visa Prepaid Gift
J3 = Visa Prepaid Healthcare
J4 = Visa Prepaid Commercial
K = Visa Corporate
K1 = Visa GSA Corporate T & E
M = MasterCard/Euro Card and Diners
Q = Private Label
Q1 = Private Label Prepaid
R = Proprietary
S = Visa Purchasing
S1 = Visa Purchasing with Fleet
S2 = Visa GSA Purchasing
S3 = Visa GSA Purchasing with Fleet
U = Visa TravelMondy
Z = Does not participate (MasterCard default value)

| | | | |
|---|---|---|---|
| 55–60 | 02 | USER-FLD-ACI | PIC X(6) |

Reserved by ACI for future use.

# Token M1 Migration ATM Data1 Token

The Migration ATM Data1 token contains fields that need to be passed over to BASE24-eps for processing by interchange interface processes that have been migrated from BASE24.

This token is created by the BASE24-atm Authorization process and is passed to BASE24-eps via a BASE24 ISO 87 Host Interface Bridge process. It allows data that is present in the STM (but not in the SEM) to be passed to BASE24-eps so that it can be used in processing. The Migration ATM Data1 token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | MIGR-ATM-DATA1-TKN | |
| 1 | 02 | TERM-CAPTURE-FLAG<br>This field indicates whether the terminal has the capability to capture the card. Valid values are as follows:<br>0 = Terminal can capture card (default)<br>1 = Terminal cannot capture card | PIC X(1) |
| 2–20 | 02 | USER-FLD-ACI<br>Reserved by ACI for future use. | PIC X(19) |

# Token M2 Migration POS Data1 Token

The Migration POS Data1 token contains fields that need to be passed over to BASE24-eps for processing by interchange interface processes that have been migrated from BASE24.

This token is created by the BASE24-pos Router/Authorization process and is passed to BASE24-eps via a BASE24 ISO 87 Host Interface Bridge process. It allows data that is present in the PSTM (but not in the SEM) to be passed to BASE24-eps so that it can be used in processing. The Migration POS Data1 token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | MIGR-POS-DATA1-TKN | |
| 1–2 | 02 | TERM-TYP | PIC X(2) |

The type of terminal from which the transaction originated. Standard BASE24-pos codes are as follows:

41 = ACI Standard POS Device Handler
43 = NCR 2123 and NCR 2126
44 = NCR 2127
45 = NCR 7000
70 = Visa I, APACS 70 (Auth Only)
71 = Visa II Dialup, APACS 70
78 = Merchant Host Interface
D0 = Hypercom TDS
D1 = Custom POS device 1
D2 = Custom POS device 2
D3 = Custom POS device 3
D4 = Custom POS device 4
D5 = Custom POS device 5
D6 = BASE24 SPDH (fuel)
D7 = BASE24 SPDH (electronic cash register)
D8 = BASE24 SPDH (script dispenser)

| | | | |
|---|---|---|---|
| 3–20 | 02 | USER-FLD-ACI | PIC X(18) |

Reserved by ACI for future use.

# Token M4 Migration EPS HISO Token

The Migration EPS HISO token contains an indicator that Authorization processes can use to determine whether a transaction about to be written to the (P)TLF has been processed by BASE24-eps. This token is added by the HISO process if its MIGR-HISO-ENABLE LCONF Param is set to true, indicating it is acting as a migration HISO process. Fields in the Migration EPS HISO token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–2 | | MIGR-EPS-HISO-TKN | |
| 1 | 02 | MIGRATION-IND<br>A flag indicating whether the transaction has been processed by a HISO acting as a Migration HISO during the migration of BASE24 to BASE24-eps. Valid values are:<br>Y = Transaction has been processed by a Migration HISO<br>N = Transaction has not been processed by a Migration HISO | PIC X(1) |
| 2 | 02 | USER-FLD-ACI<br>Reserved for future use. | PIC X(1) |

# Token M5 Migration Customer Data Token

The Migration Customer Data token contains fields that need to be passed over to BASE24-eps for processing by interchange interface processes that have been migrated from BASE24. All fields within this token are initialized to blanks. Fields in the Migration Customer Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–80 | | MIGR-CUST-DATA-TKN | |
| 1–2 | 02 | CUST-ID-TYP<br>The Customer ID Code Type used to represent the customer. Valid values are user-defined. | PIC X(2) |
| 3–38 | 02 | CUST-ID<br>The value of the Customer ID Code used to represent the customer. | PIC X(36) |
| 39–40 | 02 | BANK-ID-TYP<br>The Bank/Branch ID Code Type associated with the account number. Valid values are as follows:<br>00 = none<br>01 = IBAN<br>02 = BBAN<br>03 = BIC<br>04 = RTTN<br>05-98 = Reserved for future use<br>99 = Unknown | PIC X(2) |
| 42–74 | 02 | BANK-ID<br>The value of the Bank/Branch Code associated with the account number. | PIC X(34) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 75-80 | 02 | PROC-CDE | PIC X(6) |
| | | The value of the ISO 8583 processing code associated with the transaction. | |

# Token N8 Inventory Voucher Token—Binary Format

The Inventory Voucher token contains information associated with the purchase of top-up vouchers. This token is created by the acquiring BASE24 process when a voucher purchase transaction is received. The Inventory Voucher token is described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–54 | | INVTRY-VCHR-TKN | |
| 1–4 | 02 | ACQ-STOCK-CDE | PIC X(4) |
| | | The stock code known by the inventory manager. This field is formatted by the acquiring BASE24 process. | |
| 5–8 | 02 | ISS-STOCK-TYP | PIC X(4) |
| | | The stock type code known by the issuer (telco). This field is formatted by the acquiring BASE24 process. | |
| 9–16 | 02 | STOCK-VAL | TYPE BINARY 64 |
| | | The denomination of the voucher, in whole current units. This field is formatted by the acquiring BASE24 process. | |
| 17–28 | 02 | DLVR-SEQ-NUM | PIC X(12) |
| | | The sequence number assigned to this transaction by the acquiring process. The value in this field is left justified and space filled. This field is formatted by the acquiring BASE24 process. | |
| 29–44 | 02 | STOCK-NUM | PIC X(16) |
| | | The control or serial number associated with the stock item. The value in this field is left justified and is available for printing on receipts. This field is formatted by the inventory manager on the response. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 45–52 | 02 | STOCK-EXP-DAT | PIC X(8) |
| | | The expiration date of the stock item, in CCYYMMDD format. The value in this field is available for printing on receipts. This field is formatted by the inventory manager on the response. | |
| 53 | 02 | RVSL-TYP | PIC X |
| | | The reversal type for the stock item. The inventory manager uses the value in this field to determine whether the stock item can be made available for resale. This value is formatted by the acquiring BASE24 process. Valid values are as follows:<br>0 = Not specified. If a reversal, treat as normal reversal.<br>1 = Normal reversal. Stock item can be made available for resale.<br>2 = Doubtful reversal. Stock item should not be made available for resale. | |
| 54 | 02 | USER-FLD-ACI | PIC X |
| | | Reserved for future use. | |

# Token N8 Inventory Voucher Token—ASCII Format

The fields in the ASCII format of the Inventory Voucher token are shown below. Refer to the binary definition of the Inventory Voucher token for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–66 | | INVTRY-VCHR-TKNX | |
| 1–4 | 02 | ACQ-STOCK-CDE | PIC X(4) |
| 5–8 | 02 | ISS-STOCK-TYP | PIC X(4) |
| 9–27 | 02 | STOCK-VAL | PIC X(19) |
| 28–39 | 02 | DLVR-SEQ-NUM | PIC X(12) |
| 40–55 | 02 | STOCK-NUM | PIC X(16) |
| 56–63 | 02 | STOCK-EXP-DAT | PIC X(8) |
| 64 | 02 | RVSL-TYP | PIC X |
| 65 | 02 | USER-FLD-ACI | PIC X |
| 66 | 02 | USER-FLD1 | PIC X |

# Token S0 Intra Country Data Token

The fields in the Intra Country Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–390 | | INTRA-CNTRY-DATA-TKN | |
| 1–2 | 02 | FRMT-CDE<br>The format code identifies the format for the contents of the token. Valid values are as follows:<br>01 = Visa intra country data for Japan. | PIC X(2) |
| 3–390 | 02 | INFO<br>This field is redefined, based on the value of the format code. | PIC X(388) |
| 3–390 | 02 | VISA | REDEFINES INFO |
| 3–254 | 04 | INTRA-CNTRY-DATA<br>Visa field 118, Intra Country Data | |
| 3–6 | 06 | CNTRY-CDE | PIC X(4) |
| 7–255 | 06 | PRVT-USE-FLD | PIC X(249) |
| 256–390 | 04 | FREE-FORM-TXT<br>Visa field 117. National Use Data. | PIC X(135) |

# Token S1 Gateway Info Token

The Gateway Info token is used to explicitly reference datasets and fields that may be included in the token. Multiple occurrences of a single dataset are possible, as well as multiple types of datasets. The datasets present in the token may be in any order and may include duplicates. The maximum size for this token is 202 characters.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–202 | | GTWY-INFO-TKN | PIC X(202) |
| 1–2 | 02 | VER-ID | PIC X(2) |
| | | This field indicates the current version of the Gateway Info token. Processes that add the Gateway Info token will need to increment this value whenever one of the INFO definitions that follow is modified. The current version of GTWY-INFO-TKN is 04. | |
| 3–200 | 02 | GTWY-INFO | PIC X(200) |
| | | The following definitions are used to further define the data in the Gateway Info token. There may be 1–*n* occurrences of each dataset. They may occur in any order. | |
| | | VISA-GTWY-AX-INFO | DEFINITION |
| | | This definition is used to explicitly reference the Visa to American Express fields in the Gateway Info token. The length of this definition is 14 characters. | |
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for the VISA-GTWY-AX-INFO is 66. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | POS-DATA-CDE | PIC X(12) |
| | | This field is created by V.I.P. as part of the VisaNet Gateway mapping service, and will contain data for American Express field 22 POS Data code. | |
| | | VISA-GTWY-MC-INFO | DEFINITION |
| | | This definition is used to explicitly reference the Visa to MasterCard fields in the Gateway Info token. The length of this definition is 44 characters. | |
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for the VISA-GTWY-MC-INFO is 67. | |
| | 02 | POS-ENTRY-MDE | PIC X(3) |
| | | This field contains data from the MasterCard CIS data element 22 in MasterCard authorization request messages. | |
| | 02 | POS-PIN-CAPTR-CDE | PIC X(2) |
| | | This field contains data from the MasterCard CIS data element 26 in MasterCard authorization request messages. | |
| | 02 | MC-POS-DATA | PIC X(26) |
| | | This field contains data from the MasterCard CIS data element 61 in MasterCard authorization request messages. | |
| | 02 | ADVC-DAT-TIM | PIC X(10) |
| | | This field contains data from the MasterCard CIS data element 48.15 in MasterCard authorization advice messages. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | ECOMM-IND | PIC X(7) |
| | | This field contains data from the MasterCard CIS data element 48.42 - Electronic Commerce Indicators. | |
| | | VISA-GTWY-DSCV-INFO | DEFINITION |
| | | This definition is used to explicitly reference the Visa to Discover fields in the Gateway Info token. The length of this definition is 78 characters. | |
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for the VISA-GTWY-MC-INFO is 68. | |
| | 02 | NTWK-INFO | PIC X(29) |
| | | This field is created by V.I.P. as part of theVisaNet Gateway mapping service, and will contain the network information from Discover. | |
| | 02 | TXN-QUAL | PIC X(46) |
| | | This field is created by V.I.P, as part of the VisaNet Gateway mapping service, and will contain the transaction qualifier from Discover, | |
| | 02 | USER-FLD-ACI | PIC X(1) |
| | | Reserved by ACI for future use. | |

# Token S2 Dynamic Currency Conversion Status Token

The Dynamic Currency Conversion (DCC) Status token contains core information needed for DCC-eligible transactions. The Device Handler process creates this token when DCC applies to the transaction in progress. This token is updated by the Device Handler process and the Authorization process during DCC processing. The fields in the Dynamic Currency Conversion Status token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–24 | | DCC-STAT-TKN | |
| 1–16 | 02 | DCC-PRFL<br>The terminal DCC profile. The Device Handler process fills this field from the corresponding field in the ATD. The Authorization process uses the value in this field to locate terminal-specific data in the Dynamic Currency Conversion Data File (DCCD). | TYPE PRFL |
| 17 | 02 | DCC-STAT<br>The current status of DCC processing. Valid values are as follows:<br>␣ = Indicates that DCC has not yet been performed. Set by the Device Handler process (where ␣ indicates a blank space).<br>A = Indicates that the cardholder agreed to the conversion. The transaction should be sent to the interchange in the home currency of the cardholder. Set by the Device Handler process.<br>C = Indicates that the Authorization process converted the transaction from the dispensed currency to the BIN currency. Set by the Authorization process.<br>D = Indicates that the cardholder declined the conversion. The transaction should be sent to the interchange in the original currency. Set by the Device Handler process. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

N = Indicates that the Authorization process performed the DCCD lookup and determined that DCC is not applicable (no BIN currency record was found for the BIN, the BIN currency is the same as the terminal currency, or no currency data was found for the combination of terminal, issuer, and BIN currency).

R = Indicates that the Authorization process restored the transaction from the BIN currency to the dispensed currency. Set by the Authorization process.

S = Indicates that the Authorization process sent the DCC notification data to the Device Handler process. Set by the Authorization process.

X = Indicates that the cardholder cancelled the transaction during DCC notification. Set by the Device Handler process. Sent to the Authorization process to be logged.

18 02 DCC-SUBSTAT PIC X

The substatus field contains additional detail about the status of DCC processing. Valid values are as follows:

| Status | Substatus | Definition |
|---|---|---|
| Any | ~~b~~ | No additional detail (where ~~b~~ indicates a blank space). |
| Any | * | An error occurred during the stage of processing indicated by the status. DCC was not offered. The transaction may still complete successfully without DCC. |
| S | 0 | The Device Handler process has delayed DCC notification pending surcharge notification. |
| S | 1–8 | The Device Handler process has sent the first (second, etc.) DCC notification screen to the terminal. |
| S | 9 | The Device Handler process has sent the final DCC notification screen to the terminal. |

# Token S3 Dynamic Currency Conversion Processing Token—Binary Format

The Dynamic Currency Conversion (DCC) Processing token contains information needed to perform DCC notification and conversion. The Authorization process creates and maintains this token if DCC should be offered to the cardholder on the transaction in progress. This token is read by the Device Handler process and the Authorization process. The fields in the Dynamic Currency Conversion Processing token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–154 | | DCC-PROC-TKN | |
| 1–2 | 02 | ISS-DESIGNATOR | PIC X(2) |
| | | A code designating the issuer to whom the transaction will be or has been routed. Values are user-defined; however, BASE24 assumes that codes beginning with M represent MasterCard brands (i.e., MasterCard, Cirrus, etc.) and codes beginning with V represent Visa brands.<br><br>The Authorization module sets this value from the data in the Dynamic Currency Conversion Data File (DCCD). The Device Handler process may use the value in this field to ensure, for example, that the proper issuer-specific information is printed on the receipt. | |
| 3–5 | 02 | BIN-CRNCY-CDE | TYPE CRNCY-CDE |
| | | The currency code associated with the Bank Identification Number (BIN), which is the first six digits of the PAN. BASE24 examines up to 12 digits to enable more precise groupings within the same BIN. The Authorization process sets this value from the data in the DCCD. | |
| 6–25 | 02 | BIN-CRNCY-DESCR | PIC X(20) |
| | | A description of the BIN currency. For example, “UK POUNDS” or “GBP.” The Authorization process sets this value from the data in the DCCD. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 26–33 | 02 | CNV-RATE-NEUTRAL | PIC 9(8) |
| | | The neutral conversion rate. The Authorization process sets this value to the multiplier required to convert the transaction amount from the dispensed currency to the BIN currency, as calculated from conversion rates in the Exchange Rate File (ERF), with no mark-up applied.<br><br>This field is in ISO format; i.e., the leftmost digit denotes the number of decimal places and positions 2–8 give the rate. For example, 69972522 = 9.972522. | |
| 34–41 | 02 | CNV-RATE-EFFECTIVE | PIC 9(8) |
| | | The effective conversion rate. The Authorization process sets this value to the neutral conversion rate, increased by the mark-up.<br><br>This field is in ISO format; i.e., the leftmost digit denotes the number of decimal places and positions 2–8 give the rate. For example, 69972522 = 9.972522. | |
| 42–73 | 02 | CNV-RATE-SRC | PIC X(32) |
| | | The authority from whom the conversion rate is derived. The Authorization process sets this value from the data in the Logical Network Configuration File (LNCF). The Device Handler process may use this information in formatting receipts or screen data. | |
| 74–81 | 02 | CNV-RATE-DAT | PIC X(8) |
| | | The date, in YYYYMMDD format, on which the conversion rate was obtained. The Authorization process sets this value from the ERF data. | |
| 82–84 | 02 | SAVED-CRNCY-CDE | TYPE CRNCY-CDE |
| | | The saved currency code; the currency of the transaction as received by the Authorization process from the Device Handler process. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 85–86 | 02 | PCNT-MARK-UP | TYPE BINARY 16 |
| | | The percent mark-up, expressed in hundredths of a percent, that is provided for the acquirer on the conversion (i.e., 100 = 1%). The Authorization process sets this value from the data in the DCCD. | |
| 87–94 | 02 | CONVERTED-AMT-1 | TYPE BINARY 64 |
| | | Amount 1 in the BIN currency. The Authorization process determines this value by multiplying the amount 1 in the dispensed currency by the effective conversion rate. | |
| 95–102 | 02 | CONVERTED-AMT-2 | TYPE BINARY 64 |
| | | Amount 2 in the BIN currency. The Authorization process determines this value by multiplying the amount 2 in the dispensed currency by the effective conversion rate. | |
| 103–110 | 02 | SAVED-AMT-1 | TYPE BINARY 64 |
| | | Saved amount 1. The amount 1 of the transaction as received by the Authorization process from the Device Handler process. | |
| 111–118 | 02 | SAVED-AMT-2 | TYPE BINARY 64 |
| | | Saved amount 2. The amount 2 of the transaction as received by the Authorization process from the Device Handler process. | |
| 119–126 | 02 | CONVERTED-ACQ-SURCH | TYPE BINARY 64 |
| | | The acquirer surcharge in the BIN currency. | |
| 127–134 | 02 | CONVERTED-ISS-SURCH | TYPE BINARY 64 |
| | | The issuer surcharge in the BIN currency. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 135–154 | 02 | USER-FLD-ACI<br>Reserved by ACI for future use. | PIC X(20) |

# Token S3 Dynamic Currency Conversion Processing Token—ASCII Format

The fields in the ASCII format of the Dynamic Currency Conversion Processing token are shown below. Refer to the binary definition for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–244 | | DCC-PROC-TKNX | |
| 1–2 | 02 | ISS-DESIGNATOR | PIC XX |
| 3–5 | 02 | BIN-CRNCY-CDE | TYPE CRNCY-CDE |
| 6–25 | 02 | BIN-CRNCY-DESCR | PIC X(20) |
| 26–33 | 02 | CNV-RATE-NEUTRAL | PIC 9(8) |
| 34–41 | 02 | CNV-RATE-EFFECTIVE | PIC 9(8) |
| 42–73 | 02 | CNV-RATE-SRC | PIC X(32) |
| 74–81 | 02 | CNV-RATE-DAT | PIC X(8) |
| 82–84 | 02 | SAVED-CRNCY-CDE | TYPE CRNCY-CDE |
| 85–89 | 02 | PCNT-MARK-UP | PIC X(5) |
| 90–108 | 02 | CONVERTED-AMT-1 | PIC X(19) |
| 109–127 | 02 | CONVERTED-AMT-2 | PIC X(19) |
| 128–146 | 02 | SAVED-AMT-1 | PIC X(19) |
| 147–165 | 02 | SAVED-AMT-2 | PIC X(19) |
| 166–184 | 02 | CONVERTED-ACQ-SURCH | PIC X(19) |
| 185–203 | 02 | CONVERTED-ISS-SURCH | PIC X(19) |
| 204-244 | 02 | USER-FLD-ACI | PIC X(41) |

# Token S4 EMV Supplementary Data Token—Binary Format

The EMV Supplementary Data token is used to carry supplementary data associated with EMV transactions. The fields in the EMV Supplementary Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–84 | 00 | EMV-SUPPL-DATA-TKN | |
| 1–2 | 02 | DATASET-ID<br>The Dataset ID field indicates the specific usage of the data in the token buffer. Valid values are as follows:<br>00 = Used to carry generic data required by issuers.<br>01 = Used to carry the supplementary data required by issuers in a contactless transaction. | PIC X(2) |
| 3–4 | 02 | LGTH<br>Length of the data contained in the SUPPL-DATA field. | TYPE BINARY 16 |
| 5–84 | 02 | SUPPL-DATA<br>Specific data as indicated by the Dataset ID. | PIC X(80) |

# Token S4 EMV Supplementary Data Token—ASCII Format

The fields in the ASCII format of the EMV Supplementary Data token are shown below. Refer to the binary definition for field descriptions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–166 | | EMV-SUPPL-DATA-TKNX | |
| 1–2 | 02 | DATASET-ID | PIC X(2) |
| 3–6 | 02 | LGTH | PIC X(4) |
| 7–166 | 02 | SUPPL-DATA | PIC X(160) |

# Token S6 Track 2 Token

The Track 2 token contains the track 2 information that was used in the transaction. The fields in the Track 2 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–40 | | TRACK2-TKN | |
| 1–40 | 02 | TRACK2 | PIC X(40) |

The track 2 data, variable length up to 40 characters. This data includes the start sentinel and end sentinel. Track 2 is defined as follows:

| | | | |
|---|---|---|---|
| SS | Start Sentinel | 1 character | ; |
| PAN | Primary Account Number | Up to 19 digits | |
| FS | Field Separator | 1 character | = |
| ED | Expiration Date | 4 digits | YYMM format |
| SC | Service Code | 3 digits | |
| DD | Discretionary Data | Balance of available digits | |
| ES | End Sentinel | 1 character | ? |
| LRC | Longitudinal Redundancy Check | 1 character | |

# Token S7 Person-to-Person Transaction 2 Token

The fields in the Person-to-Person Transaction 2 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–400 | | P2P-TXN2-TKN | |
| 1–2 | 02 | FRMT-CDE<br>A code identifying the format of the token contents. Valid values are as follows:<br>00 = MasterCard person-to-person | PIC X(2) |
| 3–400 | 02 | INFO<br>This field will be redefined based on the FRMT-CDE. | PIC X(398) |
| 3–400 | 02 | MC<br>MasterCard format. | REDEFINES INFO |
| 3–201 | 04 | SENDER<br>The sender identification data which is provided by the Acquiring processor to support payment transactions. | |
| 3–21 | 06 | UNIQUE-REF-NUM<br>A unique reference number | PIC X(19) |
| 22–45 | 06 | NAM<br>Sender name, payer name, or user ID. | PIC X(24) |
| 46–136 | 06 | ADDR<br>Sender/payer address. | PIC X(91) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 137–156 | 06 | PHN-NUM<br>Telephone number. | PIC X(20) |
| 157–201 | 06 | MSG<br>Optional message. | PIC X(45) |
| 202-– | 04 | RCV<br>The recipient identification data, provided by the issuing processor to support payment transactions. | |
| 202–220 | 06 | UNIQUE-REF-NUM<br>Unique reference number. | PIC X(19) |
| 221–244 | 06 | NAM<br>Recipient name or user ID. | PIC X(24) |
| 245–335 | 06 | ADDR<br>Recipient address. | PIC X(91) |
| 336–355 | 06 | PHN-NUM<br>Telephone number. | PIC X(20) |
| 356–400 | 06 | MSG<br>Optional message. | PIC X(45) |

# Token S8 PAN Mapping Token

The PAN Mapping token is used by the MDS interface to include the cardholder's embossed PAN in a response to MDS when a contactless transaction is authorized by a backend host system. The fields in the PAN Mapping token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–30 | | PAN-MAPPING-TKN | |
| 1 | 02 | ACCT-NUM-IND<br>The type of PAN mapping account. Valid values are as follows:<br>E = Embossed account number<br>L = Loyalty program operator card<br>M = Primary account number<br>P = Proximity account number<br>R = Loyalty program card<br>V = Virtual card number | PIC X(1) |
| 2–20 | 02 | ACCT-NUM<br>The PAN mapping account number. | PIC X(19) |
| 21–24 | 02 | EXP-DAT<br>The expiration date of the PAN mapping account (YYMM). | PIC X(4) |
| 25–30 | 06 | USER-FLD-ACI | PIC X(6) |

# Token S9 Additional Authorization Data token

The base token ADNL-AUTH-DATA-TKN supports the additional authorization data that may conditionally be included in Visa field 48, usage 2. The token consists of a format code and an info field. The format code allows the token to be re-used for similar data in the future. The info field may be redefined for specific usages. The fields in the Additional Authorization Data token are defined below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–256 | 00 | ADNL-AUTH-DATA-TKN | |
| 1-2 | 02 | FRMT-CDE<br>Format Code identifies the format of the token contents. Valid values are:<br>01 = Visa MCC 6012 Additional Authorization Data | PIC X(2) |
| 3-256 | 02 | INFO<br>This field will be redefined depending on the format code. | PIC X(254) |
| 3-256 | 02 | VISA-MCC-6012<br>[REDEFINES INFO] | |
| 3-256 | 04 | FREE-FORM-TXT<br>Visa field 48, Usage 2. | PIC X(254) |

# Token SA Generic Data Token

The Generic Data token contains fields that need to be passed over to BASE24-eps for processing by interchange interface processes that have been migrated from BASE24. Currently this token is used for the RuPay interface. The fields in the Generic Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–118 | | GENERIC-DATA-TKN | |
| 1 | 02 | CRD-INPUT-CAP | PIC X(1) |

The card data input capability. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = Unknown
- 1 = Magnetic stripe read capability
- 2 = ICC capability
- 3 = Magnetic stripe and key entry capability
- 4 = Magnetic stripe and ICC capability
- 5 = Manual, no terminal
- 6 = Key entered

| | | | |
|---|---|---|---|
| 2 | 02 | CRDHLDER-AUTH-CAP | PIC X(1) |

The cardholder authentication capability. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = Unknown
- 1 = No electronic authentication
- 2 = PIN entry
- 3 = Biometric (currently not used)

| | | | |
|---|---|---|---|
| 3 | 02 | TERM-CRD-CAPTR | PIC X(1) |

The card capture capability. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = Unknown
- 1 = No capture capability
- 2 = Capture capability

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 4 | 02 | OPER-ENVMT | PIC X(1) |

The terminal operating environment. Valid values are as follows:

␣ = Not set (where ␣ indicates a blank space).
0 = Unknown
1 = On premises of card acceptor, attended
2 = On premises of card acceptor, not attended
3 = Off premises of card acceptor, attended
4 = Off premises of card acceptor, unattended
5 = On premises of cardholder, unattended
6 = No terminal used

| | | | |
|---|---|---|---|
| 5 | 02 | CRDHLDER-PRSN | PIC X(1) |

The cardholder present data. Valid values are as follows:

␣ = Not set (where ␣ indicates a blank space).
0 = Unknown
1 = Cardholder present
2 = Cardholder not present, unspecified reason
3 = Cardholder not present, mail transaction
4 = Cardholder not present, telephone transaction
5 = Cardholder not present, standing instruction
6 = E-commerce transaction
7 = IVR transaction

| | | | |
|---|---|---|---|
| 6 | 02 | CRD-PRSN | PIC X(1) |

The card present data. Valid values are as follows:

␣ = Not set (where ␣ indicates a blank space).
0 = Unknown
1 = Card not present
2 = Card present

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7 | 02 | CRD-INPUT-MDE | PIC X(1) |

The card present data. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = Unknown
- 1 = Manual input, no terminal
- 2 = Magnetic stripe read
- 3 = Online chip
- 4 = Offline chip
- 5 = E-commerce
- 6 = IVR
- 7 = Key entered

| | | | |
|---|---|---|---|
| 8 | 02 | CRDHLDER-AUTH-MTHD | PIC X(1) |

The cardholder authentication method. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = Unknown
- 1 = Not authenticated
- 2 = PIN
- 3 = Signature
- 4 = Biometric (FMR). Currently not used.
- 5 = OTP
- 6 = E-commerce type 1 PIN
- 7 = E-commerce type 1 OTP
- 8 = E-commerce type 2. Currently not used.
- 9 = IVR type 2. Currently not used.
- A = Biometric (FIR). Currently not used.
- B = Biometric (IIR). Currently not used.
- C = Biometric PIN. Currently not used.
- D = Biometric OTP. Currently not used.
- E = Biometric (FIT/FMR/IIR) with PIN. Currently not used.
- F = Biometric (FIT/FMR/IIR) with OTP. Currently not used.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9 | 02 | CRDHLDER-AUTH-ENTY | PIC X(1) |

The cardholder authentication entity. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = Unknown
- 1 = ICC
- 2 = CAD
- A = Type 1 (RuPay e-commerce implementation)
- C = Type 3 (3D if issuer opted for ICS 1 services). Currently not used.
- D = Type 4 (3D if issuer opted for ICS 2 services). Currently not used.
- I = Type 2 (3D if issuer opted for RuPay services). Currently not used.

| | | | |
|---|---|---|---|
| 10 | 02 | CRD-OUPUT-CAP | PIC X(1) |

The card data output capability. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = None
- 1 = Magnetic stripe write
- 2 = ICC write

| | | | |
|---|---|---|---|
| 11 | 02 | TERM-OUTPUT-CAP | PIC X(1) |

The terminal data output capability. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = Unknown
- 1 = Print capability
- 2 = Display capability
- 3 = Print and display capability

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 12 | 02 | PIN-CAPTR-CAP | PIC X(1) |

The PIN capture capability. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 0 = No PIN capture capability or unknown capability
- 1 = 4 characters maximum
- 2 = 5 characters maximum
- 3 = 6 characters maximum
- 4 = 7 characters maximum
- 5 = 8 characters maximum
- 6 = 9 characters maximum
- 7 = 10 characters maximum
- 8 = 11 characters maximum
- 9 = 12 characters maximum

| | | | |
|---|---|---|---|
| 13–14 | 02 | ECOMM-IND | PIC X(2) |

The e-commerce indicator. Valid values are as follows:

- ␣ = Not set (where ␣ indicates a blank space).
- 05 = Secure e-commerce with 3D. Currently not used.
- 06 = Not authenticated. Merchant attempted to authenticate using 3D secure.
- 07 = Nonsecure transactions with encrypted data.
- 08 = Nonsecure transactions.
- 15 = Secure e-commerce transaction registration required with OTP.
- 16 = Secure e-commerce transaction registration required with Internet banking.
- 17 = Secure e-commerce transaction registration required with other method.
- 21 = Secure e-commerce transaction registration required with valid image select.
- 22 = Nonsecure e-commerce transaction with invalid image select and permanent lock.
- 23 = Nonsecure e-commerce transaction browser close and one-day lock.
- 24 = Nonsecure e-commerce transaction browser close and permanent lock.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 15–26 | 02 | CUST-PHN-NUM<br>The phone number of the customer. | PIC X(12) |
| 27–31 | 02 | FRAUD-SCORE<br>The value of the fraud score. | PIC X(5) |
| 32–36 | 02 | IMAGE-CDE<br>The image code value. | PIC X(5) |
| 37–46 | 02 | TAX-PAN<br>The income tax PAN. | PIC X(10) |
| 47–88 | 02 | IP-ADDR<br>The IP address of shopper and country code associated with the IP adress. | PIC X(42) |
| 89–93 | 02 | PHRASE-CDE<br>The personal phrase value. | PIC X(5) |
| 94–102 | 02 | TXN-FEE<br>The transaction fee. | PIC X(9) |
| 103–111 | 02 | ORIG-TXN-FEE<br>The original transaction fee. | PIC X(9) |
| 112–118 | 02 | FILLER<br>Reserved for future use. | PIC X(7) |

# 6: BASE24-atm Tokens

This section describes the BASE24-atm message tokens. Tokens are described in alphanumeric order, according to token ID. The table below identifies the BASE24-atm tokens and their corresponding token IDs. For tokens with ASCII formats, the ASCII formats follow the corresponding binary format.

| ID | Token Name |
|---|---|
| 02 | Statement Print token |
| 03 | BASE24-atm Release 5.0 token |
| 06 | PIN Change token |
| 07 | Self-Service Banking Base token |
| 14 | Self-Service Banking Check token |
| 15 | Self-Service Banking Check Terminal Settlement token |
| 21 | PS2000 ATM token |
| 22 | Additional Hopper token |
| 24 | ATM Flag 1 token |
| A5 | Non-Currency Dispense token |
| A6 | ATM Interchange Compliance token |
| A7 | Multiple Account token |
| A8 | Bag Deposit token (for Diebold MBC) |
| A9 | Money Exchange token (for Diebold MBC) |
| AA | Merchant Banking Center Settlement token (for Diebold MBC) |
| AB | ATM Balances token |

| ID | Token Name |
|---|---|
| AD | Cash Acceptor Terminal Settlement token |
| AE | Bill Payment Payee List token |
| AF | Bill Payment Confirmation token |
| AG | ATM Data 1 token |
| AH | Multiple Account Inquiry token |
| AI | Interim Statement/Passbook Data token |
| AJ | Interim Statement Processing token |
| AK | Passbook Processing token |
| AL | Hold token |
| AM | Preferred Transaction token |
| AO | Diebold BNA Counts token |
| AR | Custom Response Code token |
| AS | Shared BNA Counts token |
| AT | BNA Multiple Currency token |
| AU | Check Bundle token |
| AV | Bulk Check MICR Token |
| AW | Bulk Check Amount Token |
| AX | Bulk Check SSBC Token |
| AY | Bulk Check Disposition Token |

Refer to section 5 for information about the Header token and token header.

# Token 02 Statement Print Token

The fields in the Statement Print token are described below.

**Note:** This token is only carried in the external message. It is not logged to the Transaction Log File (TLF) or extracted. Internally, the information in this token is carried in separate fields of the Standard Internal Message (STM).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–4 | | STMT-PRNT-TKN | |
| 1 | 02 | PRNT-FRMT<br>Indicates to the host whether the Device Handler process supports BASE24-atm Statement Print Data Compression. Valid values are as follows:<br>0 = Columnar format (data compression is not supported).<br>1 = BASE24-atm data compression format (data compression is supported).<br>When transactions originate at terminals directly connected to the BASE24-atm system, this value is originally obtained from the FRMT field on ATD Diebold 10XX/478X screen 10 or ATD NCR 5XXX screen 7. | PIC X(1) |
| 2–4 | 02 | PRNT-SIZ<br>The maximum number of characters per line supported by the device.<br>When transactions originate at terminals directly connected to the BASE24-atm system, this value is obtained by subtracting the value in the LEFT COLUMN field from the value in the RIGHT COLUMN field on ATD NCR 5XXX screen 7.<br>**Note**: This field is not used by the Diebold 10XX/478X Device Handler process. | PIC 9(3) |

# Token 03 BASE24-atm Release 5.0 Token

The fields in the BASE24-atm Release 5.0 token are described below.

**Note**: When this token is added to a token buffer, the process adding the token initializes all fields that are not pertinent to current processing to 0.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–6 | | AT50-TKN | |
| 1 | 02 | ERR-FLG | PIC X(1) |

A code used to provide additional information regarding the disposition of the transaction. Valid values are as follows:

C = Card verification failed.
F = Fraud possibility. Terminal did not send a PIN in the transaction and the terminal is not configured to support local PIN verification.
I = Invalid MAC.
K = KMAC synchronization error.
L = Invalid PIN length.
M = MAC failure.
N = No response received (timeout).
O = One pin try remaining.
P = PIN change error.
R = Sanity check error—previous zone.
S = Sanity check error.
T = Token error.
0 = No information available.

| | | | |
|---|---|---|---|
| 2 | 02 | LOCL-PIN-VRFY-FLG | PIC X(1) |

Indicates whether local PIN verification is allowed. Valid values are as follows:

0 = Allow local PIN verification.
1 = Do not allow local PIN verification.
2 = Allow local PIN verification for on-us transactions only.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3 | 02 | CRD-VRFY-FLG | PIC X(1) |

Indicates the results of verifying the card. Valid values are as follows:

- C = Card verification was performed and the card verification digits (CVD) were invalid. The situation was noted, and transaction processing continued. This value is used when the appropriate byte of the CV-BAD-DISP field in the Base segment of the CPF contains the value 0.
- D = Card verification was performed and the CVD was invalid. The transaction was denied and the ERR-FLG field was set to C. This value is used when the appropriate byte of the CV-BAD-DISP field in the Base segment of the CPF contains the value 1, 2, or 3.
- J = Card verification checking was not performed. The track length was in error. The BAD TRACK LEN flag in the CPF indicates that the transaction should be denied.
- L = Card verification checking was not performed. The track length was in error. The BAD TRACK LEN flag in the CPF indicates that processing should continue.
- N, 0, or ♭ = Authorizing entity has not attempted card verification or could not verify the CVD due to a security device error. (♭ indicates a blank character.)
- O = Card verification was not performed, CVD was not on the card. Not all cards have a CVD value encoded. The card expiration date must be equal to or greater than an expiration date defined on the CPF to insure that the CVD field has been encoded. If the card expiration date is equal to or greater than the CPF date, the CVD checks are performed.
- Y = Card verification was performed and the CVD was valid.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 4 | 02 | COMPLETE-TRACK2-DATA | PIC X(1) |

Indicates whether the transaction acquirer can capture and transmit complete track data for card swipe transactions. Valid values are as follows:

Y or 0 = Yes, the transaction acquirer captures and transmits complete track data.
N = No, the transaction acquirer cannot capture and transmit complete track data.

**Note:** For transactions that contain manually entered Track 2 data, this field must be set to N by the acquirer.

| | | | |
|---|---|---|---|
| 5 | 02 | UAF-FLG | PIC X(1) |

A flag indicating whether the cardholder had a Usage Accumulation File (UAF) record when the transaction request was processed by the Authorization process. This field is used to limit the number of accesses required to update the UAF. This field is only used with the Negative Authorization with Usage Accumulation method under authorization level 3 (online/offline). Valid values are as follows:

0 = UAF record did not exist
1 = UAF record did exist

| | | | |
|---|---|---|---|
| 6 | 02 | RCPT-RQSTD | PIC X(1) |

A code indicating whether the customer requested a receipt. Valid values are as follows:

0 = Customer was not asked whether a receipt was required
1 = Receipt requested
2 = Receipt not available—printer faulted
3 = No receipt requested

# Token 06 PIN Change Token

The fields in the PIN Change token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–52 | | PINC-TKN | |
| 1 | 02 | NEW-PIN-FRMT<br>The format of the new PIN field. Valid values are as follows:<br>0 = No encryption, clear PIN<br>1 = Encrypted ANSI PIN block<br>3 = Encrypted PIN/PAD PIN block | PIC X(1) |
| 2–17 | 02 | NEW-PIN-OFST<br>The PIN offset for the new PIN. | PIC X(16) |
| 18 | 02 | PIN-CNT<br>The number of new PINs present. Valid values are as follows:<br>1 = One new PIN present<br>2 = Two new PINs present | PIC X(1) |
| 19–20 | 02 | NEW-PIN-SIZE<br>The length of the new PIN (for example, 04 signifies 4 digits). If the new PIN is encrypted, this field contains the value 16. Valid values are in the range from 4 to 12 and the value 16. | PIC 9(2) |
| 21–36 | 02 | NEW-PIN-1<br>The new PIN. | PIC X(16) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 37–52 | 02 | NEW-PIN-2<br>The new PIN (second entry). This PIN is compared to the value in the NEW-PIN-1 field to ensure that the user has entered the same new PIN twice (that is, that the user did not make an error in entering the new PIN). | PIC X(16) |

# Token 07 Self-Service Banking Base Token—Binary Format

The fields in the binary format Self-Service Banking Base token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–10 | | SSBB-TKN | |
| 1 | 02 | SUPER-TLR-IND<br>Indicates whether this is a Super Teller transaction. Valid values are as follows:<br>0 = No, this is not a Super Teller transaction.<br>1 = Yes, this is a Super Teller transaction. | PIC X(1) |
| 2 | 02 | USER-FLD1 | PIC X(1) |
| 3–10 | 02 | DEP-BAL-CR<br>The deposit credit amount for the second deposit account associated with a split deposit transaction. | BINARY 64 SIGNED |

# Token 07 Self-Service Banking Base Token—ASCII Format

The fields in the ASCII format Self-Service Banking Base token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–14 | | SSBB-TKNX | |
| 1 | 02 | SUPER-TLR-IND | PIC X(1) |
| 2 | 02 | USER-FLD1 | PIC X(1) |
| 3–14 | 02 | DEP-BAL-CR | PIC X(12) |

# Token 14 Self-Service Banking Check Token

The fields in the Self-Service Banking Check token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–18 | | SSBC-TKN | |
| 1–5 | 02 | BIN-SEQ-NUM<br>The terminal depository bin and sequence number. The first byte of this field contains the depository bin position. The remainder of this field contains a sequence number ranging from 0001 to 9999. | PIC X(5) |
| 6 | 02 | CHK-RET<br>Indicates whether the check should be retained by the terminal or returned to the cardholder for denied transactions. Valid values are as follows:<br>0 = Return the check<br>1 = Retain the check | PIC X(1) |
| 7 | 02 | CHK-DISP<br>The disposition of the check for cash check transactions in which the acquiring terminal is unable to dispense the full amount requested. Valid values are as follows:<br>0 = Return the check, cancel the transaction, and generate a full reversal. If any change has been dispensed, the Authorization process generates a partial reversal for any undispensed cash amount.<br>1 = Retain the check, complete the transaction, and generate a partial reversal for the undispensed cash amount. | PIC X(1) |
| 8–17 | 02 | CORP-NUM<br>Indicates the number of the corporation that issued the check. The default corporation number is 9999999999. | PIC X(10) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 18 | 02 | CSF-TRAN-IND<br>Indicates whether the current transaction involves the Check Status File (CSF). Valid values are as follows:<br>Y = Yes, this transaction involves the CSF.<br>N or 0 = No, this transaction does not involve the CSF. | PIC X(1) |

# Token 15 Self-Service Banking Check Terminal Settlement Token—Binary Format

The fields in the binary format Self-Service Banking Check Terminal Settlement token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–40 | | SSBC-TERM-SETL-TKN | |
| 1–40 | 02 | DEP-BIN<br>The following fields contain check-related transaction counts and amounts, by depository bin.<br>The length of each DEP-BIN occurrence is 10 positions. | OCCURS 4 TIMES |
| | 04 | NUM-CHK | TYPE BINARY 16 |
| | 04 | AMT-CHK | TYPE BINARY 64 |

# Token 15 Self-Service Banking Check Terminal Settlement Token—ASCII Format

The fields in the ASCII format Self-Service Banking Check Terminal Settlement token format are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–80 | | SSBC-TERM-SETL-TKNX | |
| 1–80 | 02 | DEP-BIN | OCCURS 4 TIMES |
| | 04 | NUM-CHK | PIC 9(5) |
| | 04 | AMT-CHK | PIC 9(15) |

# Token 21 PS2000 ATM Token

The fields in the PS2000 ATM token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | PS2000-ATM-TKN | |
| 1 | 02 | SRV-IND<br>A code that provides additional information about the disposition of the transaction. Valid values are as follows:<br>Y = Transaction submitted for PS2000/ATM<br>N = Transaction not qualified for PS2000/ATM (VIP)<br>E = Transaction meets present card requirements and ATM owner and location data are present<br>X = Disqualified from PS2000 (BASE II)<br>T = No Chargeback Protection Service (CPS) program available<br>0 or ♭ = CPS qualifications info not received (where ♭ is a blank space) | PIC X |
| 2–16 | 02 | TRAN-ID<br>A Visa-generated Transaction Identifier (TID) that is unique for each original authorization and financial request. The identifier links original messages to subsequent messages, such as those for exception item processing and clearing records. The TID is a key element in both CPS and CRS processing. | PIC X(15) |
| 17–20 | 02 | VALID-CDE<br>A code indicating that the transaction has met Payment Service 2000 edits and was approved by the issuer. This field also contains the downgrade reason code if the transaction is rejected for PS2000/ATM. | PIC X(4) |

# Token 22 Additional Hopper Token—Binary Format

The fields in the binary format Additional Hopper token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–92 | | ADDL-HOPR-TKN | |
| 1–92 | 02 | HOPR | OCCURS 2 TIMES |
| | | The characteristics of the additional hopper. | |
| | 04 | CONTENTS | PIC X(2) |
| | | A code identifying the contents of the hopper. Valid values are as follows:<br>00 = Cash<br>01 = Coin | |
| | 04 | BEG-CASH | TYPE BINARY 64 |
| | | The amount of currency in the hopper at the start of the current balancing period. | |
| | 04 | CASH-INCR | TYPE BINARY 64 |
| | | The amount of currency added to the hopper during the current balancing period. | |
| | 04 | CASH-DECR | TYPE BINARY 64 |
| | | The amount of currency removed from the hopper during the current balancing period. | |
| | 04 | CASH-OUT | TYPE BINARY 64 |
| | | The amount of currency dispensed from the hopper through customer withdrawals between terminal balancing periods. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | END-CASH | TYPE BINARY 64 |
| | | The amount of currency remaining in the hopper at the end of the balancing period. | |
| | 04 | CRNCY-CDE | PIC 9(3) |
| | | A code identifying the currency in the hopper. Valid values are listed in the ISO 4217 standard ***Codes for the Representation of Currencies and Funds*.** | |
| | 04 | USER-FLD | PIC X |

# Token 22 Additional Hopper Token—ASCII Format

The fields in the ASCII format Additional Hopper token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–192 | | ADDL-HOPR-TKNX | |
| 1–192 | 02 | HOPR | OCCURS 2 TIMES |
| | 04 | CONTENTS | PICX(2) |
| | 04 | BEG-CASH | PIC9(18) |
| | 04 | CASH-INCR | PIC9(18) |
| | 04 | CASH-DECR | PIC9(18) |
| | 04 | CASH-OUT | PIC9(18) |
| | 04 | END-CASH | PIC9(18) |
| | 04 | CRNCY-CDE | PIC9(3) |
| | 04 | USER-FLD | PICX(1) |

# Token 24 ATM Flag 1 Token

The fields in the ATM Flag 1 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–10 | | AT-FLG1-TKN<br>The ATM Flag 1 token contains miscellaneous fields and flags. | |
| 1–2 | 02 | CRD-TYP<br>The two-character card type code. Card type codes are either reserved by BASE24 products or are user-defined. | PIC X(2) |
| 3 | 02 | AUTH-ONLY<br>This field contains an “A” if the message is an auth-only message. | PIC X(1) |
| 4 | 02 | SVC-IND<br>This field is used to flag a transaction as having been split by Authorization so that Authorization will know when the response comes back that it must be merged with its other side.<br>On reversals, set by Device Handlers to indicate the type of reversal.<br>Used by BASE24 Smart Card Authorization. Values:<br>0 = This is NOT one side of a split transaction.<br>1 = This is one side of a split transaction.<br>9 = This reversal is generated from a late negative completion and must be sent to any SVC issuer who receives completions. | PIC X(1) |
| 5–8 | 02 | EXP-DAT<br>The expiration date from the Track 2 discretionary data. | PIC X(4) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9 | 02 | CASH-DEP-FLG | PIC X |

Indicates whether this transaction is a regular deposit transaction or a cash deposit transaction. Valid values are as follows:

0 = Regular Deposit transaction
1 = Cash Deposit transaction

| | | | |
|---|---|---|---|
| 10 | 02 | CRD-STAT | PIC X(1) |

The card status. Valid values are as follows:

| | | |
|---|---|---|
| ␣ | = | No value set |
| 0 | = | Issued but not active, no transactions allowed |
| 1 | = | Open, transactions allowed |
| 2 | = | Lost, no transactions allowed |
| 3 | = | Stolen, no transactions allowed |
| 4 | = | Restricted, no withdrawals allowed |
| 5 | = | VIP, transactions allowed |
| 6 | = | Check status |
| 9 | = | Closed, no transactions allowed |
| A | = | Referral |
| B | = | Maybe |
| C | = | Denial |
| D | = | Signature restricted |
| E | = | Country club |
| F | = | Expired |
| G | = | Commercial |
| H–Z | = | Reserved, ACI product use |
| a–z | = | Reserved, customer use |

# Token A5 Non-Currency Dispense Token

The fields in the Non-Currency Dispense token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–4 | | NCD-TKN | |
| 1–2 | 02 | ITEM-QTY<br>The number of items being purchased. | PIC XX |
| 3–4 | 02 | HOPR-CONTENTS<br>A code that identifies the item dispensed by the ATM. Valid values are 02–11. The value 02 is reserved for travelers checks; the value 11 is reserved for mobile top-up. | PIC XX |

# Token A6 ATM Interchange Compliance Token

The fields in the ATM Interchange Compliance token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–22 | | ICHG-COMPLIANCE TKN<br>This token is used to carry interchange compliance data for MasterCard. | |
| 1 | 02 | LIFE-CYCLE-IND<br>Reserved for future use. | PIC X |
| 2–16 | 02 | TRACE-ID<br>The code assigned by the interchange to a transaction that has met the required compliance edits. | PIC X(15) |
| 17–20 | 02 | VALID-CDE<br>The code assigned by the interchange to a transaction that has met the required compliance edits and has been approved by the issuer. | PIC X(4) |
| 21 | 02 | MONITORING-STAT<br>A code returned from the interchange indicating whether MasterCard changed the Point of Service Entry Mode from 90 to 02. A value of Y indicates that the status is being monitored. | PIC X |
| 22 | 02 | ERR-IND<br>A code returned from an interchange indicating the error condition that may have occurred. A blank space indicates no error. | PIC X |

# Token A7 Multiple Account Token

The fields in the Multiple Account token are described below.

**Note:** This token is used between the BASE24-atm Device Handler and Authorization processes only. The Device Handler process deletes this token from the end of the STM, and the token is not logged to the TLF.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–12 | | MULT-ACCT-TKN | |
| 1 | 02 | FROM-TO | PIC X(1) |
| | | A code indicating the side of a two-sided transaction that is being processed. Valid values are as follows:<br>0 = The transaction is not a two-sided transaction.<br>1 = The *from* side of a two-sided transaction is being processed.<br>2 = The *to* side of a two-sided transaction is being processed. | |
| 2 | 02 | FIRST-PASS | PIC X(1) |
| | | A code indicating whether the message is the first in a multiple account selection response. Valid values are as follows:<br>0 = The message is not the first in a multiple account selection response.<br>1 = The message is the first in a multiple account selection response. | |
| 3 | 02 | MORE | PIC X |
| | | A code that indicates the side of the transaction being processed and whether there are more accounts for that side of the transaction. Valid values are as follows:<br>0 = No more one-sided accounts<br>1 = More one-sided accounts<br>2 = More *from* accounts and more *to* accounts | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | 3 = More *from* accounts and no more *to* accounts<br>4 = No more *from* accounts and more *to* accounts<br>5 = No more *from* accounts and no more *to* accounts | |
| 4 | 02 | USER-FLD1 | PIC X |
| 5–6 | 02 | FROM-CAF-IDX | TYPE BINARY 16 |
| | | The index number of the next *from* account of the specified type to be presented to the customer. | |
| 7–8 | 02 | FROM-CAF-PRI-IDX | TYPE BINARY 16 |
| | | The index number of the first valid *from* account number of the specified type. This value prevents an account from being presented to the customer more than once. | |
| 9–10 | 02 | TO-CAF-IDX | TYPE BINARY 16 |
| | | The index number of the next *to* account of the specified type to be presented to the customer. | |
| 11–12 | 02 | TO-CAF-PRI-IDX | TYPE BINARY 16 |
| | | The index number of the first valid *to* account number of the specified type. This value prevents an account from being presented to the customer more than once. | |

# Token A8 Bag Deposit Token

The fields in the Bag Deposit token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–12 | | MBC-BD-TKN | |
| 1–12 | 02 | BAG-ID<br>The Bag ID for the Bag Deposit transaction. This value distinguishes between a regular deposit and the MBC Bag Deposit transactions. | PIC X(12) |

# Token A9 Money Exchange Token—Binary Format

The fields in the binary format of the Money Exchange token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–22 | | MBC-MX-TKN | |
| | 02 | FROM-CASH<br>The following fields will be populated for a money exchange from cash transaction. | |
| 1–8 | 04 | CASH-ACCEPTED<br>The amount of cash accepted by the currency acceptor for the money exchange from cash transaction. | TYPE BINARY 64 |
| 9–16 | 04 | CASH-RET<br>The amount of cash returned to the customer for this money exchange from cash transaction, if the amount of cash inserted was more than the change requested. | TYPE BINARY 64 |
| 17 | 04 | RVSL-FLG<br>Indicates a money exchange from cash transaction will be reversed.<br>0 = No reversal.<br>1 = Dispensing error; reversal.<br>2 = Unable to process. Take inserted money; reversal.<br>3 = Exceeded bill or roll limit. Take inserted money; reversal.<br>4 = Unable to give all change requested. No reversal. | PIC 9 |
| 18 | 02 | USER-FLD<br>This field is not currently used. | PIC X |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | CHANGE-REQ<br>The following fields indicate by denomination the change requested by the customer. | OCCURS 8 TIMES |
| 19–20 | 04 | BILLS<br>The number of bills requested. | TYPE BINARY 16 |
| 21–22 | 04 | ROLLS<br>The number of rolls of coins requested. | TYPE BINARY 16 |

# Token A9 Money Exchange Token—ASCII Format

The fields in the ASCII format of the Money Exchange token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–42 | | MBC-MX-TKNX | |
| | 02 | FROM-CASH | |
| 1–18 | 04 | CASH-ACCEPTED | PIC X(18) |
| 19–36 | 04 | CASH-RET | PIC X(18) |
| 37 | 04 | REV-FLAG | PIC 9(1) |
| 38 | 02 | USER-FLD | PIC X(1) |
| | 02 | CHANGE-REQ | OCCURS 8 TIMES |
| 39–40 | 04 | BILLS | PIC X(2) |
| 41–42 | 04 | ROLLS | PIC X(2) |

# Token AA Merchant Banking Center Settlement Token—Binary Format

The fields in the binary format of the Merchant Banking Center Settlement token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–68 | | MBC-SETL-TKN | |
| | 02 | HOPR | OCCURS 6 TIMES |
| 1–2 | 04 | CONTENTS<br>A code identifying the contents of the hopper. Valid values are as follows:<br>11 = Rolled coin | PIC X(2) |
| 3–10 | 04 | BEG-CASH<br>The amount of currency in the RCD hopper at the start of the current balancing period. | TYPE BINARY 64 |
| 11–18 | 04 | CASH-INCR<br>The amount of currency added to the RCD hopper during the current balancing period. | TYPE BINARY 64 |
| 19–26 | 04 | CASH-DECR<br>The amount of currency removed from the RCD hopper during the current balancing period. | TYPE BINARY 64 |
| 27–34 | 04 | CASH-OUT<br>The amount of currency dispensed from the RCD hopper through customer withdrawals between terminal balancing periods. | TYPE BINARY 64 |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 35–42 | 04 | END-CASH<br>The amount of currency remaining in the RCD hopper at the end of the balancing period. | TYPE BINARY 64 |
| 43–45 | 04 | CRNCY-CDE<br>A code identifying the currency in the RCD hoppers. Valid values are listed in the ISO 4217 standard ***Codes for the Representation of Currencies and Funds.*** | PIC 9(3) |
| 46 | 04 | USER-FLD1<br>This field is not currently used. | PIC X |
| | 02 | CRNCY-ACCEPTOR | |
| 47–49 | 04 | CRNCY-CDE<br>A code identifying the currency accepted by the currency acceptor. Valid values are listed in the ISO 4217 standard ***Codes for the Representation of Currencies and Funds.*** | PIC 9(3) |
| 50 | 04 | USER-FLD2<br>This field is not currently used. | PIC X |
| 51–58 | 04 | CASH-IN-VAULT<br>The amount of currency vaulted by currency acceptor completed during the current balancing period. | TYPE BINARY 64 |
| 59–60 | 02 | NUM-MX<br>The number of money exchange transactions completed during the current balancing period. | TYPE BINARY 16 |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 61–68 | 02 | AMT-MX<br>The unverified amount of money exchange transactions completed during the current balancing period. | TYPE BINARY 64 |

# Token AA Merchant Banking Center Settlement—ASCII Format

The fields in the ASCII format Merchant Banking Center Settlement token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–138 | | MBC-SETL-TKNX | |
| | 02 | HOPR | OCCURS 6 TIMES |
| 1–2 | 04 | CONTENTS | PIC XX |
| 3–20 | 04 | BEG-CASH | PIC X(18) |
| 21–38 | 04 | CASH-INCR | PIC X(18) |
| 39–56 | 04 | CASH-DECR | PIC X(18) |
| 57–74 | 04 | CASH-OUT | PIC X(18) |
| 75–92 | 04 | END-CASH | PIC X(18) |
| 93–95 | 04 | CRNCY-CDE | PIC 9(3) |
| 96 | 04 | USER-FLD1 | PIC X(1) |
| | 02 | CRNCY-ACCEPTOR | |
| 97–99 | 04 | CRNCY-CDE | PIC 9(3) |
| 100 | 04 | USER-FLD2 | PIC X(1) |
| 101–118 | 04 | CASH-IN-VAULT | PIC X(18) |
| 119–123 | 02 | NUM-MX | PIC 9(5) |
| 124–138 | 02 | AMT-MX | PIC 9(15) |

# Token AB ATM Balances Token—Binary Format

The ATM Balances token is used when account balances are sent to the ATM in a different currency (i.e., the account currency) than the currency used by the ATM (i.e., the transaction currency).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–60 | 00 | ATM-BAL-TKN | |
| 1 | 02 | CUST-BAL-INFO | PIC X(1) |

A code indicating the type of customer balance information given at an ATM. This field is set in the same way as the RQST.CUST-BAL-INFO field in the BASE24-atm Standard Internal Message (STM). If the ATM Balances token is present, the value in this field takes precedence over the value in the STM field. Valid values are as follows:

0 = No information given.
1 = RQST.AMT-2 field given only.
2 = RQST.AMT-3 field given only.
3 = RQST.AMT-2 and RQST.AMT-3 fields given; AMT-2 preferred.
4 = RQST.AMT-2 and RQST.AMT-3 fields given; AMT-3 preferred.

| | | | |
|---|---|---|---|
| 2 | 02 | CRNCY-BAL-INFO | PIC X(1) |

A code indicating how the balances in the AMT-2 and AMT-3 fields are displayed and printed. The value is retrieved from the appropriate Institution Definition File (IDF) record or can be set by a host or an interchange. Valid values are as follows:

1 = Display and print balances in transaction currency only
2 = Display and print balances in account currency only
3 = Display and print balances in both currencies—transaction currency preferred
4 = Display and print balances in both currencies—account currency preferred

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | Note that a host system can set this field to any of these values, even if it does not provide the balances in the transaction currency, as BASE24 ensures that a balance returned in the account currency is always converted to the transaction currency. | |
| 3–10 | 02 | ACCT-AMT-2 | TYPE BINARY 64 SIGNED |
| | | The balance, corresponding to the RQST.AMT-2 field in the STM, specified in the account currency. | |
| 11–18 | 02 | ACCT-AMT-3 | TYPE BINARY 64 SIGNED |
| | | The balance, corresponding to the RQST.AMT-3 field in the STM, specified in the account currency. | |
| 19–26 | 02 | TXN-AMT-2 | TYPE BINARY 64 SIGNED |
| | | The balance, corresponding to the RQST.AMT-2 field in the STM, specified in the transaction currency. | |
| 27–34 | 02 | TXN-AMT-3 | TYPE BINARY 64 SIGNED |
| | | The balance, corresponding to the RQST.AMT-3 field in the STM, specified in the transaction currency. | |
| 35–37 | 02 | ACCT-CRNCY-CDE | PIC 9(3) |
| | | The ISO numeric currency code of the account currency. If the token does not contain the account currency representation of the balance, this field is set to the value 000. | |
| 38–40 | 02 | TXN-CRNCY-CDE | PIC 9(3) |
| | | The ISO numeric currency code of the transaction currency. If the token does not contain the transaction currency representation of the balance, this field is set to the value 000. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 41–48 | 02 | CONV-RATE<br>The rate used to convert the account balances to the Transaction currency.<br>This field is in ISO format (i.e., the leftmost digit specifies the number of decimal places, and positions 2–8 specify the rate). For example, 69972522 equals 9.972522.<br>If the token contains only the transaction currency representation of the balance or only the account currency representation of the balance, this field is set to the value 00000000. | PIC 9(8) |
| 49–52 | 02 | CONV-DAT<br>The date on which the currency conversion was performed, in MMDD format. If the token contains only the transaction currency representation of the balance or only the account currency representation of the balance, this field is set to the value 0000. | PIC 9(4) |
| 53–60 | 02 | USER-FLD1 | PIC X(8) |

# Token AB ATM Balances Token—ASCII Format

The fields in the ASCII format ATM Balances token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–104 | | ATM-BAL-TKNX | |
| 1 | 02 | CUST-BAL-INFO | PIC X(1) |
| 2 | 02 | CRNCY-BAL-INFO | PIC X(1) |
| 3–21 | 02 | ACCT-AMT-2 | PIC X(19) |
| 22–40 | 02 | ACCT-AMT-3 | PIC X(19) |
| 41–59 | 02 | TXN-AMT-2 | PIC X(19) |
| 60–78 | 02 | TXN-AMT-3 | PIC X(19) |
| 79–81 | 02 | ACCT-CRNCY-CDE | PIC 9(3) |
| 82–84 | 02 | TXN-CRNCY-CDE | PIC 9(3) |
| 85–92 | 02 | CONV-RATE | PIC 9(8) |
| 93–96 | 02 | CONV-DAT | PIC 9(4) |
| 97–104 | 02 | USER-FLD1 | PIC X(8) |

# Token AD Cash Acceptor Terminal Settlement Token

The fields in the Cash Acceptor Terminal Settlement token are described below. This token carries the counts for the Bunch Note Acceptor (BNA) cash counters in a settlement record.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | CASH-ACCPT-TERM-SETL-TKN | |
| 1–5 | 02 | NOTES-REFUNDED<br>The total number of bills refunded. | PIC 9(5) |
| 6–10 | 02 | NOTES-REJECTED<br>The total number of bills rejected. | PIC 9(5) |
| 11–15 | 02 | NOTES-ENCASHED<br>The total number of bills encashed. | PIC 9(5) |
| 16–20 | 02 | NOTES-ESCROWED<br>The total number of bills escrowed. | PIC 9(5) |

# Token AE Bill Payment Payee List Token—Binary Format

The Bill Payment Payee List token is used to store and pass a list of payees for the cardholder when the cardholder performs a bill payment transaction. Currently there are no BASE24 processes that utilize this token; however, it may be added by the host system. This token is supported by the BASE24-eps IFX Device Handler process. The fields in the Bill Payment Payee List token are described below.

**Note:** The Bill Payment Payee List Token is variable length. The values given are the maximum length.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–704 | | BILL-PYMT-PAYEE-LIST-TKN | |
| 1 | 02 | MORE-PAYEES<br>A flag indicating if more payees exist. Valid values are as follows:<br>Y = Yes, more payees exist<br>N = No, more payees do not exist | PIC X(1) |
| 2 | 92 | USER-FLD1 | PIC X(1) |
| 3–4 | 02 | NUM-PAYEES-RETURNED<br>The number of payees present in the company list. Valid values are 01 through 10. | PIC X(2) |
| 5–704 | 02 | COMPANY-LIST | OCCURS 0 TO 10 TIMES DEPENDING ON NUM-PAYEES-RETURNED |
| | 04 | SHORT-COMPANY-NAM<br>The abbreviated name of a company on the cardholder's list of payees. | PIC X(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | ACCT-NUM<br>The cardholder's account number at the company. | PIC X(40) |
| | 04 | MERCH-ID<br>The Host internal merchant number identification for this company. | PIC X(9) |
| | 04 | ACCT-STAT<br>The status of the account with the payee. | PIC X(8) |
| | 04 | EXTRA-IND<br>A flag indicating if extra information is required to use this payee. Valid values are as follows:<br>Y = Yes, more information is required<br>N = No, more information is not required | PIC X(1) |

# Token AE Bill Payment Payee List Token—ASCII Format

The fields in the Bill Payment Payee List token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–704 | | BILL-PYMT-PAYEE-LIST-TKNX | |
| 1 | 02 | MORE-PAYEES | PIC X(1) |
| 2 | 92 | USER-FLD1 | PIC X(1) |
| 3–4 | 02 | NUM-PAYEES-RETURNED | PIC X(2) |
| 5–704 | 02 | COMPANY-LIST | OCCURS 0 TO 10 TIMES DEPENDING ON NUM -PAYEES-RETURNED |
| | 04 | SHORT-COMPANY-NAM | PIC X(12) |
| | 04 | ACCT-NUM | PIC X(40) |
| | 04 | MERCH-ID | PIC X(9) |
| | 04 | ACCT-STAT | PIC X(8) |
| | 04 | EXTRA-IND | PIC X(1) |

# Token AF Bill Payment Confirmation Token

The Bill Payment Confirmation token is used to store and pass the payment date, confirmation number, the payee, and the account from which the bill was paid when the cardholder has performed a bill payment transaction. Currently there are no BASE24 processes that utilize this token; however, it may be added by the host system. The fields in the Bill Payment Confirmation token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–66 | | BILL-PYMT-CONF-TKN | |
| 1–12 | 02 | SHORT-COMPANY-NAME<br>The abbreviated name of a company on the cardholder's list of payees. | PIC X(12) |
| 13–52 | 02 | ACCT-NUM<br>The cardholder's account number at the company selected by the cardholder. | PIC X(40) |
| 53–58 | 02 | CONF-NUM<br>The confirmation number of the payment. | PIC X(6) |
| 59–66 | 02 | PYMT-PRO-DAT<br>The date, in YYYYMMDD format, that the payment will be processed. | PIC X(8) |

# Token AG ATM Data 1 Token

The fields in the ATM Data 1 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | ATM-DATA1-TKN | |
| 1 | 02 | TRK-USED | PIC X |
| | | The track used in processing the transaction. This field is set by the Authorization process. Valid values are as follows:<br>␣ = No information available (␣ indicates a blank space)<br>0 = No track used; manually entered or EMV PAN<br>1 = Track 1<br>2 = Track 2<br>3 = Track 3 | |
| 2 | 02 | TERM-LOC-IND | PIC X |
| | | The location of the terminal. The terminal location indicator field is set by the Device Handler process. Valid values are as follows:<br>␣ = No information available (␣ indicates a blank space)<br>0 = On premises of card acceptor facility (branch location)<br>1 = Off premises of card acceptor facility (remote location)<br>2 = On premises of card acceptor facility (counter location) | |
| 3–5 | 02 | LANG-CDE | PIC X(3) |
| | | A code used to provide additional information regarding the language selected by the cardholder at the ATM. Valid values are the three character code described in ISO 639-2. | |
| 6–20 | 02 | USER-FLD1 | PIC X(18) |

# Token AH Multiple Account Inquiry Token—Binary Format

The Multiple Account Inquiry token is used to store and pass multiple account data. Currently there are no BASE24 processes that utilize this token; however, it may be added by the host system. This token is supported by the BASE24-eps IFX Device Handler process. The fields in the Multiple Account Inquiry token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–154 | | MULT-ACCT-INQ-TKN | |
| 1–2 | 02 | NUM-ACCTS<br>The number of accounts present in the ACCT structure. | TYPE BINARY 16 SIGNED |
| 3–154 | 02 | ACCT | OCCURS 0 TO 4 TIMES BASED ON NUM-ACCTS |
| 3–4 | 04 | TYP<br>The account type. | PIC X(2) |
| 5–23 | 04 | NUM<br>The account number. | PIC X(19) |
| 24 | 04 | STAT<br>The account status. | PIC X(1) |
| 25–32 | 04 | LEDGER-BAL<br>The ledger balance of the account. | TYPE BINARY 64 SIGNED |
| 33–40 | 04 | AVAIL-BAL<br>The available balance of the account. | TYPE BINARY 64 SIGNED |

# Token AH Multiple Account Inquiry Token—ASCII Format

The fields in the Multiple Account Inquiry token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–242 | | MULT-ACCT-INQ-TKNX | |
| 1–2 | 02 | NUM-ACCTS | PIC X(2) |
| 3–240 | 02 | ACCT | OCCURS 0 TO 4 TIMES BASED ON NUM-ACCTS |
| 3–4 | 04 | TYP | PIC X(2) |
| 5–23 | 04 | NUM | PIC X(19) |
| 24 | 04 | STAT | PIC X(1) |
| 25–43 | 04 | LEDGER-BAL | PIC X(19) |
| 44–62 | 04 | AVAIL-BAL | PIC X(19) |

# Token AI Interim Statement/Passbook Data Token

The Interim Statement/Passbook Data token is used either to carry the interim statement information for statement print transactions or the passbook data information for a passbook update transaction in a single message. The Interim Statement/Passbook Data token can be added by the host system. The BASE24-eps IFX Device Handler supports this token. The fields in the Interim Statement/Passbook Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–958 | | INTSTMT-PSBK-DATA-TKN<br>Contains the data that is printed in the passbook or on the interim statement. | |
| 1–958 | | INTERIM-STMT-DATA<br>The data to be printed on the interim statement. The data contained in this token for the interim statement data is in a tag plus value format. Each field in the data is formatted as follows:<br><TT>DDDDDDDDDD<br>TT = The tag ID<br>D = The variable length data<br>Valid tags are as follows:<br><sq> = sequence number — 4 bytes<br><bf> = balance forward — 15 bytes<br>- amount — 13 bytes<br>- sign — 1 byte<br>- balance type — 1 byte<br><ti> = transaction information — 41 bytes<br>- date — 8 bytes<br>- description — 20 bytes<br>- amount — 11 bytes<br>- sign — 1 byte<br>- GCS indicator — 1 byte | PIC X(958) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | <be> = balance ending | 14 bytes |
| | | - amount | 13 bytes |
| | | - sign | 1 byte |

If a consolidated debit and credit is required, the information is placed in a <ti> tag. The DATE is set to zeroes and the DESCRIPTION is either CONSOLIDATED CR or CONSOLIDATED DR.

The GCS Indicator is currently sent in the token, but may not be used.

A 15 line statement is made up of the following tags:
<sq>
<bf>
<ti> * 15
<be>

The maximum length for a 20 line statement is 945 bytes.

| | | | |
|---|---|---|---|
| 1-958 | 02 | PSBK-DATA | REDEFINES INTERIM STMT-DATA PIC X(958) |

The data to be printed in the passbook. The data contained in this token for the passbook data is in a tag plus length plus value format. Each field in the data is formatted as follows:

<TT>LLDDDDDDDDDD

TT = The tag ID
LL = The length of the field (always two bytes)
D = The variable length data

Leading zeroes should be omitted from amounts. Valid tags are as follows:

| | | | |
|---|---|---|---|
| <id> | = | item identifier | 3 characters |
| <dt> | = | date (format MMM DD-YY) | 9 characters |
| <mn> | = | mnemonic code | 3 characters |
| <ba> | = | balance amount | 14 characters |
| | | - amount | 13 bytes |
| | | - sign | 1 byte |
| <am> | = | transaction amount | 12 characters |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | - amount | 11 bytes |
| | | - sign | 1 byte |
| | | <de> = description | 20 characters |

Each line to be printed in the passbook requires all six tags. The item identifier is not printed in the passbook, but is used to identify the last line printed in confirmation messages to the host.

Following is an example of a passbook line:
<id>03001<dt>09SEP 01-04<mn>03WDL<am>052000+
<ba>07123456-<de>15Txn description

# Token AJ Interim Statement Processing Token—Binary Format

The nterim Statement Processing token contains statement data when a passbook update transaction is performed. This token is added by the BASE24-eps IFX Device Handler process and may be updated at the host system. The fields in the Interim Statement Processing token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–8 | | INTERIM-STMT-PROC-TKN | |
| 1–2 | 02 | NUM-LINES-RQST<br>The number of items selected by the cardholder to be printed. | TYPE BINARY 16 SIGNED |
| 3–4 | 02 | NUM-LINES-RETURNED<br>The number of lines returned by the host. | TYPE BINARY 16 SIGNED |
| 5–6 | 02 | NUM-LINES-MATCHED<br>The number of lines matched. This is the total number of lines to be returned for this statement transaction. If the number of lines is unknown, a value of -1 is used. | TYPE BINARY 16 SIGNED |
| 7 | 02 | MORE-DATA-IND<br>A flag indicating if further statement data exists. Valid values are as follows:<br>Y = Yes, more data exists.<br>N = No, more data does not exist. | PIC X(1) |
| 8 | 02 | USER-FLD1 | PIC X(1) |

# Token AJ Interim Statement Processing Token—ASCII Format

The fields in the Interim Statement Processing token are described below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–8 | | INTERIM-STMT-PROC-TKNX | |
| 1–2 | 02 | NUM-LINES-RQST | PIC 9(2) |
| 3–4 | 02 | NUM-LINES-RETURNED | PIC 9(2) |
| 5–6 | 02 | NUM-LINES-MATCHED | PIC 9(2) |
| 7 | 02 | MORE-DATA-IND | PIC X(1) |
| 8 | 02 | USER-FLD1 | PIC X(1) |

# Token AK Passbook Processing Token—Binary Format

The Passbook Processing token contains the processing data when a passbook update transaction is performed. This token is added by the BASE24-eps IFX Device Handler process and may be updated at the host system. The fields in the Passbook Processing token are described below

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | PSBK-PROC-TKN | |
| 1–8 | 02 | PSBK-BAL | TYPE BNARY 64 SIGNED |
| | | The current balance amount of the last line in the passbook. | |
| 9–13 | 02 | PSBK-NEXT-LINE | PIC X(5) |
| | | The location of the next free line in the passbook. | |
| 14 | 02 | MSG-TYP-IND | PIC X(1) |
| | | A flag indicating the current stage of the passbook transaction. Valid values are as follows:<br>1 = Indicates that this is the initial request/response message<br>2 = Indicates that this is the first data request/response message<br>3 = Indicates that this is a subsequent data request/response message<br>C = Indicates that this is a confirmation message<br>R = Indicates that this is a reversal message | |
| 15 | 02 | PSBK-PRNT-FRMT | PIC X(1) |
| | | This field indicates the type of passbook being used by the customer. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 16–18 | 02 | LAST-ITEM-PRNTD | PIC X(3) |
| | | This field contains the item identifier of the last item printed in the passbook. | |
| 190 | 02 | MORE-DATA-IND | PIC X(1) |
| | | A flag indicating if further passbook data exists. Valid values are as follows:<br>M = More item exist<br>N = No more items exist | |
| 20 | 02 | CNSLDT | PIC X(1) |
| | | This field indicates if the passbook is consolidated. Valid values are as follows:<br>Y = The passbook is consolidated<br>N = The passbook is not consolidated | |

# Token AK Passbook Processing Token—ASCII Format

The fields in the Passbook Processing token are described below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–32 | | PSBK-PROC-TKNX | |
| 1–19 | 02 | PSBK-BAL | PIC X(19) |
| 20–24 | 02 | PSBK-NEXT-LINE | PIC X(5) |
| 25 | 02 | MSG-TYP-IND | PIC X(1) |
| 26 | 02 | PSBK-PRNT-FRMT | PIC X(1) |
| 27–29 | 02 | LAST-ITEM-PRNTD | PIC X(3) |
| 30 | 02 | MORE-DATA-IND | PIC X(1) |
| 31 | 02 | CNSLDT | PIC X(1) |
| 32 | 02 | PSBK-USER-FLD1 | PIC X(1) |

# Token AL Hold Token—Binary Format

The Hold token is used to carry information that will alert cardholders the authorizer has placed a hold on funds deposited via an envelope deposit at an ATM. This token may be added by the host; the BASE24-eps IFX Device Handler supports and updates this token. The fields in the Hold token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–16 | | HLD-TKN | |
| 1–8 | 02 | HLD-AMT<br>The amount of deposit that is on hold. | TYPE BINARY 64 SIGNED |
| 9–14 | 02 | HLD-DAT<br>The date, in YYMMDD format, that the deposit is scheduled to be released from hold. | PIC 9(6) |
| 15 | 02 | HLD-ACCEPTED<br>A flag indicating whether the cardholder has accepted a hold placed on a deposit. Valid values are as follows:<br>Y = The cardholder has accepted the hold<br>N = The cardholder has rejected the hold | PIC X(1) |
| 16 | 02 | HLD-USER-FLD1 | PIC X(1) |

# Token AL Hold Token—ASCII Format

The fields in the Hold token are described below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–26 | | HLD-TKNX | |
| 1–19 | 02 | HLD-AMT | PIC X(19) |
| 20–25 | 02 | HLD-DAT | PIC 9(6) |
| 26 | 02 | HLD-ACCEPTED | PIC X(1) |

# Token AM ATM Preferred Transaction Token—Binary Format

The ATM Preferred Transaction token carries information about a customer's preferences as listed on the customer's record in the Cardholder Authorization File (CAF). The fields in the ATM Preferred Transaction Token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–30 | | ATM-PFRD-TXN-TKN | |
| 1 | 02 | ACCT<br>The account number of the cardholder's preferred transaction. | TYPE * |
| 2–3 | 02 | TRAN-CDE<br>The transaction code for the cardholder's preferred transaction. | PIC X(2) |
| 4–5 | 02 | FROM-ACCT-TYP<br>The "from" account type for the cardholder's preferred transaction. | TYPE ACCT-TYP |
| 6–7 | 02 | TO-ACCT-TYP<br>The "to" account type for the cardholder's preferred transaction. | TYPE ACCT-TYP |
| 8 | 02 | RCPT-OPT<br>The receipt option for the customer's preferred transaction. Valid values are as follows:<br>Y = Yes, a receipt is required<br>N = No, a receipt is not required | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9–12 | 02 | AMT | TYPE BINARY 64 |
| | | The amount of the cardholder's preferred transaction. | |
| 13 | 02 | PRFL-UPDT-IND | PIC X(1) |
| | | Indicates if the preferred transaction profile can be updated by the customer at the terminal. Valid values are as follows:<br>Y = Yes, updates to the customer's profile is allowed at a terminal<br>N = No, updates to the customer's profile is not allowed at a terminal | |
| 14 | 02 | ADA-IND | PIC X(1) |
| | | Indicates if the customer wishes to have American Disabilities Act (ADA) support at the terminal, if available. Valid values are as follows:<br>Y = Yes, ADA support is needed at the terminal<br>N = No, ADA support is not needed at the terminal | |
| 15–16 | 02 | MRKT-SEG-IND | PIC X(2) |
| | | The market segment indicator for this cardholder. | |
| 17 | 02 | ON-US-CRD-IND | PIC X(1) |
| | | Indicates whether this transaction is considered to be an on-us transaction, which means the FIID of the card is the same as the FIID of the terminal used. This field is set on each transaction and is not stored on the Cardholder Authorization File (CAF). Valid values are as follows:<br>Y = Yes, this transaction is from an on-us card<br>N = No, this transaction is not from an on-us card | |
| 18–309 | 02 | USER-FLD-ACI | PIC X(13) |
| | | Reserved by ACI for future use. | |

# Token AM ATM Preferred Transaction Token—ASCII Format

The fields in the ASCII format ATM Preferred Transaction token are shown below. For description of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–46 | | ATM-PFRD-TXN-TKNX | |
| 1 | 02 | ACCT | TYPE * |
| 2–3 | 02 | TRAN-CDE | PIC X(2) |
| 4–5 | 02 | FROM-ACCT-TYP | TYPE ACCT-TYP |
| 6–7 | 02 | TO-ACCT-TYP | TYPE ACCT-TYP |
| 8 | 02 | RCPT-OPT | PIC X(1) |
| 9–27 | 02 | AMT | PIC X(19) |
| 28 | 02 | PRFL-UPDT-IND | PIC X(1) |
| 29 | 02 | ADA-IND | PIC X(1) |
| 30–31 | 02 | MRKT-SEG-IND | PIC X(2) |
| 32 | 02 | ON-US-CRD-IND | PIC X(1) |
| 33–46 | 02 | USER-FLD-ACI | PIC X(14) |

# Token AO Diebold BNA Counts Token—Binary Format

The Diebold BNA Counts token carries the Bulk Note Acceptor (BNA) data obtained from a Diebold ATM equipped with a currency device. These counts can only be obtained from the ATM and must be requested via the Retrieve and Set/Clear Enhanced Supply Counts Operational Command message. The data is then written to the Transaction Log File (TLF) whenever the ATM is balanced.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–222 | | DIEBOLD-BNA-CNTS-TKN | |
| 1–3 | 02 | CRNCY-CDE | PIC X(3) |
| | | A code identifying the currency in the BNA. It is in ISO format. | |
| 4 | 02 | BNA-USER-FLD1 | PIC X |
| | | This field is not used. | |
| 5–6 | 02 | NUM-ENTRIES | TYPE BINARY 16 SIGNED |
| | | The number of configured deposit note type entries in the BNA-INFO substructure. The value for this field is obtained from the ATM. | |
| 7–222 | 02 | BNA-INFO | OCCURS 0 TO 9 TIMES DEPENDING ON NUM-ENTRIES |
| | | The following fields contain the value and counts for each of the deposit note types configured at the ATM. The ninth entry is used as a "catch all" bucket if the number of configured deposit note types is greater than eight. | |
| | 04 | NOTE-VALUE | TYPE BINARY 32 SIGNED |
| | | The monetary value of the note. The value for the ninth entry (the "catch all" bucket) will be 0. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | CB4-CNT | TYPE BINARY 32 SIGNED |
| | | The number of Type 4 (genuine) notes deposited into the Cashbox since the last time the terminal was balanced. | |
| | 04 | CB3-CNT | TYPE BINARY 32 SIGNED |
| | | The number of Type 3 (suspect) notes deposited into the Cashbox since the last time the terminal was balanced. | |
| | 04 | RE4-CNT | TYPE BINARY 32 SIGNED |
| | | The number of Type 4 (genuine) notes deposited into the Retract Area since the last time the terminal was balanced. | |
| | 04 | RE3-CNT | TYPE BINARY 32 SIGNED |
| | | The number of Type 3 (suspect) notes deposited into the Retract Area since the last time the terminal was balanced. | |
| | 04 | A6-CNT | TYPE BINARY 32 SIGNED |
| | | The number of Type 2 (counterfeit) notes deposited into the Counterfeit Area since the last time the terminal was balanced. This field is only applicable when the ATM is configured to support Article 6 (a European Union requirement). It will contain a 0 if Article 6 is not configured. | |

# Token AO Diebold BNA Counts Token—ASCII Format

The fields in the ASCII format Diebold BNA Counts token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–312 | | DIEBOLD-BNA-CNTS-TKNX | |
| 1–3 | 02 | CRNCY-CDE | PIC 9(3) |
| 4 | 02 | BNA-USER-FLD1 | PIC X |
| 5–6 | 02 | NUM-ENTRIES | PIC 9(2) |
| 7–312 | 02 | BNA-INFO | OCCURS 0 TO 9 TIMES DEPENDING ON NUM-ENTRIES |
| | 04 | NOTE-VALUE | PIC 9(9) |
| | 04 | CB4-CNT | PIC 9(5) |
| | 04 | CB3-CNT | PIC 9(5) |
| | 04 | RE4-CNT | PIC 9(5) |
| | 04 | RE3-CNT | PIC 9(5) |
| | 04 | A6-CNT | PIC 9(5) |

# Token AR Custom Response Code Token

The Custom Response Code token can be added and sent by the Host or the BASE24-eps IFX Device Handler process. When this token is sent from the host, it contains the IFX response code. When this token is sent from the BASE24-eps IFX Device Handler process, it contains the IFX reversal reason code.

The Custom Response Code token is sent as 12 bytes, right-justified and zero-padded from the left. If the code is user-defined, the first byte is a (–). The fields in the Custom Response Code token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–12 | | CSTM-RESP-CDE-TKN | |
| 1–12 | 02 | CSTM-RESP-CDE | PIC X(12) |

When added and returned by the host, this token contains the custom (i.e., IFX) response code returned from the host for denied transactions.

When added by the BASE24-eps IFX Device Handler process, this token contains one of the following values:

For declined from-us transactions, the IFX response code to be returned by the ATM.

For reversals generated by the ATM, the IFX reason code to be sent to the host.

# Token AS Shared BNA Counts Token—Binary Format

The Shared BNA Counts token contains the transaction and note counts for the Bunch Note Acceptor (BNA) device. These counts are obtained from the ATM and are requested via the CashIn Supply Counters Request Terminal Command message. The data in this token is written to the Transaction Log File (TLF) whenever the ATM is balanced.

This token is used only by ATMs whose firmware conforms to Wincor Nixdorf's NDC/DDC Message Format Extension for CashIn and are equipped with a BNA device. The BASE24-atm self service banking module is required to support this function.

The fields in the Shared BNA Counts token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–16 | | SHRD-BNA-CNTS-TKN | |
| 1–2 | 02 | BNA-TXN-CNT<br>The number of CashIn transactions (BNA deposit transactions) performed at this ATM. | BINARY 16 |
| 3–4 | 02 | RETRACT-NOTES-CNT<br>The number of note retract operations performed by the ATM since the last replenishment. | BINARY 16 |
| 5–6 | 02 | RETRACT-COINS-CNT<br>The number of coin retract operations performed by the ATM since the last replenishment. | BINARY 16 |
| 7–8 | 02 | L4-NOTES-CNT<br>The number of Level 4 (genuine) notes deposited into the ATM since the last replenishment. This number includes the number of notes retained because of timeouts or jams. | BINARY 16 |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9–10 | 02 | L3-NOTES-CNT | BINARY 16 |
| | | The number of Level 3 (suspect) notes deposited into the ATM since the last replenishment. This number includes the number of notes retained because of timeouts or jams. | |
| 11–12 | 02 | L2-NOTES-CNT | BINARY 16 |
| | | The number of Level 2 (counterfeit) notes deposited into the ATM since the last replenishment. This number includes the number of notes retained because of timeouts or jams. | |
| 13–16 | 02 | USER-FLD-ACI | PIC X(4) |
| | | This field is reserved by ACI for future use. | |

# Token AS Shared BNA Counts Token—ASCII Format

The fields in the ASCII format Shared BNA Counts token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–34 | | SHRD-BNA-CNTS-TKNX | |
| 1–4 | 02 | BNA-TXN-CNT | PIC 9(4) |
| 5–9 | 02 | RETRACT-NOTES-CNT | PIC 9(5) |
| 10–14 | 02 | RETRACT-COINS-CNT | PIC 9(5) |
| 15–19 | 02 | L4-NOTES-CNT | PIC 9(5) |
| 20–24 | 02 | L3-NOTES-CNT | PIC 9(5) |
| 25–29 | 02 | L2-NOTES-CNT | PIC 9(5) |
| 30–34 | 02 | USER-FLD-ACI | PIC X(5) |

# Token AT BNA Multiple Currency Token—Binary Format

The BNA Multiple Currency token contains information regarding the individual currencies that were deposited at the ATM during a single Cash Deposit (Bunch Note Acceptor (BNA)) transaction. This token is added by the Device Handler process on a request for a Cash Deposit if the Device Handler process detects that currency other than the terminal's currency was deposited at the ATM. The token is used to format the receipt during response processing.

This token is used only by ATMs whose firmware conforms to Wincor Nixdorf's NDC/DDC Message Format Extension for CashIn and are equipped with a BNA device. The BASE24-atm self service banking module is required to support this function.

The fields in the BNA Multiple Currency token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–116 | | BNA-MCU-TKN | |
| 1–3 | 02 | TERM-CRNCY-CDE<br>The currency code for the terminal, in ISO format. | TYPE CRNCY-CDE |
| 4 | 02 | USER-FLD1-ACI<br>This field is not used. | PIC X |
| 5–116 | 02 | AMT-INFO<br>Contains the details of each of the currencies that were deposited in this single transaction. Up to four currences are supported. | OCCURS 4 TIMES |
| | 04 | ORIG-CRNCY-CDE<br>The currency code for one of the currencies deposited in this transaction. | TYPE CRNCY-CDE |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | USER-FLD2-ACI | PIC X |
| | | This field is not used. | |
| | 04 | ORIG-AMT | TYPE BINARY 64 |
| | | The amount of the deposit, using the original currency. | |
| | 04 | CONV-RATE | PIC 9(8) |
| | | The rate used to convert the original amount into the currency used by the terminal. This field is in ISO format; i.e., the leftmost digit denotes the number of decimal places and positions 2–8 give the rate. For example, 69972522 = 9.972522. | |
| | 04 | CONV-AMT | TYPE BINARY 64 |
| | | The amount of the currency deposited, converted into the currency used by the terminal. | |

# Token AT BNA Multiple Currency Token—ASCII Format

The fields in the ASCII format BNA Multiple Currency token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–196 | | BNA-MCU-TKNX | |
| 1–3 | 02 | TERM-CRNCY-CDE | TYPE CRNCY-CDE |
| 4 | 02 | USER-FLD1-ACI | PIC X |
| 5–196 | 02 | AMT-INFO | OCCURS 4 TIMES |
| | 04 | ORIG-CRNCY-CDE | TYPE CRNCY-CDE |
| | 04 | USER-FLD2-ACI | PIC X |
| | 04 | ORIG-AMT | PIC X(18) |
| | 04 | CONV-RATE | PIC 9(8) |
| | 04 | CONV-AMT | PIC X(18) |

# Token AU Check Bundle Token—Binary Format

The fields in the Check Bundle token contain the total number of checks deposited and the index of the check currently being processed. The data in this token gets written to the Transaction Log File (TLF) for each check processed.

This token is used only when processing transactions from ATMs equipped with a Check Processing Module (CPM) that supports check bundle deposits. The BASE24-atm self service banking module is required to support this function.

The fields in the Check Bundle token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–14 | | CHK-BNDL-TKN | |
| 1–2 | 02 | NUM-CHKS<br>The total number of checks contained in the bundle deposited. | TYPE BINARY 16 |
| 3–4 | 02 | CHK-IDX<br>The index into the bundle of the check currently being processed. | TYPE BINARY 16 |
| 5–14 | 02 | USER-FLD-ACI<br>Reserved by ACI for future use. | PIX X(10) |

# Token AU Check Bundle Token—ASCII Format

The fields in the ASCII format CheckBundle token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–14 | | CHK-BNDL-TKNX | |
| 1–2 | 02 | NUM-CHKS | PIC X(2) |
| 3–4 | 02 | CHK-IDX | PIC X(2) |
| 5–14 | 02 | USER-FLD1-ACI | PIC X(10) |

# Token AV Bulk Check MICR Token—Binary Format

The Bulk Check MICR Token contains a field indicating the number of checks in a a bulk check transaction. This field also indicates the number of occurrences of MICR data that follow in the token. Up to 20 occurrences of 64 bytes of MICR data are present.

**Note:** This token is nonfunctioning in the BASE24-atm standard product. Its purpose is to support the use of custom-developed Bulk Check device handler functionality.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–1282 | | BULK-CHK-MICR-TKN<br>The fields in the Bulk Check MICR Token are described below. | |
| 1–2 | 02 | NUM-CHKS<br>Indicates the number of checks that are part of the transaction. This field also indicates the number of occurrences of MICR data that will follow in the token. | TYPE BINARY 16 SIGNED |
| 3–1282 | 02 | MICR-DATA<br>Contains the MICR data for all of the checks that are deposited in the ATM as part of this bulk check transaction. This field can store MICR data for up to 20 checks. | OCCURS 20 TIMES |
| | 04 | MICR | PIC X(64) |

# Token AV Bulk Check MICR Token—ASCII Format

The fields in the ASCII format Bulk Check MICR token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–1282 | | BULK-CHK-MICR-TKN | |
| 1–2 | 02 | NUM-CHKS | PIC 9(2) |
| 3–1282 | 02 | MICR-DATA | OCCURS 20 TIMES |
| | 04 | MICR | PIC X(64) |

# Token AW Bulk Check Amount Token—Binary Format

The Bulk Check Amount Token contains a field that indicates the number of checks that are part of this transaction. This field will also indicate the number of occurrences of Amount data that will follow in the token. Up to 20 occurrences of 12 byte check amounts will be present.

**Note:** This token is nonfunctioning in the BASE24-atm standard product. Its purpose is to support the use of custom-developed Bulk Check device handler functionality.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–242 | | BULK-CHK-AMT-TKN<br>The fields in the Bulk Check Amount Token are described below. | |
| 1–2 | 02 | NUM-CHKS<br>Indicates the number of checks that are part of the transaction. This field also indicates the number of occurrences of amount data that will follow in the token. | TYPE BINARY 16 SIGNED |
| 3–242 | 02 | AMT-DATA<br>Contains the amount data for all of the checks that are deposited in the ATM as part of this bulk check transaction. This field can store amount data for up to 20 checks. | OCCURS 20 TIMES |
| | 04 | AMOUNT | PIC X(12) |

# Token AW Bulk Check Amount Token—ASCII Format

The fields in the ASCII format Bulk Check Amount token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–242 | | BULK-CHK-AMT-TKN | |
| 1–2 | 02 | NUM-CHKS | PIC 9(2) |
| 3–242 | 02 | AMT-DATA | OCCURS 20 TIMES |
| | 04 | AMOUNT | PIC X(12) |

# Token AX Bulk Check SSB Token—Binary Format

The Bulk Check SSB Token contains a field that indicates the number of checks that are part of this transaction. This field will also indicate the number of occurrences of the SSBC data that will follow in the token. The SSBC data is the same as that defined in the SSBC token (ID '14') for a single check. Up to 20 occurrences of 18 bytes of SSBC data will be present.

**Note:** This token is nonfunctioning in the BASE24-atm standard product. Its purpose is to support the use of custom-developed Bulk Check device handler functionality.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–362 | | BULK-CHK-SSBC-TKN<br>The fields in the Bulk Check SSBC Token are described below. | |
| 1–2 | 02 | NUM-CHKS<br>Indicates the number of checks that are part of the transaction. This field also indicates the number of occurrences of SSBC data that will follow in the token. | TYPE BINARY 16 SIGNED |
| 3–362 | 02 | SSBC-DATA<br>Contains the SSBC data for all of the checks that are deposited in the ATM as part of this bulk check transaction. This field can store amount data for up to 20 checks. | OCCURS 20 TIMES |
| | 04 | BIN-SEQ-NUM<br>Indicates the terminal depository bin and sequence number. | PIC X(5) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | CHK-RET | PIC X(1) |

Indicates whethter the check should be retained by the terminal or returned to the cardholder for denied transactions. Valid values are as follows:

0 = Return the check.
1 = Retain the check.

| | 04 | CHK-DISP | PIC X(1) |
|---|---|---|---|

The disposition of the check for cash check transaction when the acquiring terminal is unable to dispense the full amount requested. Valid values are as follows:

0 = Cancel the cash check transaction and return the check to the cardholder. The approved transaction will be reversed.

Note: It is possible that coins may have already been dispensed to the customer, so the reversal that is generated may be a partial reversal with a completed amount equivalent to the amount of the change dispensed.

1 = Allow the cash check transaction to complete by retaining the check and dispensing the cash that is available. BASE24-atm will generate a full or partial reversal to reflect that the check was accepted, but the full dispense was not successful.

| | 04 | CORP-NUM | PIC X(10) |
|---|---|---|---|

Indicates the number of the corporation that issued the check. The default corporation number is 9999999999.

| | 04 | CSF-TRAN-IND | PIC X(1) |
|---|---|---|---|

Determines whether or not the current transaction involves the CSF. Valid values are as follows:

N or 0 = Transaction does not involve the CSF.
Y = Transaction involves the CSF.

# Token AX  Bulk Check SSB Token—ASCII Format

The fields in the ASCII format Bulk Check SSB token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–362 | | BULK-CHK-SSBC-TKN | |
| 1–2 | 02 | NUM-CHKS | PIC 9(2) |
| 3–362 | 02 | SSBC-DATA | OCCURS 20 TIMES |
| | 04 | BIN-SEQ-NUM | PIC X(5) |
| | 04 | CHK-RET | PIC X(1) |
| | 04 | CHK-DISP | PIC X(1) |
| | 04 | CORP-NUM | PIC X(10) |
| | 04 | CSF-TRAN-IND | PIC X(1) |

# Token AY Bulk Check Disposition Token—Binary Format

This new token will contain a field that indicates the number of checks that are part of this transaction. This field will also indicate the number of occurrences of the disposition data that will follow in the token. The disposition data is a single byte that will indicate whether the check is to continue on to be deposited (approved) or is to be returned to the customer (denied). Up to 20 occurrences of 1 byte of disposition data will be present.

**Note:** This token is nonfunctioning in the BASE24-atm standard product. Its purpose is to support the use of custom-developed Bulk Check device handler functionality.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–22 | | BULK-CHK-DISP-TKN<br>The fields in the Bulk Check Disposition Token are described below. | |
| 1–2 | 02 | NUM-CHKS<br>Indicates the number of checks that are part of the transaction. This field also indicates the number of occurrences of disposition data that will follow in the token. | TYPE BINARY 16 SIGNED |
| 3-22 | 02 | DISP-DATA<br>Contains the disposition data for all of the checks that are deposited in the ATM as part of the Bulk Check transaction. The disposition data is a single byte that will indicate whether the check will be approved and deposited or whether it will be denied and returned to the customer. The field can store disposition data for up to 20 checks. Valid values are as follows:<br>N = Return the check to the customer (denied).<br>Y = Continue the check on to be deposited (approved). | OCCURS 20 TIMES |
| | 04 | DISP | PIX X(1) |

# Token AY Bulk Check Disposition Token—ASCII Format

The fields in the ASCII format Bulk Check Disposition token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–22 | | BULK-CHK-DISP-TKN | |
| 1–2 | 02 | NUM-CHKS | PIC 9(2) |
| 3-22 | 02 | DISP-DATA | OCCURS 20 TIMES |
| | 04 | DISP | PIX X(1) |

# 7: BASE24-pos Tokens

This section describes the BASE24-pos message tokens. Tokens are described in alphanumeric order, according to token ID. The table below identifies the BASE24-pos tokens and their corresponding token IDs. For tokens with ASCII formats, the ASCII formats follow the corresponding binary format.

| ID | Token Name |
|---|---|
| 01 | Address Verification token |
| 04 | BASE24-pos Release 5.0 token |
| 05 | Check Guarantee/Verification token |
| 10 | American Express token |
| 11 | Automated Clearing House (ACH) Debit token |
| 16 | Alternate Merchant ID token |
| 17 | Visa Payment Service 2000 token |
| 19 | Visa Payment Service 2000 Offline token |
| 20 | Interchange Compliance token |
| 29 | Check Guarantee/Verification 2 token |
| 31 | Check Callback token |
| C0 | BASE24-pos Release 5.1 token |
| C1 | Station ID token |
| C2 | Purchase Card and Fleet Card token |
| C3 | Certificate token |
| C4 | Point of Service Data token |

| ID | Token Name |
|---|---|
| C5 | Increased Optional Data token |
| C6 | Trans Stain XID token |
| C7 | Cardholder Serial Number token |
| C8 | Merchant Serial Number token |
| C9 | MHI Additional Data token |
| CA | DUKPT Data token |
| CB | POS Balances token |
| CE | Authentication Data token |
| CF | International Address Verification Service (IAVS) Data token |
| CH | POS Data1 token |
| CI | POS Merchant token |
| CJ | Pre-Pay Merchant token |
| CK | Industry Data token |
| CP | Healthcare token |
| CQ | Reward Program token |
| CR | POS Split Transaction Routing token |
| CS | Enhanced Reversal Routing token |
| CT | Transaction Specific Data token |
| CU | American Express Additional Data token |
| CV | Healthcare/Transit token |
| CW | Healthcare Service token |
| CX | American Express Private Use Data token |
| CY | Auto-Substantiation Data token |

| ID | Token Name |
|---|---|
| CZ | POS Data 2 token |
| F1 | E-commerce Additional Data Token |
| F2 | Installment Payment Data token |
| F3 | Transit Transaction Token |
| F4 | Digital Wallet Token |
| U0 | EBT Voucher token |
| U1 | EBT Available Balance token |
| U2 | Stored Value token |

Refer to section 5 for information about the Header token and token header.

# Token 01 Address Verification Token

The fields in the Address Verification token are described below.

**Note:** This token is only carried in the external message. It is not logged to the POS Transaction Log File (PTLF) or extracted. Internally, the information in this token is carried in separate fields of the POS Standard Internal Message (PSTM).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–30 | | ADDR-VER-TKN | |
| 1–20 | 02 | ADDR<br>The billing address for the cardholder. | PIC X(20) |
| 21–29 | 02 | ZIP-CDE<br>The ZIP code or postal code for the billing address for the cardholder. The ZIP code must be five or nine digits long. If the ZIP code is five digits long, it is left-justified and blank filled. | PIC X(9) |
| 30 | 02 | USER-FLD1 | PIC X(1) |

# Token 04 BASE24-pos Release 5.0 Token

The fields in the BASE24-pos Release 5.0 token are described below.

**Note**: When this token is added to a token buffer, the process adding the token initializes all fields that are not pertinent to current processing to spaces.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–20 | | PS50-TKN | |
| 1 | 02 | ERR-FLG | PIC X(1) |

A code used to provide additional information regarding the disposition of the transaction. Valid values are as follows:

- A = BASE24-pos Terminal Data files adjustment limit exceeded
- C = Card verification failed
- E = BASE24-pos Terminal Data files return limit exceeded
- I = Invalid MAC
- K = KMAC synchronization error
- L = Invalid PIN length
- M = MAC failure
- P = Invalid PIN block
- R = Sanity check error—previous zone
- S = Sanity check error
- T = Token error
- U = Recurring payment cancellation service
- V = Stop payment order
- W = Revocation of authorization order
- X = Revocation of all authorization orders
- Y = Unmatched Reversal - original authorization too old
- Z = Unmatched Reversal - original authorization should have been matched
- 1 = New account information available for recurring payments transaction
- 2 = Try again later, recurring payments transaction
- 3 = Do not try again for recurring payments transaction
- b = No information available (b indicates a blank space)

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 2–12 | 02 | RTE-GRP | PIC X(11) |

The routing group. The Router module uses the value in this field in the call to the SPROUTE LOOKUP procedure.

| | | | |
|---|---|---|---|
| 13 | 02 | CRD-VRFY-FLG | PIC X(1) |

Indicates the results of verifying the card. Valid values are as follows:

- 0 = Card verification was not performed because the transaction was denied before card verification processing started.
- C = Card verification was performed and the card verification digits (CVD) were invalid. The situation was noted, and transaction processing continued. This value is used when the appropriate byte of the CV-BAD-DISP field in the Base segment of the CPF contains the value 0.
- D = Card verification was performed and the CVD was invalid. The transaction was denied and the ERR-FLG field was set to C. This value is used when the appropriate byte of the CV-BAD-DISP field in the Base segment of the CPF contains the value 1, 2, or 3.
- J = Card verification was not performed. The track length was in error. The BAD TRACK LEN flag in the CPF indicates that the transaction should be denied.
- K = Card verification was not performed. The track length was in error. The BAD TRACK LEN flag in the CPF indicates that the transaction should be referred.
- L = Card verification was not performed. The track length was in error. The BAD TRACK LEN flag in the CPF indicates that processing should continue.
- N or ␣ = Authorizing entity has not attempted card verification or could not verify the CVD due to a security device error. (␣ indicates a blank character.)

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

O = Card verification was not performed, CVD was not on the card. Not all cards have a CVD value encoded. The card expiration date must be equal to or greater than an expiration date defined on the CPF to insure that the CVD field has been encoded. If the card expiration date is equal to or greater than the CPF date, the CVD checks are performed.

P = Card verification was not performed. Either the merchant ignored the CVD on purpose or the user falsely indicated no CVD was on the card.

R = Card verification was performed and the CVD was invalid. The situation was noted and the transaction should be referred.

U = The issuer has not certified or has not provided the encryption keys to the interchange.

Y = Card verification was performed and the CVD was valid.

14–18 02 CITY-EXT PIC X(5)

A city extension field. When the name of the city in which the terminal or retailer is located is longer than 13 characters, this field carries the last 5 characters of the city name. If the name of the city is longer than 18 characters, the name must be abbreviated. The information for this field comes from the BASE24-pos Terminal Data files or the POS Retailer Definition File (PRDF). A blank space indicates that no information is available.

19 02 COMPLETE-TRACK2-DATA PIC X(1)

A flag indicating whether the terminal or transaction acquirer can capture and transmit complete Track 1 or Track 2 data for card swipe transactions. Valid values are as follows:

Y or ␣ = Yes, the terminal or transaction acquirer captures and transmits complete track data. (␣ indicates a blank space.)

N = No, the terminal or transaction acquirer does not capture and transmit complete track data.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 20 | 02 | UAF-FLG | PIC X(1) |

A flag indicating whether the cardholder had a Usage Accumulation File (UAF) record when the transaction request was processed by the Router/Authorization process. This field is used to limit the number of accesses required to update the UAF. This field is only used with the Negative Authorization with Usage Accumulation method under authorization level 3 (online/offline). Valid values are as follows:

␣ = UAF record did not exist. (␣ indicates a blank space.)

1 = UAF record did exist.

# Token 05 Check Guarantee/Verification Token

The fields in the Check Guarantee/Verification token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–78 | | CHK-AUTH-TKN | |
| 1–40 | 02 | CHK-ID-NUM<br>The identification number from the ID used for the transaction. | PIC X(40) |
| 41–42 | 02 | CHK-ID-TYP<br>The type of identification used for the transaction. Valid values are as follows:<br>01 = Credit card<br>02 = Drivers license<br>03 = Checking account number<br>04 = Debit card<br>05 = Proprietary check cashing card<br>06 = State ID<br>07 = Social security number<br>08 = Student ID<br>09 = Employee ID<br>10 = Passport<br>11 = MICR data | PIC X(2) |
| 43–44 | 02 | STATE-CDE<br>The state code associated with the check guarantee/ verification transaction. | PIC X(2) |
| 45–50 | 02 | BIRTH-DAT<br>The birth date of the customer. | PIC X(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 51–61 | 02 | CHK-RTG-GRP<br>The routing group to which the check guarantee/verification transaction belongs. | PIC X(11) |
| 62 | 02 | CHK-PROVIDER<br>A code identifying the entity that guaranteed the funds (for example, TeleCredit, TeleCheck). Valid values are as follows:<br>0 = None<br>1 = 861400<br>2 = 813500<br>3 = 894400<br>4 = 810000<br>5 = 763060<br>6 = 763057<br>7 = 762135<br>8 = 730191<br>9 = 730151<br>A = 418532 | PIC X(1) |
| 63–77 | 02 | CHK-MRCHNT-ID<br>The ID by which the check acceptance vendor refers to the merchant. | PIC X(15) |
| 78 | 02 | USER-FLD1 | PIC X(1) |

# Token 10 American Express Token—Binary Format

The fields in the binary format American Express token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–108 | | AMEX-TKN | |
| 1–2 | 02 | FRMT-CDE | PIC X(2) |
| | | The AMEX Industry Format Code identifying the type of industry format in which the device is sending AMEX data. Valid values are as follows:<br>05 = Auto rental<br>11 = Lodging<br>12 = Restaurant<br>20 = General retail<br>21 = Oil company | |
| 3–108 | 02 | AMEX-DATA | PIC X(106) |
| 3–108 | 02 | AUTO-RENT | REDEFINES AMEX-DATA |
| | | The following fields contain car rental information. These fields are used when the FRMT-CDE field contains the value 05 (Auto rental). | |
| 3–10 | 04 | AUDIT-ADJ-AMT | TYPE BINARY 64 SIGNED |
| | | The amount of any charges that were added to the contract after the vehicle was checked in (for example, mileage or damages). | |
| 11–19 | 04 | AGREEMENT-NUMBER | PIC X(9) |
| | | The invoice number of the rental agreement (issued by the rental agency) signed by the card member. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 20–27 | 04 | REFERENCE-NUMBER<br>A reference number used by American Express to obtain supporting information for a charge from the Service Establishment. | PIC 9(8) |
| 28–45 | 04 | RENTAL-CITY<br>The city in which the auto was rented. | PIC X(18) |
| 46–47 | 04 | RENTAL-ST<br>The two-character state code identifying the state in which the auto was rented. | PIC X(2) |
| 48–53 | 04 | RENTAL-DAT<br>The date (YYMMDD) on which the car was rented. | PIC 9(6) |
| 54–57 | 04 | RENTAL-TIM<br>The time (hhmm), in 24-hour format, at which the car was rented. | PIC 9(4) |
| 58–75 | 04 | RETURN-CITY<br>The city in which the car was returned. | PIC X(18) |
| 76–77 | 04 | RETURN-ST<br>The two-character state code identifying the state in which the auto was returned. | PIC X(2) |
| 78–83 | 04 | RETURN-DAT<br>The date (YYMMDD) on which the car was returned. | PIC 9(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 84–87 | 04 | RETURN-TIM<br>The time (hhmm), in 24-hour format, at which the car was returned. | PIC 9(4) |
| 88–107 | 04 | RENTER-NAME<br>The name of the person who rented the car. | PIC X(20) |
| 108 | 04 | USER-FLD1 | PIC X(1) |
| 3–108 | 02 | LDG<br>The following fields contain lodging information. These fields are used when the FRMT-CDE field contains the value 11 (Lodging). | REDEFINES AMEX-DATA |
| 3–11 | 04 | REFERENCE-CDE<br>A reference number used by American Express to obtain supporting information for a charge from the Service Establishment. | PIC 9(9) |
| 12 | 04 | CHARGE-TYPE<br>The type of purchase associated with the charge. Valid values are as follows:<br>0 = Reserved<br>1 = Lodging<br>2 = Restaurant<br>3 = Gift shop<br>4–9 = Reserved | PIC 9(1) |
| 13–22 | 04 | TAB-ROC-ID<br>The original Record of Charge (ROC) invoice number, or another number associated with the charge. | PIC X(10) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 23–30 | 04 | TAX-AMT | TYPE BINARY 64 SIGNED |
| | | The total tax amount in dollars and cents. | |
| 31–36 | 04 | ARRIVAL-DAT | PIC 9(6) |
| | | The arrival date (MMDDYY) of the guest. | |
| 37–42 | 04 | DEPART-DAT | PIC 9(6) |
| | | The departure date (MMDDYY) of the guest. | |
| 43–50 | 04 | ROOM-RATE | TYPE BINARY 64 SIGNED |
| | | The per diem rate charged for the customer's stay at the establishment. | |
| 51 | 04 | PROGRAM-CDE | PIC 9(1) |
| | | Indicates the reason for the charge.<br>If there is no special circumstance, the value 1 must appear in this field. Valid values are as follows:<br>0 = Reserved<br>1 = Used if no other code pertains<br>2 = Assured reservation—no show<br>3 = CARDeposit<br>4 = Delayed charge<br>5 = Express service<br>6 = Assured reservation<br>7–9 = Reserved | |
| 52–108 | 04 | USER-FLD2 | PIC X(57) |
| 3–108 | 02 | RSTRNT | REDEFINES AMEX-DATA |
| | | The following fields contain restaurant information. These fields are used when the FRMT-CDE field contains the value 12 (Restaurant). | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–11 | 04 | REFERENCE-CDE | PIC 9(9) |
| | | A reference number used by American Express to obtain supporting information for a charge from the Service Establishment. | |
| 12–15 | 04 | CHARGE-CDE1 | PIC X(4) |
| | | Identifies the type of purchase associated with the charge. This field is defined by American Express. | |
| 16–25 | 04 | TAB-ROC-ID | PIC X(10) |
| | | The original Record of Charge (ROC) invoice number, or another number associated with the charge. | |
| 26 | 04 | DESCRIPTION-CDE | PIC X(1) |
| | | A code defining the type of charge. Valid values are as follows:<br>0 = Food<br>1 = Food or beverage<br>2 = B-MY-GST<br>3–9, A–Z = Arbitrary | |
| 27–34 | 04 | TAX-AMT | TYPE BINARY 64 SIGNED |
| | | The total tax amount in dollars and cents. | |
| 35–42 | 04 | FOOD-AMT | TYPE BINARY 64 SIGNED |
| | | The total cost of the food, or the food and beverages, if these charges are combined on the original Record of Charge. | |
| 43–50 | 04 | BEVERAGE-AMT | TYPE BINARY 64 SIGNED |
| | | The total cost of beverages, if itemized separately from the food. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 51 | 04 | TIP1-CDE | PIC X(1) |
| | | A code identifying an employee receiving a tip. | |
| 52 | 04 | TIP2-CDE | PIC X(1) |
| | | A code identifying a second employee receiving a tip. | |
| 53–60 | 04 | TIP1-AMT | TYPE BINARY 64 SIGNED |
| | | The amount of the tip given to the employee identified by the TIP1-CDE field. | |
| 61–68 | 04 | TIP2-AMT | TYPE BINARY 64 SIGNED |
| | | The amount of the tip given to the employee identified by the TIP2-CDE field. | |
| 69–108 | 04 | USER-FLD5 | PIC X(40) |
| 3–108 | 02 | GEN-RETAIL | REDEFINES AMEX-DATA |
| | | The following fields contain retail charges information. These fields are used when the FRMT-CDE field contains the value 20 (General retail). | |
| 3–10 | 04 | TAX-AMT | TYPE BINARY 64 SIGNED |
| | | The total tax amount in dollars and cents. | |
| 11–19 | 04 | REFERENCE-CDE | PIC 9(9) |
| | | A reference number used by American Express to obtain supporting information for a charge from the Service Establishment. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 20–23 | 04 | CHARGE-CDE1<br>A code used to identify the type of purchase associated with the charge. This field is defined by American Express. | PIC X(4) |
| 24–33 | 04 | TAB-ROC-ID<br>The original Record of Charge (ROC) invoice number, or another number associated with the charge. | PIC X(10) |
| 34–37 | 04 | CHARGE-CDE2<br>A code used to identify the type of purchase associated with the charge. This field is defined by American Express. | PIC X(4) |
| 38–41 | 04 | CHARGE-CDE3<br>A code used to identify the type of purchase associated with the charge. This field is defined by American Express. | PIC X(4) |
| 42–45 | 04 | CHARGE-CDE4<br>A code used to identify the type of purchase associated with the charge. This field is defined by American Express. | PIC X(4) |
| 46–49 | 04 | CHARGE-CDE5<br>A code used to identify the type of purchase associated with the charge. This field is defined by American Express. | PIC X(4) |
| 50–108 | 04 | USER-FLD3 | PIC X(59) |
| 3–108 | 02 | OIL<br>The following fields contain oil company information. These fields are used when the FRMT-CDE field contains the value 21 (Oil company). | REDEFINES AMEX-DATA |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–14 | 04 | REFERENCE-CDE<br>A reference number used by American Express to obtain supporting information for a charge from the Service Establishment. | PIC 9(12) |
| 15–18 | 04 | CHARGE-CDE1<br>A code used to identify the type of purchase associated with the charge. This field is defined by American Express. | PIC X(4) |
| 19–28 | 04 | TAB-ROC-ID<br>The original Record of Charge (ROC) invoice number, or another number associated with the charge. | PIC X(10) |
| 29–36 | 04 | TAX-AMT<br>The total tax amount in dollars and cents. | TYPE BINARY 64 SIGNED |
| 37–108 | 04 | USER-FLD4 | PIC X(72) |

# Token 10 American Express Token—ASCII Format

The fields in the ASCII format American Express token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–118 | | AMEX-TKNX | |
| 1–2 | 02 | FRMT-CDE | PIC X(2) |
| 3–118 | 02 | AMEX-DATA | PIC X(116) |
| 3–118 | 02 | AUTO-RENT | REDEFINES AMEX-DATA |
| 3–20 | 04 | AUDIT-ADJ-AMT | PIC 9(18) |
| 21–29 | 04 | AGREEMENT-NUMBER | PIC X(9) |
| 30–37 | 04 | REFERENCE-NUMBER | PIC 9(8) |
| 38–55 | 04 | RENTAL-CITY | PIC X(18) |
| 56–57 | 04 | RENTAL-ST | PIC X(2) |
| 58–63 | 04 | RENTAL-DAT | PIC 9(6) |
| 64–67 | 04 | RENTAL-TIM | PIC 9(4) |
| 68–85 | 04 | RETURN-CITY | PIC X(18) |
| 86–87 | 04 | RETURN-ST | PIC X(2) |
| 88–93 | 04 | RETURN-DAT | PIC 9(6) |
| 94–97 | 04 | RETURN-TIM | PIC 9(4) |
| 98–117 | 04 | RENTER-NAME | PIC X(20) |
| 118 | 04 | USER-FLD1 | PIC X(1) |
| 3–118 | 02 | LDG | REDEFINES AMEX-DATA |
| 3–11 | 04 | REFERENCE-CDE | PIC 9(9) |
| 12 | 04 | CHARGE-TYPE | PIC 9(1) |
| 13–22 | 04 | TAB-ROC-ID | PIC X(10) |
| 23–40 | 04 | TAX-AMT | PIC 9(18) |
| 41–46 | 04 | ARRIVAL-DAT | PIC 9(6) |
| 47–52 | 04 | DEPART-DAT | PIC 9(6) |
| 53–70 | 04 | ROOM-RATE | PIC 9(18) |
| 71 | 04 | PROGRAM-CDE | PIC 9(1) |
| 72–118 | 04 | USER-FLD2 | PIC X(47) |
| 3–118 | 02 | RSTRNT | REDEFINES AMEX-DATA |
| 3–11 | 04 | REFERENCE-CDE | PIC 9(9) |
| 12–15 | 04 | CHARGE-CDE1 | PIC X(4) |
| 16–25 | 04 | TAB-ROC-ID | PIC X(10) |
| 26 | 04 | DESCRIPTION-CDE | PIC X(1) |
| 27–44 | 04 | TAX-AMT | PIC 9(18) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 45–62 | 04 | FOOD-AMT | PIC 9(18) |
| 63–80 | 04 | BEVERAGE-AMT | PIC 9(18) |
| 81 | 04 | TIP1-CDE | PIC X(1) |
| 82 | 04 | TIP2-CDE | PIC X(1) |
| 83–100 | 04 | TIP1-AMT | PIC 9(18) |
| 101–118 | 04 | TIP2-AMT | PIC 9(18) |
| 3–118 | 02 | GEN-RETAIL | REDEFINES AMEX-DATA |
| 3–20 | 04 | TAX-AMT | PIC 9(18) |
| 21–29 | 04 | REFERENCE-CDE | PIC 9(9) |
| 30–33 | 04 | CHARGE-CDE1 | PIC X(4) |
| 34–43 | 04 | TAB-ROC-ID | PIC X(10) |
| 44–47 | 04 | CHARGE-CDE2 | PIC X(4) |
| 48–51 | 04 | CHARGE-CDE3 | PIC X(4) |
| 52–55 | 04 | CHARGE-CDE4 | PIC X(4) |
| 56–59 | 04 | CHARGE-CDE5 | PIC X(4) |
| 60–118 | 04 | USER-FLD3 | PIC X(59) |
| 3–118 | 02 | OIL | REDEFINES AMEX-DATA |
| 3–14 | 04 | REFERENCE-CDE | PIC 9(12) |
| 15–18 | 04 | CHARGE-CDE1 | PIC X(4) |
| 19–28 | 04 | TAB-ROC-ID | PIC X(10) |
| 29–46 | 04 | TAX-AMT | PIC 9(18) |
| 47–118 | 04 | USER-FLD4 | PIC X(72) |

# Token 11 Automated Clearing House (ACH) Debit Token

The fields in the Automated Clearing House (ACH) Debit token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–12 | | ACH-DB-TKN | |
| 1–11 | 02 | RTTN<br>The routing and transit number (RTTN) of the account-issuing institution. | PIC X(11) |
| 12 | 02 | USER-FLD1 | PIC X(1) |

# Token 16 Alternate Merchant ID Token

The fields in the Alternate Merchant ID token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–16 | | ALT-MERCH-ID-TKN | |
| 1–15 | 02 | ALT-MERCH-ID<br>The alternate merchant ID defined by card type. The alternate merchant ID can be assigned by interchanges to merchants that accept their cards. | PIC X(15) |
| 16 | 02 | USER-FLD1 | PIC X(1) |

# Token 17 Visa Payment Service 2000 Token

The fields in the Visa Payment Service 2000 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–24 | | PS2000-TKN | |
| 1 | 02 | SRV-IND | PIC X(1) |

A code used to provide additional information regarding the disposition of the transaction. Valid values are as follows:

| | | |
|---|---|---|
| A | = | Transaction approved for PS2000 by Visa |
| C | = | Meets cardholder activated terminal requirements for PS2000 |
| E | = | Meets card-present requirements for PS2000 and merchant name and location are present |
| F | = | Meets CPS/Account funding requirements |
| I | = | Incremental authorization to a previously approved transaction |
| K | = | Valid retail key entry; card present |
| M | = | Meets national payment service requirements |
| N | = | Transaction downgraded for PS2000 |
| P | = | Preferred customer |
| R | = | Recurring payment |
| S | = | 3-D Secure CAVV attempt transaction |
| T | = | No CPS program available |
| U | = | Meets Preferred CPS/Electronic Commerce program (3-D Secure) |
| V | = | Meets address verification requirements for PS2000 |
| W | = | Meets requirements for Basic CPS/Electronic Commerce program (non-3-D Secure) |
| Y | = | Transaction submitted for PS2000 |
| 0 or ␣ | = | CPS qualification info not received (where ␣ is a blank space) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 2–16 | 02 | TRAN-ID | PIC X(15) |
| | | A Visa-generated Transaction Identifier (TID) that is unique for each original authorization and financial request. The identified links original messages to subsequent messages, such as those for exception item processing and clearing records. The TID is a key element in both CPS and CRS processing. | |
| 17–20 | 02 | VALID-CDE | PIC X(4) |
| | | A value calculated by V.I.P. to ensure that key fields in authorization messages match their respective fields in the BASE II deferred clearing message. This field can also contain a downgrade reason code for authorization requests that fail CPS qualification. | |
| 21 | 02 | MKT-SPFC-DATA-ID | PIC X(1) |
| | | A code identifying the industry for which market-specific data has been provided in the duration and prestigious property indicator fields. Valid values are as follows:<br>A = Auto rental<br>B = Bill payment<br>E = Electronic commerce transaction aggregation<br>H = Hotel<br>M = Healthcare (medical)<br>N = Failed market-specific edits<br>T = Transit (healthcare transactions only) | |
| 22–23 | 02 | DUR | PIC 9(2) |
| | | The number of days anticipated for the auto rental or hotel stay. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 24 | 02 | PRSTGS-PROP-IND | PIC X(1) |

A code indicating the property's floor limit for participants in Visa's Prestigious Lodging program. This field is required when the merchant requests a $1.00 status check. Valid values are as follows:

D = Visa Classic ($500 limit), Visa Gold ($1000 limit), Visa Business ($1500 limit)
B = $1000 limit
S = $1500 limit

# Token 19 Visa Payment Service 2000 Offline Token

The fields in the Visa Payment Service 2000 Offline token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–60 | | PS2000-OFFL-TKN<br>This token contains information from Visa used in the clearing of the transaction. | |
| 1–26 | 02 | PUR-ID<br>Identifies the purchase to the issuer and the cardholder. | |
| 1 | 04 | FRMT-ID<br>The type number of the purchase identifier. Valid values are as follows:<br>1 = Order number (direct marketing)<br>3 = Rental agreement number (automobile rental)<br>4 = Hotel folio number (hotel) | PIC X |
| 2–26 | 04 | INFO<br>The actual number of the purchase. | PIC X(25) |
| 27–32 | 02 | CHK-DAT<br>The date (YYMMDD) the customer picked up the automobile or checked into the hotel. | PIC 9(6) |
| 33 | 02 | NO-SHW-IND<br>Indicates whether the clearing message is for a no-show. This field contains a 1 when the message is for a no-show. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 34–39 | 02 | EXTRA-CHRGS | PIC X(6) |

Codes indicating additional charges that are added to the customer's bill after the hotel check-out or auto rental return. Valid values for automobile rental are as follows:

1 = Gasoline
2 = Extra mileage
3 = Late return
4 = One-way service fee (drop off charge)
5 = Parking or moving violation

Valid values for hotels are as follows:

2 = Restaurant or room service
3 = Gift shop
4 = Mini-bar
5 = Telephone
6 = Other
7 = Laundry

| | | | |
|---|---|---|---|
| 40–41 | 02 | MULT-CLRNG-SEQ-NUM | PIC 9(2) |

Identifies a specific BASE II clearing transaction record when multiple clearing records are submitted for single authorization.

| | | | |
|---|---|---|---|
| 42–43 | 02 | MULT-CLRNG-SEQ-CNT | PIC 9(2) |

Identifies the total number of BASE II clearing transactions when multiple clearing records are submitted for a single authorization.

| | | | |
|---|---|---|---|
| 44 | 02 | RSTRCTD-TCKT-IND | PIC X(1) |

Indicates whether the transaction was for the purchase of nonrefundable tickets. This field contains a 1 when the purchase was for nonrefundable tickets.

| | | | |
|---|---|---|---|
| 45–56 | 02 | TOT-AMT-AUTH | PIC 9(12) |

The total amount authorized when multiple authorizations or reversals were submitted for a single clearing.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 57 | 02 | RQSTD-PYMNT-SRVC | PIC X(1) |

The specific custom payment service the acquirer requested. Valid values are as follows:

1 = Passenger Transport 1
2 = Passenger Transport 2
3 = Hotel 1
4 = Hotel 2
5 = Auto Rental 1
6 = Auto Rental 2
7 = Direct Marketing
8 = Automated Fuel Dispenser
A = Retail

| | | | |
|---|---|---|---|
| 58–59 | 02 | CHRGBCK-RGHTS-IND | PIC X(2) |

The authorization-related chargeback protection level for the custom payment service. Valid values are as follows:

00 = Card present/Non-T&E
01 = Card present/T&E
02 = Card not present AVS/Non-T&E
03 = Card not present/T&E
04 = Card not present—preferred customer/T&E
05 = Card not present—no signature/Non-T&E

| | | | |
|---|---|---|---|
| 60 | 02 | USER-FLD1 | PIC X(1) |

# Token 20 Interchange Compliance Token

The fields in the Interchange Compliance token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–22 | | ICHG-COMPLIANCE TKN<br>This token is used to carry interchange compliance data for MasterCard. | |
| 1 | 02 | LIFE-CYCLE-IND<br>The indicator acquirers are required to forward in their clearing transactions.<br>b = Single authorization (where b indicates a blank space)<br>F = Force post settlement message where incremental authorizations may have been received<br>I = Incremental authorization<br>O = Original authorization for which incremental authorizations may be received | PIC X(1) |
| 2–16 | 02 | TRACE-ID<br>The code assigned by the interchange to a transaction that has met the required compliance edits. A combination of the Network ID, Reference number, and date is filled into this field, depending on the interchange. | PIC X(15) |
| 17–20 | 02 | VALID-CDE<br>The code assigned by the interchange to a transaction that has met the required compliance edits and has been approved by the issuer. | PIC X(4) |
| 21 | 02 | MONITORING-STAT<br>A code returned from the interchange indicating whether MasterCard changed the Point of Service Entry Mode from 90 to 02. A value of Y indicates that the status is being monitored. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 22 | 02 | ERR-IND | PIC X(1) |

A code returned from the interchange indicating an error condition that may have occurred. Valid values are as follows:

␣ = No error occurred (where ␣ indicates a blank space)

A = Track 1 or Track 2 data not present in message
B = Track 1 and Track 2 data present in message
C = PAN not equal in PAN data
D = Expiration date not equal in PAN data
E = Card type invalid in track data
F = Field separator(s) invalid in track data
G = A field within the track data exceeds the maximum length
H = Transaction category code is T
I = POS customer presence indicator is 1
J = POS card presence indicator is 1

# Token 29 Check Guarantee/Verification 2 Token

The Check Guarantee/Verification 2 token contains electronic check conversion information. The fields in the Check Authorization 2 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–68 | | CHK-AUTH2-TKN | |
| 1–12 | 02 | CHK-NUM<br>The check serial number filled in by the Interchange Interface process or the Device Handler module. | PIC X(12) |
| 13–18 | 02 | PROC-CNTL-NUM<br>The process control number from the check. This field is filled in by the Interchange Interface process or the Device Handler module. | PIC X(6) |
| 19–38 | 02 | PHONE-NUM<br>The customer's phone number if available. This field is filled in by the Interchange Interface process or the Device Handler module. | PIC X(20) |
| 39-44 | 02 | PROC-CDE<br>The external message processing code of the acquiring interchange. | PIC X(6) |
| 45 | 02 | AUTH-TYP<br>The authorization type obtained from the Card Authorization Routing Table File (CART). This field is filled in and used by the Authorization module. This field is blank filled by the Device Handler module and Interchange Interface process. Valid values are as follows:<br>0 = Host authorization<br>3 = Positive balance authorization | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 46 | 02 | AUTH-LVL | PIC X(1) |
| | | The authorization level obtained from the CART. This field is filled in and used by the Authorization module. This field is blank filled by the Device Handler module and Interchange Interface process. Valid values are as follows:<br>0 = Online (no PBF update)<br>1 = Online (with PBF update)<br>2 = Offline<br>3 = Online/offline | |
| 47 | 02 | CHK-SPF | PIC X(1) |
| | | This field is reserved for future use. | |
| 48 | 02 | SETL-FLG | PIC X(1) |
| | | The settlement flag. Interchange Interface processes can set this value. Valid values are as follows:<br>0 = No settlement.<br>1 = Settled by interchange.<br>2 = Automated clearinghouse (ACH) settled. | |
| 49 | 02 | CONV-FLG | PIC X(1) |
| | | A flag indicating the type of electronic conversion to perform. This field can be set by Device Handler modules and Interchange Interface processes. Valid values are as follows:<br>0 = No conversion.<br>1 = Perform a check verification or check guarantee transaction with conversion.<br>2 = Conversion only transaction. | |
| 50–68 | 02 | USER-FLD1 | PIC X(19) |
| | | This field is not currently used. | |

# Token 31 Check Callback Token

The fields in the Check Callback token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–90 | 00 | CHK-CALLBACK-TKN | |
| 1–25 | 02 | AUTH-NAM<br>The name of the non-bank authorizer's name. This field is set by the Interchange Interface process. | PIC X(25) |
| 26–69 | 02 | ADDR<br>The following fields contain the non-bank authorizer's address. These fields are set by the Interchange Interface process. | |
| 26–45 | 04 | STREET | PIC X(20) |
| 46–58 | 04 | CITY | PIC X(13) |
| 59–60 | 04 | ST | PICX(2) |
| 61–69 | 04 | POSTAL-CDE | PIC X(9) |
| 70–89 | 02 | PHONE-NUM<br>The non-bank authorizer's customer service phone number. This field is set by the Interchange Interface process. | PIC X(20) |
| 90 | 02 | USER-FLD1<br>This field is not currently used. | PIC X(1) |

# Token C0 BASE24-pos Release 5.1 Token

The fields in the BASE24-pos Release 5.1 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–26 | | PS51-TKN | |
| 1–4 | 02 | CVD-FLD | PIC X(4) |
| | | Manually entered values such as the American Express Card Identification Code (CID), MasterCard Card Validation Codes (CVC2), and Visa Card Verification Value 2 (CVV2). Only the leftmost three bytes are used in CVC2 and CVV2 verification processing. A blank space indicates that no information is available. | |
| 5 | 02 | RESUB-STAT | PIC X(1) |
| | | A code indicating the status for resubmission. Valid values are as follows:<br>␣ = Normal transaction (where ␣ indicates a blank space)<br>A = Resubmission approved<br>D = Resubmission hard declined<br>R = Resubmission<br>S = Store and Forward submission<br>T = Resubmission for tries or time out condition<br>A blank space indicates that no information is available. | |
| 6–8 | 02 | RESUB-CNTR | PIC X(3) |
| | | The number of times the transaction has been submitted for processing. A blank space indicates that no information is available. | |
| 9–18 | 02 | TERM-POSTAL-CDE | PIC X(10) |
| | | The 10-byte postal ZIP code of the location of the terminal. A blank space indicates that no information is available. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 19 | 02 | E-COM-FLG | PIC X(1) |

The electronic commerce flag. Valid values are as follows:

- ␣ = Not applicable, or acquirer did not specify (where ␣ indicates a blank space)
- 0 = Not an electronic commerce transaction
- 1 = Single mail or telephone order transaction
- 2 = Recurring mail or telephone order transaction
- 3 = Mail or telephone order installment payment
- 4 = Mail or telephone order unknown classification
- 5 = Secure electronic commerce transaction with cardholder authentication or authentication value
- 6 = Encrypted electronic commerce transaction where the merchant is capable of authenticating the cardholder but was unable to complete the authentication, (e.g., because the issuer or cardholder does not participate in the appropriate authentication program)
- 7 = Encrypted electronic commerce transaction
- 8 = Non-secure electronic commerce transaction
- 9 = Non-authenticated security transaction that does not comply with secure electronic transaction and the merchant supports secure electronic transactions
- S = Internet electronic delivery for AMEX transactions only
- T = Internet physical delivery for AMEX transactions only

| | | | |
|---|---|---|---|
| 19 | 02 | MOTO-FLG | REDEFINES E-COM-FLG |

The mail/telephone order flag. Valid values are as follows:

- 0 = Not a mail or telephone order transaction
- 1 = Single transaction
- 2 = Recurring transaction
- 3 = Installment payment
- 4 = Unknown classification

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 20 | 02 | CMRCL-CRD-TYP | PIC X(1) |

A code indicating the commercial card type. Valid values are as follows:

␣ = Normal transaction (where ␣ indicates a blank space)
0 = Request for card type
B = Business card
R = Corporate card
S = Purchasing card

| | | | |
|---|---|---|---|
| 21 | 02 | ADNL-DATA-IND | PIC X(1) |

A code indicating whether additional data was captured with the original transaction. Valid values are as follows:

0 = No additional data was captured with the original transaction.
1 = Additional data was captured with the original transaction.
␣ = This field is not used (where ␣ is a blank space).

| | | | |
|---|---|---|---|
| 22 | 02 | CVD-FLD-PRESENT | PIC X(1) |

A code indicating whether the Card Verification Value is on the card. Valid values areas follows:

␣ = No information available (where ␣ is a blank space).
0 = CVD2/CSC value is deliberately bypassed or not provided by the merchant.
1 = CVD2/CSC value is present.
2 = CVD2/CSC value is on the card but is illegible.
3 = The expiration date is not required to be sent from the interchange, so the CVD2 field cannot be validated (for example, a BASE24-billpay transaction).
9 = Cardholder states that the card has no CVD2/CSC imprint.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 23 | 02 | SAF-OR-FORCE-POST | PIC X(1) |

A code indicating whether the transaction is a store-and-forward or force-post transaction. Valid values are as follows:

␣ = Neither a force-post nor a store-and-forward transaction (where ␣ is a blank space). This is the default.
F = Force-post transaction.
S = Store-and-forward transaction.

| | | | |
|---|---|---|---|
| 24 | 02 | AUTHN-COLL-IND | PIC X(1) |

The authentication collection indicator. Valid values are as follows:

0 = Universal Cardholder Authentication field (UCAF) data collection is not supported at the merchant's Web site.
1 = Universal Cardholder Authentication field (UCAF) data collection is supported by the merchant, but UCAF data was not populated.
2 = Universal Cardholder Authentication field (UCAF) data collection is supported by the merchant, and UCAF data was populated.
3 = Universal Cardholder Authentication field (UCAF) data collection is supported by the merchant, and UCAF data was populated by a static authentication value.

| | | | |
|---|---|---|---|
| 25 | 02 | FRD-PRN-FLG | PIC X(1) |

The fraud prone outlet flag. This flag indicates the fraud rating of the retailer (field 48 in the inter-member message). Valid values are as follows:

0 = Not fraud prone
1 = Prone to fraud
2 = Highly fraud prone

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 26 | 02 | CAVV-AAV-RSLT-CDE | PIC X(1) |

The CAVV/AAV result code field indicates the result of the CAVV (VISA method) or AAV (MasterCard method) validation. Valid values are as follows:

- ␣ = Not validated yet (where ␣ is a blank space).
- 0 = Not validated due to erroneous data.
- 1 = Failed validation - authentication.
- 2 = Passed validation - authentication.
- 3 = CAVV passed validation - attempt. Authentication was attempted at the issuer's ACS but did not complete successfully.
- 4 = CAVV failed validation - attempt at issuer's ACS.
- 5 = Reserved for future use.
- 6 = CAVV not validated; issuer not participating in CAVV validation.
- 7 = CAVV failed validation - attempt.
- 8 = CAVV passed validation - attempt. Authentication was attempted at the interchange's ACS but did not complete successfully.
- 9 = CAVV failed validation - attempt at interchange during stand-in.
- A = CAVV passed validation - attempt at interchange during stand-in.
- B = CAVV passed validation - information only, no liability shift.
- C = CAVV not validated, attempt. This issuer did not return the results code in the authorization response.
- D = CAVV was not validated - authentication. The issuer failed to return the result value.

The following values are generated by BASE24 only:

- W = CAVV/AAV validation could not be performed (no EAF).
- X = CAVV/AAV validation could not be performed due to system error, or failure prevented authentication (error accessing EAF).
- Y = The acquirer is participating in authentication but the issuer is not participating.
- Z = Duplicate CAVV/AAV.

# Token C1 Station ID Token

The fields in the Station ID token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–16 | | STA-ID-TKN | |
| 1–16 | 02 | STA-ID<br>The 16-byte station ID. This field is used to track the station or port usage for a given transaction. | PIC X(16) |

# Token C2 Purchase Card and Fleet Card Token

The fields in the Purchase Card and Fleet Card token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–876 | | PURCHASE-TKN | |
| 1–30 | 02 | PURCHASE | |
| 1 | 04 | CRD-TYP | PIC X(1) |
| | | A code indicating the type of card used in the transaction. Valid values are as follows:<br>A = American Express purchasing card<br>M = MasterCard purchasing card—level 3<br>C = MasterCard fleet card<br>F = Visa fleet card<br>V = Visa purchasing card—level 3<br>*b* = Level 2 purchasing card (where *b* is a blank space) | |
| 2–18 | 04 | CUST-REF-IND | PIC X(17) |
| | | The customer identification number. | |
| 19–30 | 04 | TAX-AMT | PIC 9(12) |
| | | The sales tax assessed and included in the transaction. | |
| 31–876 | 02 | TKN-DATA | PIC X(846) |
| 31–876 | 02 | MC-PURCHASE-DATA | REDEFINES TKN-DATA |
| | | The following fields contain MasterCard purchasing card data. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 31–39 | 04 | SHIP-FROM-CDE<br>The postal code from which the items were shipped. | PIC X(9) |
| 40–48 | 04 | DEST-POSTAL-CDE<br>The postal code to which items will be delivered. | PIC X(9) |
| 49–51 | 04 | DEST-CNTRY-CDE<br>A code indicating the country to which items will be delivered. | PIC X(3) |
| 52–68 | 04 | MRCH-REF-NUM<br>The reference number supplied by the merchant for records management. | PIC X(17) |
| 69–77 | 04 | FREIGHT-AMT<br>The freight charges portion of the transaction amount. | PIC X(9) |
| 78–86 | 04 | DUTY-AMT<br>The importing fee (duty) assessed for the transaction. | PIC X(9) |
| 87–98 | 04 | PROD-CDE<br>The product code for the item. | PIC X(12) |
| 99–133 | 04 | DESCR<br>The description of the purchased item. | PIC X(35) |
| 134–137 | 04 | QTY<br>The number of items purchased. | PIC X(4) |
| 138–140 | 04 | UNIT-OF-MEASURE<br>A code indicating the unit of measure. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 141–149 | 04 | EXTENDED-AMT<br>The total amount of items purchased. | PIC X(9) |
| 150 | 04 | DB-CR-IND<br>A code indicating whether the extended amount is a debit or a credit. Valid values are as follows:<br>C = Credit<br>D = Debit | PIC X(1) |
| 151 | 04 | DISCOUNT-IND<br>A flag indicating whether a discount was applied to the purchase amount. Valid values are as follows:<br>Y = Yes, a discount was applied to the purchase amount.<br>N = No, a discount was not applied to the purchase amount. | PIC X(1) |
| 152–160 | 04 | DISCOUNT-AMT<br>The amount of the discount applied to the purchase amount. | PIC X(9) |
| 161 | 04 | NET-GROSS-IND<br>A flag indicating whether the amount includes a tax. Valid values are as follows:<br>Y = Yes, a tax is included in the amount.<br>N = No, a tax is not included in the amount. | PIC X(1) |
| 162 | 04 | SALES-TAX-IND<br>A flag indicating whether the amount includes sales tax. Valid values are as follows:<br>Y = Yes, sales tax is included in the amount.<br>N = No, sales tax is not included in the amount. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 163–167 | 04 | VAT-TAX-RATE<br>The value-added tax rate. Five decimal places are implied. | PIC 9(5) |
| 168–171 | 04 | VAT-TAX-TYP<br>The type of value-added tax. | PIC X(4) |
| 172–180 | 04 | VAT-TAX-AMT<br>The amount of the value-added tax. | PIC X(9) |
| 181 | 04 | ALT-TAX-AMT-IND<br>A flag indicating whether an alternate tax is included in the purchase amount. Valid values are as follows:<br>Y = Yes, an alternate tax is included in the purchase amount.<br>N = No, an alternate tax is not included in the purchase amount. | PIC X(1) |
| 182–190 | 04 | ALT-TAX-AMT<br>The amount of the alternate tax. | PIC X(9) |
| 191-205 | 04 | ALT-TAX-ID<br>The alternate tax ID number. | PIC X(15) |
| 206 | 04 | USER-FLD1<br>Reserved for future use. | PIC X(1) |
| 207–876 | 04 | USER-FLD2<br>Reserved for future use. | PIC X(670) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 31–876 | 02 | VISA-PURCHASE-DATA REDEFINES TKN-DATA<br>The following fields contain Visa purchasing card level 3 data. | |
| 31–42 | 04 | PROD-CDE<br>The product code of the item sold. | PIC X(12) |
| 43–68 | 04 | DESCR<br>The description of the item sold. | PIC X(26) |
| 69–80 | 04 | COMMODITY-CDE<br>The item commodity code. | PIC X(12) |
| 81–92 | 04 | QTY<br>The number of items purchased. Four decimal places are implied. | PIC 9(12) |
| 93–104 | 04 | UNIT-OF-MEASURE<br>A code indicating the unit of measure. | PIC 9(12) |
| 105–116 | 04 | UNIT-COST<br>The cost for each item. Four decimal places are implied. | PIC 9(12) |
| 117–128 | 04 | VAT-TAX-AMT<br>The value-added tax amount. Two decimal places are implied. | PIC 9(12) |
| 129–132 | 04 | VAT-TAX-RATE<br>The value-added tax rate. Two decimal places are implied. | PIC 9(4) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 133–144 | 04 | DISCOUNT-AMT | PIC 9(12) |
| | | The amount of the discount applied to the purchase. Two decimal places are implied. | |
| 145–156 | 04 | TTL-AMT | PIC 9(12) |
| | | The total amount of items purchased. | |
| 157 | 04 | DETL-IND | PIC X(1) |
| | | A line item detail indicator. | |
| 158 | 04 | USER-FLD3 | PIC X(1) |
| | | Reserved for future use. | |
| 159–876 | 04 | USER-FLD4 | PIC X(718) |
| | | Reserved for future use. | |
| 31–876 | 02 | MC-FLEET-DATA | REDEFINES TKN-DATA |
| | | The following fields contain MasterCard fleet card data. | |
| 31–34 | 04 | BRAND-CDE | PIC 9(4) |
| | | The oil company brand code. | |
| 35 | 04 | SVC-TYP | PIC 9(1) |
| | | The type of service. Valid values are as follows:<br>1 = Self service<br>2 = Full service | |
| 36–37 | 04 | FUEL-PROD-CDE | PIC 9(2) |
| | | A two-digit code defined by MasterCard that identifies the product. Valid values are 01 through 29. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 38–42 | 04 | UNIT-COST<br>The fuel price per unit. Three decimal places are implied. | PIC 9(5) |
| 43 | 04 | UNIT-OF-MEASURE<br>The unit of measure. Valid values are as follows:<br>1 = Gallon<br>2 = Liter | PIC (9) |
| 44–49 | 04 | QTY<br>The quantity of fuel purchased. Two decimal places are implied. | PIC 9(6) |
| 50–58 | 04 | GROSS-FUEL-PRICE<br>The gross fuel price. Two decimal places are implied. | PIC 9(9) |
| 59–65 | 04 | ODOMETER<br>The odometer reading at the time of purchase. | PIC 9(7) |
| 66–82 | 04 | VEHICLE-NUM<br>The vehicle identification number. | PIC 9(17) |
| 83–99 | 04 | ID-NUM<br>The drivers license number. | PIC 9(17) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 100 | 04 | PROD-TYP | PIC X(1) |
| | | A code read from the card track indicating which prompts occur at the POS device. Valid values are as follows:<br>1 = Prompt for ID number (drivers license number) and odometer reading.<br>2 = Prompt for vehicle identification number and odometer reading.<br>3 = Prompt for user-defined driver number and odometer reading.<br>4 = Prompt for odometer reading only.<br>5 = No prompt. | |
| 101–105 | 04 | TAX-EXEMPTION-AMT | PIC 9(5) |
| | | The tax amount for tax exempt fleets. Two decimal places are implied. | |
| 106–120 | 04 | ALT-TAX-ID | PIC X(15) |
| | | A code identifying an alternate tax. Two decimal places are implied. | |
| 121–122 | 04 | DCLN-RSN-CDE | PIC X(2) |
| | | A code indicating the reason the transaction was denied. Valid values are as follows:<br>01 = Invalid ID number<br>02 = Invalid drivers license number<br>03 = Invalid vehicle identification number | |
| 123 | 04 | NON-FUEL-ITEMS | PIC 9(1) |
| | | The number of nonfuel items included in the purchase. | |
| 124 | 04 | USER-FLD5 | PIC X(1) |
| | | Reserved for future use. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 125–876 | 04 | NON-FUEL-DATA<br>The following fields contain information about nonfuel purchases. | OCCURS 0 TO 8 TIMES DEPENDING ON NON-FUEL-ITEMS |
| | 06 | PROD-CDE<br>A two-digit code defined by MasterCard that identifies nonfuel products. Valid values are 30 through 99. | PIC X(12) |
| | 06 | DESCR<br>The description of the nonfuel product. | PIC X(35) |
| | 06 | QTY<br>The quantity of the nonfuel item purchased. | PIC X(5) |
| | 06 | UNIT-OF-MEASURE<br>The unit of measure of the nonfuel item. | PIC X(3) |
| | 06 | EXTND-AMT<br>The total purchase amount of the nonfuel item. | PIC X(9) |
| | 06 | DISCOUNT-IND<br>A flag indicating whether a discount was applied to the purchase amount. Valid values are as follows:<br>Y = Yes, a discount was applied to the purchase amount.<br>N = No, a discount was not applied to the purchase amount. | PIC X |
| | 06 | DISCOUNT-AMT<br>The amount of the discount applied to the price. | PIC X(9) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 06 | NET-GROSS-IND | PIC X |
| | | A flag indicating whether the amount includes a tax. Valid values are as follows:<br>Y = Yes, tax is included in the amount.<br>N = No, tax is not included in the amount. | |
| | 06 | VAT-TAX-RATE | PIC X(5) |
| | | The value-added tax rate. Five decimal places are implied. | |
| | 06 | VAT-TAX-TYP | PIC X(4) |
| | | The type of value added tax applied to the transaction. | |
| | 06 | TAX-AMT | PIC X(9) |
| | | The tax paid on the nonfuel items. | |
| | 06 | DB-CR-IND | PIC X |
| | | A code indicating whether the transaction is a debit or a credit. Valid values are as follows:<br>C = Credit<br>D = Debit | |
| 31–876 | 02 | VISA-FLEET-DATA | REDEFINES TKN-DATA |
| | | The following fields contain Visa fleet card data. | |
| 31 | 04 | PURCH-TYP | PIC X |
| | | A code indicating the type of merchandise purchased. Valid values are as follows:<br>1 = Fuel purchase<br>2 = Nonfuel purchase<br>3 = Fuel and nonfuel purchase | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 32–33 | 04 | FUEL-TYP<br>A two-character code defined by Visa indicating the type of fuel purchased. | PIC X(2) |
| 34 | 04 | UNIT-OF-MEASURE<br>A code indicating the unit of measure. Valid values are as follows:<br>G = Gallon<br>I = Imperial gallon<br>K = Kilo<br>L = Liter<br>P = Pound | PIC X |
| 35–46 | 04 | QTY<br>The quantity purchased. Four decimal places are implied. | PIC 9(12) |
| 47–58 | 04 | UNIT-COST<br>The unit cost. Four decimal places are implied. | PIC 9(12) |
| 59–70 | 04 | GROSS-FUEL-PRICE<br>The gross fuel price. Four decimal places are implied. | PIC 9(12) |
| 71–82 | 04 | NET-FUEL-PRICE<br>The net fuel price. Four decimal places are implied. | PIC 9(12) |
| 83–94 | 04 | NET-NON-FUEL<br>The net nonfuel price. Two decimal places are implied. | PIC 9(12) |
| 95–111 | 04 | VEHICLE-NUM<br>The vehicle number. | PIC 9(17) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 112–128 | 04 | ID-NUM<br>The driver identification number. | PIC 9(17) |
| 129–135 | 04 | ODOMETER<br>The odometer reading at the time of purchase. | PIC X(7) |
| 136–139 | 04 | VAT-TAX-RATE<br>The value-added tax rate. Two decimal places are implied. | PIC 9(4) |
| 140–151 | 04 | MISC-FUEL-TAX<br>The miscellaneous fuel tax amount. | PIC 9(12) |
| 152–163 | 04 | OTHER-TAX<br>The other tax amount. | PIC 9(12) |
| 164–175 | 04 | MISC-NON-FUEL-TAX<br>The miscellaneous nonfuel tax amount. Two decimal places are implied. | PIC 9(12) |
| 176 | 04 | SVC-TYP<br>A code indicating the type of fuel service. Valid values are as follows:<br>F = Full service<br>S = Self service | PIC X |
| 177–192 | 04 | PROD-CDE<br>A two-digit code defined by Visa that identifies a nonfuel product. Valid values are 30 through 99. | PIC X(2)<br>OCCURS 8 TIMES |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 193–204 | 04 | GROSS-NON-FUEL<br>The gross nonfuel price. | PIC 9(12) |
| 205 | 04 | MISC-FUEL-STAT<br>The miscellaneous fuel tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 206 | 04 | MISC-NON-FUEL-STAT<br>The miscellaneous nonfuel tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 207 | 04 | FED-NON-FUEL-EXCISE-STAT<br>The federal excise tax exemption status for nonfuel products. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 208–219 | 04 | FED-NON-FUEL-EXCISE-TAX<br>The federal excise tax amount for nonfuel products. Two decimal places are implied. | PIC 9(12) |
| 220 | 04 | FED-FUEL-EXCISE-STAT<br>The federal excise tax exemption status for fuel. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 221–232 | 04 | FED-FUEL-EXCISE-TAX<br>The federal excise tax amount for fuel products. Two decimal places are implied. | PIC 9(12) |
| 233 | 04 | ST-FUEL-STAT<br>The state motor fuel tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 234–245 | 04 | ST-FUEL-TAX<br>The state motor fuel tax amount. Two decimal places are implied. | PIC 9(12) |
| 246 | 04 | CNTY-FUEL-SALES-STAT<br>The county fuel sales tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 247–258 | 04 | CNTY-FUEL-SALES-TAX<br>The county fuel tax amount. Two decimal places are implied. | PIC 9(12) |
| 259 | 04 | ST-LOC-NON-FUEL-SALES-STAT<br>The nonfuel state and local sales tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 260–271 | 04 | ST-LOC-NON-FUEL-SALES-TAX<br>The nonfuel state and local tax amount. Two decimal places are implied. | PIC 9(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 272 | 04 | CNTY-FUEL-STAT<br>The county motor fuel tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 273–284 | 04 | CNTY-FUEL-TAX<br>The county motor fuel tax amount. Two decimal places are implied. | PIC 9(12) |
| 285 | 04 | CITY-FUEL-SALES-STAT<br>The city fuel tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 286–297 | 04 | CITY-FUEL-SALES-TAX<br>The city fuel tax amount. Two decimal places are implied. | PIC 9(12) |
| 298 | 04 | CITY-FUEL-STAT<br>The city motor fuel tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 299–310 | 04 | CITY-FUEL-TAX<br>The city motor fuel tax amount. Two decimal places are implied. | PIC 9(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | LOCAL-TAX | REDEFINES CITY-FUEL-TAX |
| 299–310 | 06 | AMT<br>The amount of the local tax. | PIC 9(12) |
| 311 | 04 | ST-SCND-FUEL-STAT<br>The secondary state fuel tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 312–323 | 04 | ST-SCND-FUEL-TAX<br>The secondary state fuel tax amount. Two decimal places are implied. | PIC 9(12) |
| 324 | 04 | FED-SALES-STAT<br>The federal sales tax exemption status. Valid values are as follows:<br>0 = Nonexempt<br>1 = Exempt | PIC X |
| 325–336 | 04 | FED-SALES-TAX<br>The federal sales tax amount. Two decimal places are implied. | PIC 9(12) |
| 325–336 | 04 | NATL-TAX | REDEFINES FED-SALES-TAX |
| 325–336 | 06 | AMT<br>The national tax sales amount. | PIC 9(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 337 | 04 | NATL-TAX-INCL<br>Indicates whether the transaction includes the national tax. Valid values are as follows:<br>0 = Not subject to tax<br>1 = Subject to tax | PIC X |
| 338 | 04 | LOCAL-TAX-INCL<br>Indicates wheter the transaction includes the local tax. Valid values are as follows:<br>0 = Local tax is not included<br>1 = State or provincial tax is included<br>2 = The transaction is not subject to tax | PIC X |
| 339–358 | 04 | MRCH-VAT-RGSTR-NUM<br>The merchant VAT registration number/single business reference number. | PIC X(20) |
| 359–371 | 04 | CUST-VAT-RGSTR-NUM<br>The customer VAT registration number. | PIC X(13) |
| 372–388 | 04 | CUST-ID<br>The customer code/customer reference identifier. | PIC X(17) |
| 389–403 | 04 | MSG-ID<br>The message identifier that is used to link the separate line item detail messages. | PIC X(15) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 404 | 04 | ADNL-DATA-IND<br>Used to indicate that additional data is provided. Valid values are as follows:<br>Y = Additional data is provided separately in the Visa BASE II Draft Data TC 50 text message transaction.<br>N = Additional data is not provided | PIC X |
| 405–408 | 04 | SUM-COMMODITY-CDE<br>The summary commodity code. | PIC 9(4) |
| 409-412 | 04 | FUEL-BRAND<br>The fuel brand. | PIC X(4) |
| 413-417 | 04 | FUEL-TXN-VALID-RSLT<br>The fuel transaction validation results. | PIC X(5) |
| 418 | 04 | FUEL-ACCPT-MDE<br>The fuel acceptance mode. | PIC X(1) |
| 419-438 | 04 | DRV-ID<br>The driver identification. | PIC X(20) |
| 439-448 | 04 | JOB-NUM<br>The job number. | PIC X(10) |
| 449-456 | 04 | FLEET-NUM<br>The fleet number. | PIC X(8) |
| 457-470 | 04 | VEHICLE-RGSTR-NUM<br>The vehicle registration number. | PIC X(14) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 471-476 | 04 | PROD-QUAL<br>The product qualifier. | PIC X(6) |
| 477-480 | 04 | EXPAND-FUEL-TYP<br>The expanded fuel type. | PIC X(4) |
| 481-640 | 04 | EXPAND-NON-FUEL<br>The expanded non-fuel data. | OCCURS 8 TIMES |
| 481-484 | 06 | PROD-CDE<br>The expanded non-fuel product code. | PIC X(4) |
| 485-488 | 06 | PROD-CDE-QTY<br>The expanded non-fuel product code quantity. | PIC X(4) |
| 489-500 | 06 | PROD-CDE-UNIT-COST<br>The expanded non-fuel product code unit cost. | PIC X(12) |
| 641-876 | 04 | USER-FLD6<br>Reserved for future use. | PIC X(236) |

# Token C3 Certificate Token

The Certificate token contains cardholder and merchant certificate serial numbers for MasterCard electronic commerce transactions. The fields in the Certificate token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–32 | | CERT-TKN | |
| 1–16 | 02 | MERCH-CERT-SERIAL-NUM<br>The merchant certificate serial numbers for MasterCard's Electronic Commerce scheme. | PIC X(16) |
| 17–32 | 02 | CRD-HLDR-CERT-SERIAL-NUM<br>The cardholder certificate serial numbers for MasterCard's Electronic Commerce scheme. | PIC X(16) |

# Token C4 Point of Service Data Token

The fields in the Point of Service Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–12 | | PT-SRV-DATA-TKN | |
| 1 | 02 | TERM-ATTEND-IND<br>A code indicating whether the terminal is attended by the card acceptor. Valid values are as follows:<br>0 = The terminal is attended.<br>1 = The terminal is unattended (for example, a cardholder-activated terminal or a personal computer).<br>2 = No terminal used (for example, voice or ARU authorization). | PIC 9 |
| 2 | 02 | TERM-OPER-IND<br>Reserved for future use. This field is zero filled. | PIC 9 |
| 3 | 02 | TERM-LOC-IND<br>A code indicating the location of the terminal. Valid values are as follows:<br>0 = The terminal is on the premises of the card acceptor facility.<br>1 = The terminal is off the premises of the card acceptor facility (merchant terminal in a remote location).<br>2 = The terminal is on the premises of the cardholder location (home personal computer).<br>3 = No terminal was used (voice or ARU authorization). | PIC 9 |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 4 | 02 | CRDHLDR-PRESENT-IND | PIC 9 |

A code indicating whether the cardholder is present at the POS terminal. Valid values are as follows:

0 = The cardholder is present
1 = The cardholder is not present—unspecified reason
2 = The cardholder is not present—mail or fax order
3 = The cardholder is not present—telephone or ARU order
4 = The cardholder not present—standing order or recurring transaction
5 = The cardholder is not present—electronic order (home personal computer or Internet)

| | | | |
|---|---|---|---|
| 5 | 02 | CRD-PRESENT-IND | PIC 9 |

A code indicating whether the card is present at the POS terminal. Valid values are as follows:

0 = The card is present.
1 = The card is not present.

| | | | |
|---|---|---|---|
| 6 | 02 | CRD-CAPTR-IND | PIC 9 |

A code indicating whether the terminal has card capture capabilities. Valid values are as follows:

0 = The terminal does not have card capture capabilities.
1 = The terminal has card capture capabilities.

| | | | |
|---|---|---|---|
| 7 | 02 | TXN-STAT-IND | PIC 9 |

A code indicating the purpose or status of the request. Valid values are as follows:

0 = Normal request
1 = Merchant authorization
2 = Secure phone order
3 = Installment inquiry request
4 = Preauthorized request
5 = Stand-in

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | 6 = Address verification request: verify shipping address<br>7 = Cash back<br>8 = Downtime submission request | |
| 8 | 02 | TXN-SEC-IND<br>A code indicating the card acceptor's security level. Valid values are as follows:<br>0 = No security concern<br>1 = Suspected fraud (merchant suspicious—code 10)<br>2 = Identification verified | PIC 9 |
| 9 | 02 | TXN-RTN-IND<br>Reserved for future use. This field is zero filled. | PIC 9 |
| 10 | 02 | CRDHLDR-ACTVT-TERM-IND<br>A code indicating whether the cardholder activated the terminal with a card, and if so, the level of security. Contactless/proximity terminals should be set to the value 2. Valid values are as follows:<br>0 = The transaction is not a cardholder activated terminal transaction<br>1 = Automated dispensing machine with PIN—level 1 security<br>2 = Self-service terminal—level 2 security<br>3 = Limited amount terminal—level 3 security<br>4 = In-flight commerce—level 4 security<br>5 = Script device<br>6 = Electronic commerce<br>7 = Radio frequency device | PIC 9 |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 11 | 02 | TERM-INPUT-CAP-IND | PIC 9 |

A code indicating the terminal capabilities for transferring the data on the card to the terminal. If the terminal supports both contact and contactless transactions, contactless capabilities (values 3 and 4) take priority over other values. Valid values are as follows:

0 = Unknown or unspecified
1 = No terminal used (voice or audio response unit authorization)
2 = Magnetic stripe reader
3 = Contactless chip (EMV)
4 = Contactless magnetic stripe
5 = Magnetic stripe reader and EMV-compatible ICC reader
6 = Key entry only
7 = Magnetic stripe reader and key entry
8 = Magnetic stripe reader, key entry, and EMV-compatible ICC reader
9 = EMV-compatible ICC reader

| | | | |
|---|---|---|---|
| 12 | 02 | CRDHLDR-ID-METHOD | PIC X |

A code indicating how the cardholder was verified at the point-of-service. Valid values are as follows:

0 = Unknown (default)
1 = Signature
2 = PIN
3 = None (Cardholder present)
4 = None (Cardholder not present)
5 = Authentication value
6 = Electronic signature analysis
7 = Biometrics
8 = Biographics
9 = Other

# Token C5 Increased Optional Data Token

The fields in the Increased Optional Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–250 | | OPT-DATA-TKN | |
| 1–250 | 02 | OPT-DATA<br>Optional data that is longer than 80 bytes in length. | PIC X(250) |

# Token C6 Trans Stain XID Token

The Trans Stain XID token carries the authentication information for secure internet transactions. The fields in the Trans Stain XID token are described below. These fields should be initialized to blanks.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–80 | | TRANS-STAIN-XID-TKN | |
| 1–40 | 02 | XID<br>A unique transaction identifier assigned to one of the following:<br>• A SET transaction.<br>• A Visa transaction using the 3-D Secure (Three-Domain Secure Electronic Commerce Verification Service) method. | PIC X(40) |
| 41–80 | 02 | TRANS-STAIN<br>A value used to authenticate the cardholder. This field is used by one of the following methods:<br>• For a SET transaction, this field contains a hash value calculated by applying a secure hash algorithm to the XID and CardSecret (a secret SET-defined value known only to the cardholder and the issuer of the cardholder certificate).<br>• For a Visa transaction using the 3-D Secure method, this field conains the Cardholder Authentication Verification Value (CAVV). | PIC X(40) |

# Token C7 Cardholder Serial Number Token

The fields in the Cardholder Serial Number token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–32 | | CRDHLDR-SERAIL-NUM-TKN | |
| 1–32 | 02 | SERIAL-NUM<br>The cardholder certificate serial number for secure electronic commerce. | PIC X(32) |

# Token C8 Merchant Serial Number Token

The fields in the Merchant Serial Number token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–32 | | MRCH-SERIAL-NUM-TKN | |
| 1–32 | 02 | SERIAL-NUM<br>The merchant certificate serial number for secure electronic commerce. | PIC X(32) |

# Token C9 MHI Additional Data Token—Binary Format

The fields in the binary format MHI Additional Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–202 | | MHI-ADDL-TKN | |
| 1–2 | 02 | LGTH<br>The length of the token data. | TYPE BINARY 16 |
| 3–10 | 02 | ORIG-AMT<br>The original amount that must be maintained on a preauthorization transaction if the MHI module is operating in host control mode. | TYPE BINARY 64 |
| 11–202 | 02 | USER-FLD<br>Reserved for future use. | PIC X(192) |

# Token C9 MHI Additional Data Token—ASCII Format

The fields in the ASCII format MHI Additional Data token are shown below. For descriptions of these fields, refer for the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–216 | | MHI-ADDL-DATA-TKNX | |
| 1–5 | 02 | LGTH | PIC 9(5) |
| 6–23 | 02 | ORIG-AMT | PIC 9(18) |
| 24 | 02 | USER-FLD2 | PIC X |
| 25–216 | 02 | USER-FLD | PIC X(192) |

# Token CA DUKPT Data Token

The Derived Unique Key per Transaction (DUKPT) Data token contains the key serial number set by the Standard POS Device Handler (SPDH) module. The fields in the DUKPT Data token are shown below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–40 | | DUKPT-DATA-TKN | |
| 1–20 | 02 | KEY-SERIAL-NUM<br>The key serial number. This field is filled in by the SPDH module. | PIC X(20) |
| 21–40 | 02 | USER-FLD1<br>This field is reserved for future use. | PIC X(20) |

# Token CB POS Balances Token—Binary Format

The POS Balances token is used by BASE24 on a balance inquiry when the currency of the account is different from the currency used by the POS device. This token also can be added by one of the following methods:

- By the ACI Standard POS Device Handler when the PTD Return Balances flag is set to the value Y
- By the BASE24-pos Authorization module when the CPF Return Balances flag is set to the value 1
- By the VisaNet Interface when balances are returned by the issuer

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1-40 | | POS-BAL-TKN | |
| 1-8 | 02 | ACCT-AMT-1<br>A balance specified in the account currency. | TYPE BINARY 64 SIGNED |
| 9-16 | 02 | TXN-AMT-1<br>A balance specified in the transaction currency. | TYPE BINARY 64 SIGNED |
| 17-19 | 02 | ACCT-CRNCY-CDE<br>The ISO numeric currency code of the account currency. | PIC 9(3) |
| 20-22 | 02 | TXN-CRNCY-CDE<br>The ISO numeric currency code of the transaction currency. | PIC 9(3) |
| 23-30 | 02 | CONV-RATE<br>The rate used to convert the account balances to the transaction currency.<br>This field is in ISO format (i.e., the leftmost digit specifies the number of decimal places, and positions 2-8 specify the rate). For example, 69972522 equals 9.972522. | PIC 9(8) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 31-34 | 02 | CONV-DAT<br>The date on which the currency conversion was performed, in MMDD format. | PIC 9(4) |
| 35 | 02 | CPF-RTRN-BAL<br>The value of the Return Balance Flag on the CPF. | PIC X(1) |
| 36-40 | 02 | USER-FLD1 | PIC X(5) |

# Token CB POS Balances Token—ASCII Format

The fields in the ASCII format POS Balances token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1-62 | | POS-BAL-TKNX | |
| 1-19 | 02 | ACCT-AMT-1 | PIC X(19) |
| 20-38 | 02 | TXN-AMT-1 | PIC X(19) |
| 39-41 | 02 | ACCT-CRNCY-CDE | PIC 9(3) |
| 42-44 | 02 | TXN-CRNCY-CDE | PIC 9(3) |
| 45-52 | 02 | CONV-RATE | PIC 9(8) |
| 53-56 | 02 | CONV-DAT | PIC 9(4) |
| 57 | 02 | CPF-RTRN-BAL | PIC X(1) |
| 58-62 | 02 | USER-FLD1 | PIC X(5) |

# Token CE Authentication Data Token

The Authentication Data token standardizes the transport cardholder authentication data for e-commerce transactions. The fields in the Authentication Data token are described below. These fields should be initialized to blanks.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1-202 | | AUTHN-DATA-TKN | |
| 1–2 | 02 | AUTHN-IND-FLG<br>The authentication indicator flag. Valid values are as follows:<br>01 = UCAF<br>02 = Chip Authentication Program (CAP) token | PIC X(2) |
| 3–202 | 02 | AUTHN-IND-DATA<br>The generic data. This is a variable length field that can contain up to 200 characters.<br>For a MasterCard transaction using the Secure Payment Application Universal Cardholder Authentication Field (SPA UCAF) method, this field contains the Accountholder Authentication Value (AAV). | PIC X(200) |
| 3–202 | 02 | CAP-TKN-DATA | REDEFINES AUTHN-IND-DATA |
| 3–20 | 04 | CAP-TKN<br>The CAP token from the request message. | PIC X(18) |
| 21–28 | 04 | UNPREDICT-NUM<br>The unpredictable number from the request message. If not present, then this field should be set to zeroes. | PIC X(8) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 29–32 | 04 | CRNCY-CDE | PIC 9(4) |
| | | The transaction currency code from the request message (padded to the left with a zero). If not present, then this field should be set to zeroes. | |
| 33–44 | 04 | AMT-AUTH | PIC 9(12) |
| | | The amount to be authorized from the request message. If not present, then this field should be set to zeroes. | |
| 45–46 | 04 | APSN | PIC X(2) |
| | | The application PAN sequence number from the request message. If not present, then this field should be set to spaces. | |
| 47–50 | 04 | RESP-ATC | PIC X(4) |
| | | The application transaction counter (ATC) returned from the hardware security module (HSM), following a successful CAP token validation. | |
| 51 | 04 | RSLT-CDE | PIC X |
| | | The result code returned from the HSM following a CAP token validation. | |
| 52–202 | 04 | USER-FLD-ACI | PIC X(151) |
| | | Reserved by ACI for future use. | |

# Token CF International Address Verification Service (IAVS) Data Token

The IAVS Data token supports the International Address Verification Service (IAVS). The fields in the IAVS Data token are described below. These fields should be initialized to blanks.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–9 | 02 | PSTL-CDE<br>The postal code. This field is left justified and blank filled. | PIC X(9) |
| 10–49 | 02 | ADDR<br>The address. | PIC X(40) |
| 50 | 02 | USER-FLD1<br>Reserved for future use. | PIC X |

# Token CH POS Data1 Token—Binary Format

The fields in the binary format POS Data1 token are described below. These fields should be initialized to blanks.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–36 | | POS-DATA1-TKN | |
| 1 | 02 | RESP-SRC-RSN-CDE | PIC X(1) |

A code indicating the response source or reason code. This field is set by an interchange. Valid values are as follows:

1 = Request timed out at interchange
2 = Transaction amount below issuer limit
3 = Issuer is in suppress inquiries mode
4 = Issuer is not available for processing
5 = Response provided by issuer
7 = Reversal advice provided by interchange to identify a potential duplicate transaction
8 = Reversal advice provided by interchange to identify a probable duplicate authorization
A = Third party agent

| | | | |
|---|---|---|---|
| 2 | 02 | CRD-VRFY-FLG2 | PIC X(1) |

Indicates whether the card involved in the card-read transaction has already been verified using the CVV2/CVD2. Processes that generate this token but do not use this field should initialize the value to ␣, where ␣ is a blank space. Valid values are as follows:

0 = Card verification was not performed because the transaction was denied before card verification processing started.
C = Card verification was performed and the card verification digits (CVD) were invalid. The situation was noted and the transaction processing continued.
D = Card verification was performed and the CVD was invalid. The transaction was denied and the ERR-FLG field was set to C.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

N or ␣ = Card verification was not attempted or a security device error occurred (where ␣ indicates a blank space).

O = Card verification was not performed, CVD was not on the card. Not all cards have a CVD value encoded. The card expiration date must be equal to or greater than an expiration date defined on the CPF to insure that the CVD field has been encoded. If the card expiration date is equal to or greater than the CPF date, the CVD checks are performed.

P = Card verification was not performed. Either the merchant ignored the CVD on purpose or the user falsely indicated no CVD was on the card.

R = Card verification was performed and the CVD was invalid. The situation was noted and the transaction should be referred.

U = Issuer has not certified or has not provided the encryption keys to the interchange.

Y = Card verification was performed and the CVD was valid.

3–10 02 ONLINE-LMT TYPE BINARY 64 SIGNED

The value against which the transaction amount is compared to determine whether under limit or over limit authorization is performed, using information configured in the Routing Table File (RTBL). The value is retrieved from the POS Terminal Data File (PTD) or the Authorization Selection Table File (AST).

11–14 02 RETL-CLASS-CDE PIC X(4)

Classification code of the retailer from the POS Retailer Definition File (PRDF).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 15 | 02 | EMV-CAPABLE-OUTLET | PIC X(1) |
| | | Indicates the EMV capability of the outlet from the POS Retailer Definition File (PRDF) or the Interchange. Valid values are as follows:<br>N = No, the outlet is not EMV capable<br>Y = Yes, the outlet is EMV capable | |
| 16 | 02 | RECUR-PMNT-IND | PIC X(1) |
| | | A recurring payment indicator. The only valid value is R, which represents a recurring payment. | |
| 16 | 02 | PMNT-IND | REDEFINES RECUR-PMNT-IND |
| | | Indicates the type of payment associated with a transaction. Valid values are as follows:<br>I = Installment payment<br>R = Recurring payment | |
| 17–18 | 02 | NUM-INSTL | PIC X(2) |
| | | The total number of installment payments required for the full purchase amount. | |
| 19–20 | 02 | NUM-MM-GRATUITY | PIC X(2) |
| | | The grace period from the transaction date when the acquirer will send the first installment payment to the issuer. This value can be specified in days, weeks, or months, as defined by the INSTL-GRATUITY-PRD field. | |
| 21–23 | 02 | PMNT-PLAN | PIC X(3) |
| | | The type of payment plan program that was selected at the point of purchase. This field may be two to three characters in length. This field is left-justified and space-filled. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 24 | 02 | TERM-OUTPUT-CAP-IND | PIC X(1) |

This field indicates the ability of the terminal to print or display messages. Valid values are:

0 = Unknown
1 = None
2 = Print
3 = Display
4 = Print and display

| | | | |
|---|---|---|---|
| 25 | 02 | CRDHLDR-AUTHN-CAP-IND | PIC X(1) |

This field indicates the primary means of verifying the cardholder at the terminal. Valid values are:

0 = No electronic authentication
1 = PIN
2 = Electronic signature analysis
3 = Biometrics
4 = Biographics
5 = Electronic authentication inoperative
6 = Other
7 = Reserved for future use
8 = Reserved for future use
9 = Authentication value

| | | | |
|---|---|---|---|
| 26 | 02 | PARTIAL-AUTH-OPT | PIC X(1) |

This field indicates the support type provided for partial authorizations. Valid values are:

␣ = No information available (where ␣ equals a blank space)
R = Transaction cannot be authorized for lesser amount than was requested, and the requested amount must not impact cardholder accumulators or balances.
P = Transaction may be authorized for a lesser amount
G = Transaction may be authorized for a greater amount
D = Not yet determined whether transaction may be partially authorized

Any other value indicates that the transaction may not be partially authorized.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 27–28 | 02 | INSTL-PLAN-TYP | PIC X(2) |

The type of credit associated with the installment payment. Valid vallues are:

20 = Issuer-financed
21 = Merchant-financed

| | | | |
|---|---|---|---|
| 29 | 02 | INSTL-GRATUITY-PRD | PIC X(1) |

The unit of the grace period before the first installment payment, as specified in the NUM-MM-GRATUITY field. Any values other than the valid values listed below indicate that NUM-MM-GRATUITY contains a number of months:

0 = Months
1 = Weeks
2 = Days

| | | | |
|---|---|---|---|
| 30 | 02 | RVSL-RSN-IND | PIC X(1) |

An indicator specifying the reason for a reversal. Valid values are:

b = No information available (where b equals a blank space)
0 = Unknown or unspecified
1 = High fraud risk
2 = Recalculated currency conversion fees
3 = Automatic fuel dispenser
4 = Card authentication method (CAM) failure

| | | | |
|---|---|---|---|
| 31 | 02 | FAILED-CVM-ALWD | PIC X(1) |

An indicator specifying whether an EMV transaction can be approved if cardholder verification failed.

Valid values are:

b = No information available (where b equals a blank space)
0 = Standard authorization processing applies
1 = Transaction may be approved if cardholder verification is not successful

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 32 | 02 | DUP-CHK-REQ | PIC X(1) |

An indicator specifying whether duplicate checking is required for this transaction. Valid values are:

- ~~b~~ = No information available (where ~~b~~ equals a blank space)
- 0 = Duplicate checking is required
- 1 = Duplicate checking is not required

| | | | |
|---|---|---|---|
| 33 | 02 | AUTH-MSG-IND | PIC X(1) |

An indicator specifying the type of authorization message.

Valid values are:

- ~~b~~ = No information available (where ~~b~~ equals a blank space)
- 0 = Normal authorization
- 1 = Final authorization

| | | | |
|---|---|---|---|
| 34 | 02 | TERM-TYP | PIC X(1) |

An indicator specifying additional information about the terminal used to initiate the transaction.

Valid values are:

- ~~b~~ = No additional information available (where ~~b~~ equals a blank space)
- 9 = Mobile acceptance solution. Customer is using a mobile telecommunications device that is not solely dedicated to POS functions and has the ability to accept a card payment.

| | | | |
|---|---|---|---|
| 35–36 | 02 | USER-FLD1 | PIC X(2) |

Reserved for future use.

# Token CH POS Data1 Token—ASCII Format

The fields in the ASCII format POS Data1 token are described below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–40 | | POS-DATA1-TKN | |
| 1 | 02 | RESP-SRC-RSN-CDE | PIC X |
| 2 | 02 | CRD-VRFY-FLG2 | PIC X |
| 3–14 | 02 | ONLINE-LMT | PIC X(12) |
| 15–18 | 02 | RETL-CLASS-CDE | PIC X(4) |
| 19 | 02 | EMV-CAPABLE-OUTLET | PIC X |
| 20 | 02 | RECUR-PMNT-IND | PIC X |
| 21–22 | 02 | NUM-INSTL | PIC X(2) |
| 23–24 | 02 | NUM-MM-GRATUITY | PIC X(2) |
| 25–27 | 02 | PMNT-PLAN | PIC X(3) |
| 28 | 02 | TERM-OUTPUT-CAP-IND | PIC X |
| 29 | 02 | CRDHLDR-AUTHN-CAP-IND | PIC X |
| 30 | 02 | PARTIAL-AUTH-OPT | PIC X |
| 31-32 | 02 | INSTL-PLAN-TYP | PIC X(2) |
| 33 | 02 | INSTL-GRATUITY-PRD | PIC X |
| 34 | 02 | RVSL-RSN-IND | PIC X |
| 35 | 02 | FAILED-CVM-ALWD | PIC X |
| 36 | 02 | DUP-CHK-REQ | PIC X |
| 37 | 02 | AUTH-MSG-IND | PIC X |
| 38 | 02 | TERM-TYP | PIC X |
| 39-40 | 02 | USER-FLD1 | PIC X(2) |

# Token CI POS Merchant Token

The fields in the POS Merchant token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–70 | | POS-MRCH-TKN | |
| 1 | 02 | E-COMM-GOODS-IND | PIC X(1) |
| | | A code indicating the type of merchandise being sold. This code is passed from the acquiring terminal and can be specific to a transaction or merchant. Valid values are as follows:<br>D = Digital<br>P = Physical goods<br>S = Services | |
| 2 | 02 | EXISTING-DEBT-IND | PIC X(1) |
| | | A code indicating whether a credit card is used to pay for an existing debt. If a credit card was not used to pay for an existing debt, this field is blank. This field is passed from the acquiring terminal. The only valid value is 9 (Payment on existing debt). | |
| 3 | 02 | DEFERRED-BILLING-IND | PIC X(1) |
| | | A code indicating whether a purchase is made with the payment deferred until a later date. This code is passed from the acquiring terminal. Valid values are as follows:<br>0 = Deferred billing is not provided.<br>1 = Deferred billing is used at the point of service. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 4 | 02 | RELN-PARTICIPANT-IND | PIC X(1) |
| | | A code indicating whether the merchant or acquirer has a special relationship with the cardholder. This code is passed from the acquiring terminal. Valid values are as follows:<br>0 = Not a relationship participant or relationship not provided.<br>1 = Relationship participant. | |
| 5–8 | 02 | DPC-NUM | PIC X(4) |
| | | The DPC number of the accepting entity to which transactions are routed for POS pass-through processing. This field is set from the DESTINATION DPC field on BASE24-pos Terminal Data file (PTD) screen 6. Valid values are 0000–9999. | |
| 9–24 | 02 | PINPAD-ID | PIC X(16) |
| | | The PIN pad identifier passed from the acquiring terminal. This value is supported in a pass through mode only. | |
| 25–40 | 02 | ACQ-TERM-ID | PIC X(16) |
| | | The terminal ID passed in the message from the acquiring terminal. | |
| 41 | 02 | RCNCL-ENT | PIC X(1) |
| | | A code identifying the entity used to reconcile for POS pass-through processing. This field is set from the RECONCILIATION ENTITY field on BASE24-pos Terminal Data file (PTD) screen 6. Valid values are as follows:<br>0 = No totals<br>1 = Totals by DPC<br>2 = Totals by retailer ID<br>3 = Totals by retailer group<br>4 = Totals by terminal ID<br>5 = Totals by clerk ID | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 42–47 | 02 | ACQ-TERM-POST-DAT | PIC X(6) |
| | | The transaction posting date passed from the acquiring terminal. | |
| 48 | 02 | PRE-AUTH-CHRGBK | PIC X(1) |
| | | A flag indicating whether online chargebacks are generated for preauthorization hold completion transactions where the preauthorization hold has expired, is not found, or the completion amount is greater than the originally authorized hold amount. This field is set from the PREAUTH CHARGEBACKS field on BASE24-pos Terminal Data file (PTD) screen 6. Valid values are as follows:<br>Y = Yes, chargebacks are generated.<br>N = No, chargebacks are not generated (default). | |
| 49 | 02 | ENHANCED-PRE-AUTH | PIC X(1) |
| | | A flag indicating whether enhanced preauthorization processing is used for the acquiring terminal. Valid values are as follows:<br>Y = Yes, use enhanced preauthorization processing.<br>N = No, do not use enhanced preauthorization processing (default). | |
| 50–52 | 02 | ADNL-RESP-CDE | PIC X(3) |
| | | Used to send a response code to the device that is not an auth response code. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 53 | 02 | MC-ELEC-ACCPT-IND | PIC X(1) |

Indicates whether or not the acquirer participates in MasterCard Electronic. Valid values are as follows:

C = MasterCard only participant (considered not a MasterCard Electronic transaction).
E = Acquirer and their merchant both participate in MasterCard Electronic (considered a MasterCard electronic transaction).
M = Acquirer participates, but the merchant that processed this transaction does not participate in MasterCard Electronic (considered not a MasterCard Electronic transaction).
U = Unidentified acquirer. It is unknown if the acquirer is a MasterCard Electronic participant.

| | | | |
|---|---|---|---|
| 54–70 | 02 | USER-FLD1 | PIC X(17) |

Reserved for future use.

# Token CJ Pre-Pay Merchant Token—Binary Format

The fields in the Pre-Pay Merchant token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–48 | | PRE-PAY-MRCH-TKN | |
| 1–4 | 02 | REFUND-TIMR<br>The number of seconds before a refund expires. This value is populated by the Transaction Context Manager process using the telco's Mobile Operator File (MOF) record. Valid values are 0 to 99999, the default is 600. | TYPE BINARY 32 |
| 5–24 | 02 | PROD-NAM<br>The product name of the telco's pre-pay scheme. | PIC X(20) |
| 25 | 02 | END-OF-DAY-FLG<br>The end of day flag. This field is populated by BASE24-eps. | PIC X(1) |
| 26–27 | 02 | SOL-PROV-ID<br>The solution provider ID. This field is populated by the Transaction Context Manager process. | PIC X(2) |
| 28–30 | 02 | TELCO-MCC<br>The specific Merchant Category Code for the telco. This field is populated by BASE24-eps. | PIC X(3) |
| 31–32 | 02 | REL-IND<br>The relationship indicator. This field is populated by BASE24-eps. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 33–47 | 02 | PURCH-REF<br>The reference number of a purchase transaction. This field is populated and used by the Telco Interface on a refund reversal when the reversal ID is stored in PPTU-TKN.REF field and the refund ID is stored in PPTU-TKN.ORIG-REF field. | PIC X(15) |
| 48 | 02 | USER-FLD-ACI<br>Reserved for future use. | PIC X(1) |

# Token CJ Pre-Pay Merchant Token—ASCII Format

The fields in the ASCII format Pre-Pay Merchant token are described below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–54 | | PRE-PAY-MRCH-TKNX | |
| 1–10 | 02 | REFUND-TIMR | PIC X(10) |
| 11–30 | 02 | PROD-NAM | PIC X(20) |
| 31 | 02 | END-OF-DAY-FLG | PIC X(1) |
| 32–33 | 02 | SOL-PROV-ID | PIC X(2) |
| 34–36 | 02 | TELCO-MCC | PIC X(3) |
| 37–38 | 02 | REL-IND | PIC X(2) |
| 39–53 | 02 | PURCH-REF | PIC X(15) |
| 54 | 02 | USER-FLD-ACI | PIC X(1) |

# Token CK Industry Data Token

The Industry Data token contains information associated with lodging and vehicle rental. The fields in the Industry Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–172 | | INDUSTRY-DATA-TKN | |
| 1–2 | 02 | INDUSTRY-TYP | PIC X(2) |
| | | A code that identifies the type of industry data. Valid values are as follows:<br>LG = Lodging<br>VR = Vehicle rental | |
| 3–172 | 02 | INDUSTRY-DATA | PIC X(170) |
| 3–172 | 02 | LODGING | REDEFINES INDUSTRY-DATA |
| | | The following fields are used for lodging data when the INDUSTRY-TYP field is equal to LG. | |
| 3–8 | 04 | ARRIVAL-DAT | PIC X(6) |
| | | The date (YYMMDD) the customer checked in. For a no-show or advanced lodging transaction, this is the scheduled arrival date. | |
| 9–14 | 04 | DEPART-DAT | PIC X(6) |
| | | The date (YYMMDD) the customer checked out. | |
| 15–18 | 04 | TTL-ROOM-NIGHTS | PIC X(4) |
| | | The total number of room nights during the lodging stay. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 19–30 | 04 | ROOM-RATE<br>The daily room charges exclusive of taxes and fees. Two decimal places are implied. | PIC X(12) |
| 31–42 | 04 | ROOM-TAX<br>The daily room tax. Two decimal places are implied. | PIC X(12) |
| 43–54 | 04 | PHN-CHRGS<br>The total amount of charges for all phone calls. Two decimal places are implied. | PIC X(12) |
| 55–66 | 04 | LAUNDRY-CHRGS<br>The total amount of laundry and dry cleaning charges. Two decimal places are implied. | PIC X(12) |
| 67–78 | 04 | GIFT-SHOP-CHRGS<br>The total amount of gift shop and specialty shop charges. Two decimal places are implied. | PIC X(12) |
| 79–90 | 04 | BAR-CHRGS<br>The total amount of bar and in-room mini-bar charges. Two decimal places are implied. | PIC X(12) |
| 91–102 | 04 | OTHER-CHRGS<br>The total amount of other charges associated with the lodging stay. Two decimal places are implied. | PIC X(12) |
| 103–114 | 04 | TTL-TAX-AMT<br>The total amount of sales tax or value-added tax on the total purchase. Two decimal places are implied. | PIC X(12) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 115–129 | 04 | PROPERTY-PHN-NUM | PIC X(15) |
| | | Identifies the specific lodging property location by its local phone number. | |
| 130–144 | 04 | CUST-SVC-PHN-NUM | PIC X(15) |
| | | The phone number used to resolve cardholder questions and disputes. | |
| 145–154 | 04 | FOLIO-NUM | PIC X(10) |
| | | The merchant's internal invoice or billing ID reference number. | |
| 155 | 04 | FIRE-SAFETY-ACT-IND | PIC X(1) |
| | | A code that identifies whether the facility is in compliance with the Hotel and Motel Fire Safety Act of 1990 (PL101-391), or similar legislation. Valid values are as follows:<br>Y = Yes, the facility is in compliance.<br>N = No, the facility is not in compliance. | |
| 156 | 04 | NO-SHOW-IND | PIC X(1) |
| | | A code indicating whether the individual showed up after making a reservation for lodging. Valid values are as follows:<br>0 = Not applicable.<br>1 = No show. Transaction amount is due. | |
| 157–172 | 04 | USER-FLD1 | PIC X(16) |
| 03–172 | 02 | VEHICLE | REDEFINES INDUSTRY-DATA |
| | | The following fields are used for vehicle rental data when the INDUSTRY-TYP field is equal to VR.. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–31 | 04 | RENTER-NAM<br>The name of the individual making the vehicle rental agreement. | PIC X(29) |
| 32–35 | 04 | RENTAL-CLAS-ID<br>The classification of the vehicle rented, such as midsize or luxury. | PIC X(4) |
| 36–41 | 04 | RENTAL-DAT<br>The date (YYMMDD) the customer picked up the vehicle from the rental agency. | PIC X(6) |
| 42–59 | 04 | RENTAL-CITY<br>The city where the customer picked up the vehicle. | PIC X(18) |
| 60–62 | 04 | RENTAL-ST<br>The state or province where the customer picked up the vehicle. This field must contain a valid U.S. status code if the rental country is USA. | PIC X(3) |
| 63–65 | 04 | RENTAL-CNTRY<br>The country where the customer picked up the vehicle. This field must contain a valid alphabetic ISO country code. | PIC X(3) |
| 66–71 | 04 | RTRN-DAT<br>The date (YYMMDD) the customer returned the vehicle. | PIC X(6) |
| 72–89 | 04 | RTRN-CITY<br>The city where the customer returned the vehicle. | PIC X(18) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 90–92 | 04 | RTRN-ST<br>The state or province where the customer returned the vehicle. This field must contain a valid U.S. status code if the rental country is USA. | PIC X(3) |
| 93–95 | 04 | RTRN-CNTRY<br>The country where the customer returned the vehicle. This field must contain a valid alphabetic ISO country code. | PIC X(3) |
| 96–105 | 04 | RTRN-LOC-ID<br>The code, address, phone number, or other identifier used to identify the location where the customer returned the vehicle. | PIC X(10) |
| 106–109 | 04 | DAYS-RENTED<br>The number of days the vehicle was rented. | PIC X(4) |
| 110–121 | 04 | DLY-RENTAL-RATE<br>The daily rental rate, exclusive of taxes and fees. Two decimal places are implied. | PIC X(12) |
| 122–133 | 04 | EXTRA-CHRGS<br>The total amount of extra charges associated with the vehicle rental. Two decimal places are implied. | PIC X(12) |
| 134–145 | 04 | TTL-TAX-AMT<br>The total amount of sales tax or value-added tax on the total purchase. Two decimal places are implied. | PIC X(12) |
| 146–160 | 04 | CUST-SVC-PHN-NUM<br>The phone number used to resolve cardholder questions and disputes. | PIC X(15) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 161–169 | 04 | AGREEMENT-NUM<br>The invoice number of the original rental agreement. | PIC X(9) |
| 170 | 40 | TAX-EXEMPT-IND<br>A code indicating whether the goods or services were tax exempt. Valid values are as follows:<br>0 = Not applicable<br>1 = Tax exempt | PIC X(1) |
| 171 | 04 | NO-SHOW-IND<br>A code indicating whether the individual showed up after making a reservation for the vehicle. Valid values are as follows:<br>0 = Not applicable.<br>1 = No show. Transaction amount is due. | PIC X(1) |
| 172 | 04 | USER-FLD2 | PIC X(1) |

# Token CP Healthcare Token

The Healthcare token contains information associated with transit autosubstantiation transactions for healthcare and healthcare eligibility inquiry transactions. The fields in the Healthcare token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–424 | | HEALTHCARE-TKN | |
| 1–120 | 02 | ADDL-AMT<br>A code indicating the account balance information for inquiry transactions to support verification of healthcare elibitility and auto-substantiation transactions for healthcare and transit. | |
| 1–20 | 04 | BAL1<br>This field contains the first account balance. | |
| 1–2 | 06 | ACCT-TYP<br>The account type for the first balance. A value of 00 is used to indicate auto-substantiation and healthcare eligibility transactions. | PIC X(2) |
| 3–4 | 06 | AMT-TYP<br>The type of payment amount in the first balance. Valid values are as follows:<br>3S = Amount co-payment<br>4S = Amount healthcare<br>4T = Amount transit | PIC X(2) |
| 5–7 | 06 | CRNCY-CDE<br>The standard three-digit ISO numeric currency code for the first balance. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 8 | 06 | AMT-SIGN<br>A code indicating whether the amount is positive or negative in the first balance. Valid values are as follows:<br>C = Credit, positive balance<br>D = Debit, negative balance | PIC X(1) |
| 9–20 | 06 | AMT<br>This field contains the amount specified by the amount type in the first balance. | PIC X(12) |
| 21–40 | 04 | BAL2<br>This field contains the second account balance. | |
| 21–22 | 06 | ACCT-TYP<br>The account type of the second balance. A value of 00 is used to indicate auto-substantiation and healthcare eligibility transactions. | PIC X(2) |
| 23–24 | 06 | AMT-TYP<br>The type of payment amount in the second balance. Valid values are as follows:<br>3S = Amount co-payment<br>4S = Amount healthcare<br>4T = Amount transit | PIC X(2) |
| 25–27 | 06 | CRNCY-CDE<br>The standard three-digit ISO numeric currency code for the second balance. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 28 | 04 | AMT-SIGN<br>A code indicating whether the amount is positive or negative in the second balance. Valid values are as follows:<br>C = Credit, positive balance<br>D = Debit, negative balance | PIC X(1) |
| 29–40 | 04 | AMT<br>This field contains the amount specified by the amount type in the second balance. | PIC X(12) |
| 41–60 | 04 | BAL3<br>This field contains the third balance. | |
| 41–42 | 06 | ACCT-TYP<br>The account type of the third balance. A value of 00 is used to indicate auto-substantiation and healthcare eligibility transactions. | PIC X(2) |
| 43–44 | 06 | AMT-TYP<br>The type of payment amount in the third balance. Valid values are as follows:<br>3S = Amount co-payment<br>4S = Amount healthcare<br>4T = Amount transit | PIC X(2) |
| 45–47 | 06 | CRNCY-CDE<br>The standard three-digit ISO numeric currency code for the third balance. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 48 | 04 | AMT-SIGN<br>A code indicating whether the amount is positive or negative in the third balance. Valid values are as follows:<br>C = Credit, positive balance<br>D = Debit, negative balance | PIC X(1) |
| 49–60 | 04 | AMT<br>This field contains the amount specified by the amount type in the third balance. | PIC X(12) |
| 61–80 | 04 | BAL4<br>This field contains the fourth account balance. | |
| 61–62 | 06 | ACCT-TYP<br>The account type of the fourth balance. A value of 00 is used to indicate auto-substantiation and healthcare eligibility transactions. | PIC X(2) |
| 63–64 | 06 | AMT-TYP<br>The type of payment amount in the fourth balance. Valid values are as follows:<br>3S = Amount co-payment<br>4S = Amount healthcare<br>4T = Amount transit | PIC X(2) |
| 65–67 | 06 | CRNCY-CDE<br>The standard three-digit ISO numeric currency code for the fourth balance. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 68 | 04 | AMT-SIGN<br>A code indicating whether the amount is positive or negative in the fourth balance. Valid values are as follows:<br>C = Credit, positive balance<br>D = Debit, negative balance | PIC X(1) |
| 69–80 | 04 | AMT<br>This field contains the amount specified by the amount type in the fourth balance. | PIC X(12) |
| 81–100 | 04 | BAL5<br>This field contains the fifth account balance. | |
| 81–82 | 06 | ACCT-TYP<br>The account type of the fifth balance. A value of 00 is used to indicate auto-substantiation and healthcare eligibility transactions. | PIC X(2) |
| 83–84 | 06 | AMT-TYP<br>The type of payment amount in the fifth balance. Valid values are as follows:<br>3S = Amount co-payment<br>4S = Amount healthcare<br>4T = Amount transit | PIC X(2) |
| 85–87 | 06 | CRNCY-CDE<br>The standard three-digit ISO numeric currency code for the fifth balance. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 88 | 04 | AMT-SIGN<br>A code indicating whether the amount is positive or negative in the fifth balance. Valid values are as follows:<br>C = Credit, positive balance<br>D = Debit, negative balance | PIC X(1) |
| 89–100 | 04 | AMT<br>This field contains the amount specified by the amount type in the fifth balance. | PIC X(12) |
| 101–120 | 04 | BAL6<br>This field contains the sixth account balance. | |
| 101–102 | 06 | ACCT-TYP<br>The account type of the sixth balance. A value of 00 is used to indicate auto-substantiation and healthcare eligibility transactions. | PIC X(2) |
| 103–104 | 06 | AMT-TYP<br>The type of payment amount in the sixth balance. Valid values are as follows:<br>3S = Amount co-payment<br>4S = Amount healthcare<br>4T = Amount transit | PIC X(2) |
| 105–107 | 06 | CRNCY-CDE<br>The standard three-digit ISO numeric currency code for the sixth balance. | PIC X(3) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 108 | 04 | AMT-SIGN<br>A code indicating whether the amount is positive or negative in the sixth balance. Valid values are as follows:<br>C = Credit, positive balance<br>D = Debit, negative balance | PIC X(1) |
| 109–120 | 04 | AMT<br>This field contains the amount specified by the amount type in the sixth balance. | PIC X(12) |
| 121–224 | 02 | TXN-SPCF-DATA<br>This field contains multiple datasets and each dataset contains multiple sub-elements. The format of this field is based on the International Organization for Standardization (ISO) Tag, Length, Value (TLV) format, defined as follows:<br>The Tag field contains a one- to two-byte hexadecimal code identifying the contents of the value field.<br>The Length field defines the length of the Value field's positions.<br>The Value field is a field of variable length containing the requested data. | |
| 121 | 04 | DATASET-ID<br>The dataset identification code. | PIC X(1) |
| 122–123 | 04 | DATASET-LGTH<br>The length of the dataset. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 124–224 | 04 | INFO | PIC X(101) |
| 124–224 | 04 | HEALTHCARE | REDEFINES INFO |
| 124–150 | 06 | TLV-DATA | |
| 124–134 | 08 | HLTHCR-PROV-ID<br>The healthcare provider identification code. This sub-element contains the medical license number of the provider. | |
| 124 | 10 | TAG | PIC X(1) |
| 125 | 10 | LGTH | PIC X(1) |
| 126–134 | 10 | TAG-DATA | PIC X(9) |
| 135–138 | 08 | SRV-TYP-CDE<br>This sub-element contains the standard code for healthcare treatment defined by the Health Insurance Portability and Accountability Act (HIPAA). | |
| 135 | 10 | TAG | PIC X(1) |
| 136 | 10 | LGTH | PIC X(1) |
| 137–138 | 10 | TAG-DATA | PIC X(2) |
| 139–146 | 08 | PAYER-ID<br>This sub-element contains the identification code of the health insurance carrier/payer. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 139 | 10 | TAG | PIC X(1) |
| 140 | 10 | LGTH | PIC X(1) |
| 141–146 | 10 | TAG-DATA | PIC X(6) |
| 147–150 | 08 | REASON-CDE<br>This sub-element contains the HIPAA-defined codes for approvals and declines of eligibility transactions. | |
| 147 | 10 | TAG | PIC X(1) |
| 148 | 10 | LGTH | PIC X(1) |
| 149–150 | 10 | TAG-DATA | PIC X(2) |
| 151–224 | 06 | RESERVED | PIC X(74) |
| 225–424 | 02 | RESERVED | PIC X(200) |

# Token CQ Reward Program Token

The fields of the Reward Program token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–46 | | REWARD-PGM-TKN | |
| 1–6 | 02 | PRGRM-ID<br>This field contains the Reward Program Identifier to support the Reward Program Identification Number (RPIN). | PIC X(6) |
| 7–46 | 02 | RESERVED<br>This field is reserved for future use. | PIC X(40) |

# Token CR POS Split Transaction Routing Token

The POS Split Transaction Routing token carries POS data allowing BASE24 to route multiple transaction requests related to a single cardholder request. This token also allows BASE24 to identify and merge the multiple responses received into a single response bound for the cardholder. For transactions involving split transaction routing, this token is set by the Transaction Context Manager (TCM) to store the values from the original POS Standard Internal Message (PSTM).

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–60 | | POS-SPLIT-TXN-RTE-TKN | |
| 1–16 | 02 | AST-RTN-PRO-NAME [SYM-NAME]<br>The symbolic name of the primary authorizer for this transaction, as defined in the Authorization Selection Table (AST). | PIC X(16) |
| 17–32 | 02 | ROUTER1-NAME [SYM-NAME]<br>The symbolic process name of the first Router Module to handle this transaction. | PIC X(16) |
| 33–48 | 02 | ROUTER2-NAME [SYM-NAME]<br>The symbolic process name of the second Router Module to handle this transaction. | PIC X(16) |
| 49–60 | 02 | USER-FLD-ACI<br>This field is reserved by ACI for future use. | PIC X(12) |

# Token CS Enhanced Reversal Routing Token

The fields in the Enhanced Reversal Routing token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–16 | | ENHNC-RVSL-RTE-TKN | |
| 1–16 | 02 | AUTH-DEST<br>The symbolic name of the interchange interface process that authorized the original transaction and to which the reversal will be sent. | PIC X(16) |

# Token CT Transaction Specific Data Token Using Redefines

The Transaction Specific Data token may be used in conjunction with the definitions that follow in this section to explicitly reference the datasets and fields that may be included in the token. The fields in the Transaction Specific Data token are described below.

The Transaction Specific Data token can also use multiple datasets. Refer to the separate description later in this manual for more information.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–254 | | TXN-SPCF-DATA-TKN | |
| 1–2 | 02 | DATASET-ID | PIC X(2) |
| | | Contains the value used to identify the data that follows. Valid values are as follows:<br>56 = Dial term data<br>57 = Related transaction data<br>59 = Promotion data<br>5D = Installment Payment Data<br>5F = Money Transfer Data<br>65 = MasterCard Member-Defined Data<br>71 = Free form text area<br>~5 = Extended Dataset Version 5 | |
| 3–254 | 02 | INFO | PIC X(252) |
| 3–254 | 02 | DIAL-TERM | REDEFINES INFO |
| | | Future changes to the DIAL-TERM redefine of INFO will be maintained in the DIAL-TERM-DATA definition. Refer to "Token CT Transaction Specific Data Token Using Extended Datasets" topic later in this manual. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3 | 04 | PHN-SVC-CDE<br>This sub-element contains the phone service code. | PIC X(1) |
| 4–254 | 04 | USER-FLD-DIAL-TERM<br>Reserved by ACI for future use. | PIC X(251) |
| 3–254 | 02 | RELATED-TXN-DATA<br>Future changes to the RELATED-TXN-DATA redefine of INFO will be maintained in the RELATED-TXN-DATA definition. Refer to "Token CT Transaction Specific Data Token Using Extended Datasets" topic later in this manual. | REDEFINES INFO |
| 3–4 | 04 | BUS-APPL-ID<br>This subelement contains the Business Application Identifier used for Visa Original Credit transactions. This subelement is also used to map to/from the Payment Transaction Type Identifier for Banknet and MasterCard payment transactions. Valid values are as follows:<br>AA = Account-to-account<br>BB = Business-to-business<br>BI = Money transfer, bank initiated<br>BP = Bill payment<br>BC = Business-to-consumer<br>C1 = Person-to-person<br>C2 = Rebate<br>C3 = Load value<br>C7 = MoneySend person-to-person<br>C8 = Mobile-initiated payment<br>C9 = Card activation<br>CP = Card bill payment<br>FD = Funds disbursement (general)<br>GD = Government disbursement<br>GP = Gambling payment (other than online gambling)<br>I1 = Utility payments<br>I2 = Government services<br>I3 = Mobile phone top-ups | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | I4 = Coupon booklet payments<br>MD = Merchant disbursement<br>MI = Money transfer, merchant initiated<br>OG = Online gambling payout<br>PA = Payment transaction<br>PD = Payroll/pension disbursement<br>PG = Payment to government<br>PS = Payment for goods and services (general)<br>VC = Visa commerce | |
| 5 | 04 | SRC-OF-FUND<br>This subelement contains the source of funds. | PIC X(1) |
| 6–254 | 04 | RELATED-TXN-DATA-USERFLD<br>Reserved by ACI for future use. | PIC X(249) |
| 3–254 | 02 | PRMTN<br>Future changes to the PRMTN-DATA redefine of INFO will be maintained in the PRMTN-DATA-DATA definition. Refer to "Token CT Transaction Specific Data Token Using Extended Datasets" topic later in this manual. | REDEFINES INFO |
| 3–4 | 04 | TYP<br>This subelement contains the promotion type. | PIC X(2) |
| 5–29 | 04 | CDE<br>This subelement contains the promotion code. | PIC X(25) |
| 30–54 | 04 | DESCR<br>This subelement contains the promotion description. | PIC X(25) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 55–254 | 04 | USER-FLD-PRMTN<br>Reserved by ACI for future use. | PIC X(200) |
| 3–254 | 02 | FREE-FORM-TXT<br>Future changes to the FREE-FORM-TXT redefine of INFO will be maintained in the FREE-FORM-TXT definition. Refer to “Token CT Transaction Specific Data Token Using Extended Datasets” topic later in this manual. | REDEFINES INFO |
| 3–101 | 04 | DESCR<br>Contains either member-to-member data or Gateway transactions - MasterCard data. | PIC X(99) |
| 102–151 | 04 | ORIG-CR-APPL-DATA | PIC X(50) |
| 152–254 | 04 | USER-FLD-FREE-FORM | PIC X(103) |

# Token CT Transaction Specific Data Token Using Extended Datasets

The Transaction Specific Data token using extended datasets may be used in conjunction with the definitions that follow in this section to explicitly reference the datasets and fields that can be included in the token. Multiple occurrences of a single dataset are possible, as well as multiple types of datasets. The datasets present in the token may be in any order and may include duplicates. This format is indicated by a value with a ~ in the first byte of the DATASET-ID field. The second byte of the DATASET-ID field will be incremented whenever the definitions are modified. The fields in the Transaction Specific Data token using multiple datasets are described below. The maximum size for this token is 254 characters.

The Transaction Specific Data token can also use redefines. Refer to the separate description documented previously in this manual for more information.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–254 | | TXN-SPCF-DATA-TKN | |
| 1–2 | 02 | DATASET-ID | PIC X(2) |

Contains the value used to identify the data that follows. Valid values are as follows:

- 56 = Dial term data
- 57 = Related transaction data
- 59 = Promotion data
- 5D = Installment payment data
- 5F = Money transfer data
- 65 = MasterCard member-defined data
- 69 = Multiple payment forms data
- 6C = Travel tag data
- 6E = Loan details data
- 71 = Free form text data
- ~8 = Extended dataset Version 8
  The token uses the DEFINITIONS that follow in this section to explicitly reference the datasets and fields that can be included in the token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3–254 | 02 | INFO | PIC X(252) |

The following definitions are used to further define the data in the Transaction Specific Data Token. There may be 1–*n* occurrences of each dataset. They may occur in any order.

---

DIAL-TERM-DATA DEFINITION

This definition is used to explicitly reference the Dial Terminal Data fields in the Transaction Specific Data Token. The length of this definition is 4 characters.

02 DATASET-ID PIC (2)

The DATASET-ID for the DIAL-TERM-DATA is 56.

02 PHN-SVC-CDE PIC X(1)

Contains the phone service code. Valid values are as follows:

A = Leased line service
B = WATS
C = Local service
D = Digital Radio Network (DRN/LATA)

02 USER-FLD-ACI PIC X(1)

Reserved by ACI for future use.

---

RELATED-TXN-DATA DEFINITION

This definition is used to explicitly reference the Related Transaction Data fields in the Transaction Specific Data Token. The length of this definition is 6 characters.

02 DATASET-ID PIC (2)

The DATASET-ID for the RELATED-TXN-DATA is 57.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | BUS-APPL-ID | PIC X(2) |

This field is used to map to/from the following external message fields:

- Business Application Identifier used for Visa Original Credit transactions

- Payment Transaction Type Identifier for Banknet and MasterCard payment transactions

- Application Identifier used for Shazam funds transfer Debit and Credit transactions

Valid values are as follows:

AA = Account-to-account
BB = Business-to-business
BI = Money transfer, bank initiated
BP = Bill payment
BC = Business-to-consumer
C1 = Person-to-person
C2 = Rebate
C3 = Load value
C7 = MoneySend person-to-person
C8 = Mobile-initiated payment
C9 = Card activation
CP = Card bill payment
FD = Funds disbursement (general)
GD = Government disbursement
GP = Gambling payment (other than online gambling)
I1 = Utility payments
I2 = Government services
I3 = Mobile phone top-ups
I4 = Coupon booklet payments
MD = Merchant disbursement
MI = Money transfer, merchant initiated
OG = Online gambling payout
PA = Payment transaction
PD = Payroll/pension disbursement
PG = Payment to government
PS = Payment for goods and services (general)
VC = Visa commerce

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | SRC-OF-FUND | PIC X(1) |
| | | Contains the source of funds. Valid values are as follows:<br>1 = Cash<br>2 = Check<br>3 = Card | |
| | 02 | USER-FLD-ACI | PIC X(1) |
| | | Reserved by ACI for future use. | |

| | | | |
|---|---|---|---|
| | | PRMTN-DATA | DEFINITION |
| | | This definition is used to explicitly reference the Related Transaction Data fields in the Transaction Specific Data Token. The length of this definition is 55 characters. | |
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for the PRMTN-DATA is 59. | |
| | 02 | TYP | PIC X(2) |
| | | This field contains the promotion type. | |
| | 02 | CDE | PIC X(25) |
| | | This field contains the promotion code. | |
| | 02 | DESCR | PIC X(25) |
| | | This field contains the promotion description. | |

| | | | |
|---|---|---|---|
| | | INSTL-PMNT-DATA | DEFINITION |
| | | This definition is used to explicitly reference the Installment Payment data fields in the Transaction Specific Data Token. The length of this definition is 36 characters. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for the INSTL-PMNT-DATA is 5D. | |
| | 02 | TTL-AMT | PIC X(12) |
| | | This field contains the total amount of all payments. This field is right-justified ad zero-filled. | |
| | 02 | CRNCY-CDE | PIC X(3) |
| | | This field contains the currency code of the payment submitted. | |
| | 02 | NUM-INSTL | PIC X(3) |
| | | This field contains the number of installment payments that will occur. This field is right-justified and zero-filled. | |
| | 02 | AMT-EACH-INSTL | PIC X(12) |
| | | This field contains the amount of each istallment payment. This field is right-justified and zero-filled. | |
| | 02 | INSTL-PMNT-NUM | PIC X(3) |
| | | This field contains the installment payment number. This field is right-justified and zero-filled. | |
| | 02 | FREQ-INSTL | PIC X(1) |
| | | This field contains the frequency of the installment payments. Valid values are as follows:<br>B = Bi-weekly<br>M = Monthly<br>W = Weekly | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | MONEY-XFER-DATA | DEFINITION |
| | | This definition is used to explicitly reference the Money Transfer data fields in the Transaction Specific Data Token. The length of this definition is 150 characters. | |
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for the MONEY-XFER-DATA is 5F. | |
| | 02 | SEND-REF-NUM | PIC X(16) |
| | | Contains a transaction reference number that is provided by the originator and can be used to uniquely identify the sender. | |
| | 02 | SEND-ACCT-NUM | PIC X(34) |
| | | Contains the sender's card number or bank account number from which the funds are to be taken. | |
| | 02 | SEND-NAM | PIC X(30) |
| | | Contains the name of the person for the account from which the funds are to be taken. | |
| | 02 | SEND-ADDR | PIC X(35) |
| | | Contains the address of the person for the account from which the funds are to be taken. | |
| | 02 | SEND-CITY | PIC X(25) |
| | | Contains the city of the person for the account from which the funds are to be taken. | |
| | 02 | SEND-ST | PIC X(2) |
| | | Contains the geographical state or province of the person for the account from which the funds are to be taken. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | SEND-CNTRY | PIC X(3) |
| | | Contains the country of the person for the account from which the funds are to be taken. | |
| | 02 | FUND-SRC | PIC X(2) |
| | | Contains the method used by the sender to fund an enhanced money transfer OCT. Valid values are:<br>01 = Visa credit<br>02 = Visa debit<br>03 = Visa prepaid<br>04 = Cash<br>05 = Non-cash/non-credit other than Visa card<br>06 = Credit other than Visa card, includes credit cards and proprietary credit lines | |
| | 02 | USER-FLD-ACI | PIC X(1) |
| | | Reserved by ACI for future use. | |

---

| | | | |
|---|---|---|---|
| | | MC-MBR-DEF-DATA | DEFINITION |
| | | This definition is used to explicitly reference the MasterCard Member-Defined Data fields in the Transaction Specific Data Token. The length of this definition is 112 characters. | |
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for MC-MBR-DEF-DATA is 65. | |
| | 02 | MBR-DEF-DATA | PIC X(98) |
| | | This field contains MasterCard-specific transaction data. | |
| | 02 | AUTH-AGENT-ID-CDE | PIC X(6) |
| | | This field contains the value from MasterCard Data Element 121 - Authorizing Agent ID Code. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | MC-ASGN-ID | PIC X(6) |
| | | This field contains the value from MasterCard Data Element 48, Sub-element 32 - MasterCard Assigned ID | |
| | | MULT-PMNT-FORMS-DATA | DEFINITION |
| | | This definition is used to explicitly reference the Multiple Payment Forms Data fields in the Transaction Specific Data Token. The length of this definition is 4 characters. | |
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for MULT-PMNT-FORMS-DATA is 69. | |
| | 02 | NUM-OF-PMNT-FORMS | PIC X(1) |
| | | The number of different payment forms used in this split transaction. Valid values are as follows:<br>1 - 9 = Single digit indicating number of forms<br>+ = More than 9 forms | |
| | 02 | USER-FLD-ACI | PIC X(1) |
| | | Reserved by ACI for future use. | |
| | | TRVL-TAG-DATA | DEFINITION |
| | | This definition is used to explicitly reference the Travel Tag Data fields in the Transaction Specific Data Token. The length of this definition is 4 characters. | |
| | 02 | DATASET-ID | PIC X(2) |
| | | The DATASET-ID for TRVL-TAG-DATA is 6C. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | TRVL-TAG-CDE | PIC X(1) |

A code that describes the cardholder travel status. Valid values are as follows:

A = Cardholder may be traveling, destination matches purchased itinerary
B = Cardholder may be traveling, destination unknown

| | | | |
|---|---|---|---|
| | 02 | USER-FLD-ACI | PIC X(1) |

Reserved by ACI for future use.

---

LOAN-DETL-DATA DEFINITION

This definition is used to explicitly reference the Loan Details Data fields in the Transaction Specific Data Token. The length of this definition is 48 characters.

| | | | |
|---|---|---|---|
| | 02 | DATASET-ID | PIC X(2) |

The DATASET-ID for MC-MBR-DEF-DATAis 6E.

| | | | |
|---|---|---|---|
| | 02 | CRDHLDR-TAX-ID-TYP | PIC X(4) |

The cardholder tax ID type. Valid values are as follows:

CNPJ = Company tax ID
CPF = Consumer tax ID

| | | | |
|---|---|---|---|
| | 02 | CRDHLDR-TAX-ID | PIC X(15) |

The cardholder tax ID number.

| | | | |
|---|---|---|---|
| | 02 | ASSET-IND | PIC X(1) |

The asset indicator. Valid values are as follows:

Y = Yes
N = No

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 02 | LOAN-TYP<br>The loan ID type. | PIC X(20) |
| | 02 | MRCH-PROG-ID<br>The merchant program identifier. | PIC X(6) |

| | | | |
|---|---|---|---|
| | | FREE-FORM-TXT<br>This definition is used to explicitly reference the Free Form Text fields in the Transaction Specific Data Token. The length of this definition is 152 characters. | DEFINITION |
| | 02 | DATASET-ID<br>The DATASET-ID for the PRMTN-DATA is 71. | PIC X(2) |
| | 02 | FREE-FORM-DATA<br>This field contains either member-to-member data or Gateway transactions - MasterCard data. | PIC X(99) |
| | 02 | ORIG-CR-APPL-DATA<br>This field contains original credit application data. | PIC X(50) |
| | 02 | USER-FLD-ACI<br>Reserved by ACI for future use. | PIC X(1) |

# Token CU American Express Additional Data Token

The fields in the American Express Additional Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–304 | | AMEX-ADNL-DATA-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4–304 | 02 | INFO<br>Contains the Card Not Present (ITD) or Airline Passenger Data (APD) information. | PIC X(301) |

# Token CV Healthcare/Transit Token—Binary Format

The binary format Healthcare/Transit token contains information associated with healthcare/transit auto-substantiation transactions and healthcare eligibility inquiry transactions. The fields in the Healthcare/Transit token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–98 | | HLTHCR-TRANSIT-TKN | |
| 1–2 | 02 | NUM-ADNL-AMT<br>Indicates the number of entries in the ADNL-AMT table. | TYPE BINARY 16 |
| 3–98 | 02 | ADNL-AMT<br>Additional amount information. Contains up to six entries. | OCCURS 0 TO 6 TIMES DEPENDING ON NUM-ADNL-AMT |
| | 04 | ACCT-TYP<br>The type of account being used. A value of 00 indicates a non-specified type is used for healthcare/transit auto-substantiation transactions and healthcare eligibility inquiry transactions. | PIC X(2) |
| | 04 | AMT-TYP<br>The type of payment amount. Valid values are as follows:<br>3S = Amount co-payment<br>4S = Total amount healthcare<br>4T = Amount transit<br>4U = Amount prescription/Rx<br>4V = Amount vision/optical<br>4W = Amount clinic/other qualified medical<br>4X = Amount dental<br>57 = Original amount | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | CRNCY-CDE<br>The standard ISO currency code of the amount. | PIC 9(3) |
| | 04 | AMT-SIGN<br>A code indicating whether the amount is positive or negative. Valid values are as follows:<br>C = Credit, positive balance<br>D = Debit, negative balance | PIC X(1) |
| | 04 | AMT<br>The amount specified by the AMT-TYP field. | TYPE BINARY 64 |

# Token CV Healthcare/Transit Token—ASCII Format

The fields in the ASCII format Healthcare/Transit token are described below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–122 | | HLTHCR-TRANSIT-TKN | |
| 1–2 | 02 | NUM-ADNL-AMT | PIC X(2) |
| 3–122 | 02 | ADNL-AMT | OCCURS 6 TIMES |
| | 04 | ACCT-TYP | PIC X(2) |
| | 04 | AMT-TYP | PIC X(2) |
| | 04 | CRNCY-CDE | PIC 9(3) |
| | 04 | AMT-SIGN | PIC X(1) |
| | 04 | AMT | PIC X(12) |

# Token CW Healthcare Service Token—Binary Format

The binary format Healthcare Service token contains information associated with healthcare eligibility inquiry transactions. The fields in the Healthcare Service token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–102 | | HLTHCR-SRVC-TKN | |
| 1–2 | 02 | NUM-SRVC<br>Indicates the number of entries in the SVC table. | TYPE BINARY 16 |
| 3–102 | 02 | SRVC<br>Service information. Contains up to five entries. | OCCURS 0 TO 5 TIMES DEPENDING ON NUM-SRVC |
| | 04 | PROVIDER-ID<br>The medical license number of the healthcare service provider. | PIC X(9) |
| | 04 | TYP-CDE<br>The healthcare service type code as defined by the Health Insurance Portability and Accountability Act (HIPAA). | PIC X(2) |
| | 04 | PAYER-ID<br>The healthcare insurance carrier/payer identification. | PIC X(6) |
| | 04 | RSN-CDE<br>The eligibility approval or rejection reason code as defined by the HIPAA. | PIC X(2) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | USER-FLD1<br>Reserved for future use. | PIC X(1) |

# Token CW Healthcare Service Token—ASCII Format

The fields in the ASCII format Healthcare Service token are described below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–102 | | HLTHCR-SRVC-TKN | |
| 1–2 | 02 | NUM-SRVC | PIC X(2) |
| 3–102 | 02 | SRVC | OCCURS 5 TIMES |
| | 04 | PROVIDER-ID | PIC X(9) |
| | 04 | TYP-CDE | PIC X(2) |
| | 04 | PAYER-ID | PIC X(6) |
| | 04 | RSN-CDE | PIC X(2) |
| | 04 | USER-FLD1 | PIC X(1) |

# Token CX American Express Private Use Data Token

The American Express Private Use Data token contains information to provide support for all address verification formats. The fields in the American Express Private Use Data token are described below.

| | | | |
|---|---|---|---|
| 1–204 | | AMEX-PRVT-USE-DATA-TKN | |
| 1–3 | 02 | LGTH | PIC X(3) |
| 4-204 | 02 | INFO | PIC X(204) |
| 4–204 | 02 | AVS | REDEFINES INFO |
| 4–18 | 04 | FIRST-NAM | PIC X(15) |
| 19–48 | 04 | LST-NAM | PIC X(30) |
| 49–58 | 04 | PHN-NUM | PIC X(10) |
| 59–67 | 04 | SHIP-TO-ZIP-CDE | PIC X(9) |
| 68–117 | 04 | SHIP-TO-ADDR | PIC X(50) |
| 118–132 | 04 | SHIP-TO-FIRST-NAM | PIC X(15) |
| 133–162 | 04 | SHIP-TO-LST-NAM | PIC X(30 |
| 163–172 | 04 | SHIP-TO-PHN-NUM | PIC X(10) |
| 173–175 | 04 | SHIP-TO-CNTRY-CDE | PIC X(3) |
| 176–204 | 04 | USER-FLD1 | PIC X(29) |

# Token CY Auto-Substantiation Data Token

The fields in the Auto-Substantiation Data token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–50 | 00 | AUTO-SUBSTAN-DATA-TKN | |
| 1 | 02 | IIAS-IND | PIC X(1) |
| | | An indicator used to specify that an Inventory Informational Approval System (IIAS) was used by the merchant to identify qualified medical items at the point of sale. Valid values are as follows:<br>0 = An IIAS was not used at the point of sale.<br>1 = An IIAS was used at the point of sale.<br>2 = Merchant is exempt from using an IIAS.<br>4 = Transaction submitted as real-time substantiated but from a non-IIAS certified merchant. | |
| 2-11 | 02 | ASGN-ID | PIC X(10) |
| | | An identifier assigned to the merchant by MasterCard for IIAS validation, or for a Visa Merchant Verification Value (MVV). Identifies the merchant for real-time auto-substantiation. | |
| 12–50 | 02 | USER-FLD-ACI | PIX X(39) |
| | | This field is reserved for future expansion of auto-substantiation. The $offset of this field should be used as the token length when adding the token. | |

# Token CZ POS Data 2 Token—Binary Format

The fields in the POS Data 2 token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–34 | 00 | POS-DATA2-TKN | |
| 1–2 | 02 | ATC | PIC X(2) |
| | | The Application Transaction Counter (ATC) value from the base segment of the Cardholder Authorization File (CAF). The value in the token is the current value after ATC verification and Dynamic Card Verification have been performed. The largest ATC value is 65,535. The field is defined as non-integer, but will contain binary data. It should be initialized with binary zeroes. | |
| 3-6 | 02 | FORM-FACTR-IND | PIC X(4) |
| | | This field contains Visa-defined data to be used for the identification of the cardholder device, its security features, and the communication technology used to acquire a contactless transaction. The field is defined as non-integer, but will contain binary data. It should be initialized with binary zeroes. Valid values are as follows:<br>Byte 1 = Cardholder device type<br>Byte 2 = Cardholder device security features<br>Byte 3 = Reserved<br>Byte 4 = Communication technology | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 7 | 02 | ATC-VALID-IND<br>Indicates whether there is a discrepancy between the ATC value in the token and the ATC valule on the database. Valid values are as follows:<br>b = No information available (where b indicates a blank space)<br>G = ATC is greater than the maximum value allowed<br>L = ATC is less than the minimum value allowed<br>0 = ATC validity is unknown<br>W = ATC is within the validity limits | PIC X(1) |
| 8–34 | 02 | USER-FLD-ACI<br>This field is reserved for future use by ACI. | PIC X(27) |

# Token CZ  POS Data 2 Token—ASCII Format

The fields in the ASCII format POS Data 2 token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token. The ATC and FORM-FACTR-IND fields contain the hexidecimal character representation of the binary data.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–40 | | POS-DATA2-TKNX | |
| 1–4 | 02 | ATC | PIC X(4) |
| 5–12 | 02 | FORM-FACTR-IND | PIC X(8) |
| 13 | 02 | ATC-VALID-IND | PIC X(1) |
| 14–40 | 02 | USER-FLD-ACI | PIC X(27) |

# Token F1 E-commerce Additional Data Token

The fields of the E-commerce Additional Data token are detailed below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–30 | | E-COM-ADDL-DATA-TKN | |
| 1–15 | 02 | SCHEME-MERCH-ID<br>A scheme-assigned value that identifies the merchant in an e-commerce transaction. | PIC X(15) |
| 16 | 02 | CRDHLDR-AUTHN-IND<br>Indicates the cardholder authentication. Valid values are:<br>0 = None<br>1 = Authentication Value<br>2 = Electronic Commerce Service Indicator<br>3 = Chip Cryptogram | PIC X(1) |
| 17 | 02 | DEV-TYP<br>A code indicating the e-commerce device type. Valid values are:<br>␣ = Normal, no special conditions (where ␣ indicates a blank space). This is the default value<br>0 = Mobile device or server | PIC X(1) |
| 18–30 | 02 | USER-FLD-ACI<br>Reserved by ACI for future use. | PIC X(13) |

# Token F2 Installment Payment Data Token

This token contains installment payment response data required when a Greek issuer authorizes an installment transaction. The fields of the Installment Payment Data token are detailed below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–44 | | INSTL-PMNT-DATA-TKN | |
| 1-12 | 02 | TOTAL-AMT<br>Total issuer-calculated transaction amount, including any interest, insurance, or other charges. | PIC X(12) |
| 13-24 | 02 | INSTL-AMT<br>Installment amount, including amy issuer-calculated interest, insurance, or other charges. | PIC X(12) |
| 25-30 | 02 | FIRST-PMNT-DAT<br>Due date of first installment (DDMMYY). | PIC X(6) |
| 31-33 | 02 | ISS-CRNCY-CDE<br>Numeric currency code in which the issuer will finance the transaction. | PIC X(3) |
| 34-37 | 02 | INTRST-RAT<br>Issuer-calculated monthly interest rate (two decimal places). | PIC X(4) |
| 38-44 | 02 | USER-FLD-ACI<br>Reserved by ACI for future use. | PIC X(7) |

# Token F3 Transit Transaction Token

The fields of the Transit Transaction token are detailed below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–8 | | TRANSIT-TXN-TKN | |
| 1-2 | 02 | TXN-TYP-IND | PIC X(2) |

Identifies the attributes of a transit transaction, and whether the transaction relates to a single purchase or multiple (aggregated) purchases. Valid values are:

| | | |
|---|---|---|
| ␣ | = | No information available (where ␣ indicates a blank space) |
| 00 | = | Unknown |
| 01 | = | Pre-funded |
| 02 | = | Real-time authorized |
| 03 | = | Authorized aggregated |
| 04 | = | Authorized aggregate split clearing |
| 05 | = | Other |
| 06 | = | Reserved for future use |
| 07 | = | Debt recovery |
| 08-99 | = | Reserved for future use |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3-4 | 02 | TRNSPRT-MDE-IND | PIC X(2) |

Identifies the transportation mode for a transit transaction. Valid values are:

| | | |
|---|---|---|
| ␣ | = | No information available (where ␣ indicates a blank space) |
| 00 | = | Unknown |
| 01 | = | Urban bus |
| 02 | = | Inter-urban bus |
| 03 | = | Light train mass transit (underground metro, LTR) |
| 04 | = | Train |
| 05 | = | Commuter Train |
| 06 | = | Water-born vehicle |
| 07 | = | Toll |
| 08 | = | Parking |
| 09 | = | Taxi |
| 10 | = | High speed train |
| 11 | = | Rural bus |
| 12 | = | Express commuter train |
| 13 | = | Para transit |
| 14 | = | Self drive vehicle |
| 15 | = | Coach |
| 16 | = | Locamotive |
| 17 | = | Powered motor vehicle |
| 18 | = | Trailer |
| 19 | = | Regional train |
| 20 | = | Inter-city |
| 21 | = | Funicular train |
| 22 | = | Cable car |
| 23-99 | = | Reserved for future use |

| | | | |
|---|---|---|---|
| 5-8 | 02 | USER-FLD-ACI | PIC X(4) |

Reserved by ACI for future use.

# Token F4 Digital Wallet Token

The fields of the Digital Wallet token are detailed below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–14 | | DGTL-WALLET-TKN | |
| 1-2 | 02 | WALLET-IND-FLG<br>The wallet indicator flag.<br>Valid values are:<br>01 = MasterCard wallet<br>02 = Visa V.me<br>03 = NYCE Mobile Access | PIC X(2) |
| 3-14 | 02 | DGTL-WALLET-DATA<br>The generic data, variable length up to 12 characters. | PIC X(12) |
| | 02 | WALLET<br>MasterCard wallet program data. | REDEFINES DGTL-WALLET-DATA |
| 3-5 | 04 | ID<br>The wallet identifier. Identifies wallet payment transactions originated through MasterCard's through the PayPass Online platform or a different wallet.<br>Valid values are:<br>101 = PPOL Remote<br>102 = PPOL Remote NFC Payment | PIC X(3) |
| 6-14 | 04 | USER-FLD-ACI<br>Reserved by ACI for future use. | PIC X(9) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3-14 | 02 | VME<br><br>Visa V.me service. | REDEFINES DGTL-WALLET-DATA |
| 3-7 | 04 | ID<br><br>The agent unique ID. This value indicates the transaction was processed through V.me.<br><br>Valid values are:<br><br>a9001 = Visa Europe V.me | PIC X(5) |
| 8-9 | 04 | ADNL-AUTH-MTHD<br><br>The additional authentication method indicator that identifies how the V.me platform authenticated the transaction.<br><br>Valid values are:<br><br>01 = V.me authentication<br>02 = Additional 3D-Secure authentication<br>03 = Additional 3D-Secure attempted<br>04 = Additional one-time password | PIC X(2) |
| 10-11 | 04 | ADNL-AUTH-RSN-CDE<br><br>The additional authentication reason code used to indicate when one or more reasons have caused V.me to seek additional authentication. | PIC X(2) |
| 12-14 | 04 | USER-FLD-ACI<br><br>Reserved by ACI for future use. | PIC X(3) |
| 3-14 | 02 | NYCE-MOBILE-ACCESS<br><br>NYCE Mobile Access | REDEFINES DGTL-WALLET-DATA |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 3-4 | 04 | WALLET-TYP<br>Wallet type. Identifies the NYCE Mobile Access Wallet Type. Valid values are:<br><br>84 = Merchant Wallet<br>85 = FI Wallet | PIC X(2) |
| 5-14 | 04 | USER-FLD-ACI<br>Reserved by ACI for future use. | PIC X(10) |

# Token U0 EBT Voucher Token

The Electronic Benefit Transfer (EBT) Voucher token contains voucher information for manually entered EBT transactions. The fields in the EBT Voucher token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–48 | 00 | EBT-VOUCHER-NUM-TKN | |
| 1–24 | 02 | VOUCHER-NUM<br>The voucher number on the manually entered electronic benefit transfer (EBT) transaction. This number is used for voice-authorized transactions. | PIC X(24) |
| 25–48 | 02 | USER-FLD1<br>This field is reserved for future use. | PIC X(24) |

# Token U1 EBT Available Balance Token—Binary Format

The Electronic Benefit Transfer (EBT) Available Balance token contains the available balance for a food stamp or cash account for EBT transactions. The fields in the EBT Available Balance token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1-40 | 00 | EBT-AVAIL-BAL-TKN | |
| 1 | 02 | CASH-ACCT-BAL-IND<br>A flag indicating whether a balance is present for a cash benefit account. Valid values are as follows:<br>0 = No, the balance is not available.<br>1 = Yes, the balance present. | PIC 9(1) |
| 2 | 02 | USER-FLD1A<br>This field ensures word alignment. | PIC X(1) |
| 3–10 | 02 | CASH-ACCT-BAL<br>The available balance for the cash benefit account. | TYPE BINARY 64 SIGNED |
| 11 | 02 | FOOD-STMP-BAL-IND<br>A flag indicating whether a balance is present for a food stamp account. Valid values are as follows:<br>0 = No, the balance is not available.<br>1 = Yes, the balance present. | PIC 9(1) |
| 12 | 02 | USER-FLD1B<br>This field ensures word alignment. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 13–20 | 02 | FOOD-STMP-BAL<br>The available balance for the food stamp account. | TYPE BINARY 64 SIGNED |
| 21–40 | 02 | USER-FLD1<br>This field is not currently used. | PIC X(20) |

# Token U1 EBT Available Balance Token—ASCII Format

The fields in the ASCII format EBT Available Balance token are shown below. For descriptions of these fields, refer for the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–58 | 00 | EBT-AVAIL-BAL-TKNX | |
| 1 | 02 | CASH-ACCT-BAL-IND | PIC X(1) |
| 2–19 | 02 | CASH-ACCT-BAL | PIC X(18) |
| 20 | 02 | FOOD-STMP-BAL-IND | PIC X(1) |
| 21–38 | 02 | FOOD-STMP-BAL | PIC X(18) |
| 39–58 | 02 | USER-FLD1 | PIC X(20) |

# Token U2 Stored Value Token—Binary Format

The Router/Authorization module looks for the Stored Value token on all transactions initiated with a Stored Value card. If the transaction is not a stored value transaction, the Router/Authorization module adds the Stored Value token if it is not found. The Router/Authorization module then updates the token with the following information:

- PBF balance
- CAF expiration date
- CPF maximum balance as cash flag
- Reversal flag

The available balance in the Stored Value token is taken from the available balance from the PBF minus the total preauthorization holds. The balance can be negative if the purchase amount is greater than the available balance minus the total preauthorization holds, in which case the negative amount returned will indicate that portion of the purchase amount that must be tendered through some other form of payment.

*ACI Worldwide, Inc.*

The Stored Value token is identified with a token ID of U2. The format of the Stored Value token is as follows:

| Position | Level | Field name and Description | Data Type |
|---|---|---|---|
| 1-54 | | STORED-VALUE-TKN | |
| 1-8 | 02 | CRD-BAL | TYPE BINARY 64 SIGNED |
| | | The remaining balance on the stored value account. This amount is returned to the terminal following a stored value transaction. | |
| 9-12 | 02 | EXP-DAT | PIC 9(4) |
| | | The expiration date (YYMM) of the account. This field is returned to the terminal following a stored value transaction. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 13 | 02 | BAL-AS-CASH<br>Indicates whether or not the remaining balance on the stored value account can be given as cash. This flag is returned to the terminal following a stored value transaction. | PIC X(1) |
| 13 | 02 | PREV-ACCT-STAT<br>The previous status of the account. | REDEFINES BAL-AS-CASH PIC X(1) |
| 14 | 02 | SV-RVSL-FLG<br>A code indicating the records added in a stored value transaction. This flag is used in reversal processing. The Authorization module uses it to determine which, if any, records need to be deleted. Valid values are as follows:<br>0 = No records were added<br>1 = CAF record added<br>2 = PBF record added<br>3 = CAF and PBF record added<br>4 = No records added with PBF preloaded balance.<br>5 = CAF record added with PBF preloaded balance. | PIC X(1) |
| 14 | 02 | CUR-ACCT-STAT<br>The current status of the account. | REDEFINES SV-RVSL-FLG PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 15-54 | 02 | TRK2 | PIC X(40) |

The Track 2 data from the additional card. The data is used by the Router/Authorization processes and is only present on additional card activation transactions. The value of the field is set by the SPDH module. The Track 2 format is as follows:

1 = Start sentinel
2 = PAN, left-justified
3 = An equal sign (=)
4 = Member number (if one exists)
5 = County code (if one exists)
6 = Expiration date (if one exists)
7 = PIN offset (if one exists)
8 = Algorithm offset (if one exists)
9 = End sentinel
10 = Blank fill to the right

# Token U2 Stored Value Token—ASCII Format

The fields in the ASCII format Stored Value token are shown below. For descriptions of these fields, see the documentation for the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1-64 | | STORED-VALUE-TKNX | |
| 1-18 | 02 | CRD-BAL | PIC 9(18) |
| 19-22 | 02 | EXP-DAT | PIC 9(4) |
| 23 | 02 | BAL-AS-CASH | PIC X(1) |
| 23 | 02 | PREV-ACCT-STAT | REDEFINES BAL-AS-CASH |
| 24 | 02 | SV-RVSL-FLG | PIC X(1) |
| 24 | 02 | CUR-ACCT-STAT | REDEFINES SV-RVSL-FLG |
| 25-64 | 02 | TRK2 | PIC X(40) |

*ACI Worldwide, Inc.*

# 8: BASE24-teller Tokens

This section describes the BASE24-teller message tokens. Tokens are described in alphanumeric order, according to token ID. The table below identifies the BASE24-teller tokens and their corresponding token IDs. For tokens with ASCII formats, the ASCII formats follow the corresponding binary format.

| ID | Token Name |
|---|---|
| T0 | Financial token |
| T1 | CAF Inquiry token |
| T2 | CAF Update token |
| T3 | NBF token |
| T4 | PBF Inquiry token |
| T5 | PBF Update token |
| T6 | SPF Inquiry token |
| T7 | SPF Update token |
| T8 | WHFF Inquiry token |
| T9 | WHFF Update token |
| TA | Administrative token |
| TB | Account token |
| TC | Override token |
| TD | PIN token |
| TE | Native Message token |

Refer to section 5 for information about the Header token and token header.

# Token T0 Financial Token—Binary Format

The fields in the binary format Financial token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–86 | | FNCL-TKN<br>This token is used for financial transactions in conjunction with the following types of messages:<br>• Financial Transaction Request (0200)<br>• Financial Transaction Response (0210)<br>• Financial Transaction Advice (0220)<br>• Financial Transaction Advice Response (0230)<br>• Financial Transaction Reversal Advice (0420)<br>• Financial Transaction Reversal Response (0430) | |
| 1–2 | 02 | RVSL-CDE<br>A code indicating the reason for the reversal. This field is valid for 0420 messages only. Valid values are as follows:<br>01 = Time-out<br>02 = Command reject<br>03 = Destination not available<br>08 = Customer canceled<br>10 = Hardware error | PIC 9(2) |
| 3–10 | 02 | AMT-1<br>The original transaction amount requested.<br>For split deposit transactions, this is the amount for the first account. | TYPE BINARY 64 SIGNED |
| 11–18 | 02 | AMT-2<br>The amount of a customer's deposit that was in cash. | TYPE BINARY 64 SIGNED |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 19–26 | 02 | AMT-3<br>The amount disbursed to a customer in cash. | TYPE BINARY 64 SIGNED |
| 27–34 | 02 | AMT-4<br>The fee amount for a purchase transaction, or the deposit amount for the second account for split deposit transactions. | TYPE BINARY 64 SIGNED |
| 35–40 | 02 | SAF-DAT<br>The store-and-forward posting date (YYMMDD), if returned with the forced post transaction. | PIC X(6) |
| 41–51 | 02 | FRWD-INST-ID<br>The forwarding institution ID, taken from the LCONF param FORWARD-INST-ID. This field is used for logging only. Data within this field is right-justified and zero filled. | PIC 9(11) |
| 52–62 | 02 | CRD-ACCPT-ID<br>The institution ID number of the card acceptor on a 0200 transaction originating from an acquirer host. This information is used for logging only. Data within this field is right-justified and zero filled. This field is reserved for future use. | PIC 9(11) |
| 63–73 | 02 | CRD-OWNER-ID<br>The institution ID number of the actual card owner on a 0210 response from an authorizing host. The information is used for logging only. Data within this field is right-justified and zero filled. | PIC 9(11) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 74 | 02 | CNTR-CNT | PIC X(1) |
| | | The currency transaction report counter that indicates the number of reports filled out for a given account during the calendar day. The value for this field is taken from the CTR-CNT field in the BASE24-teller segment of the PBF. | |
| 75–85 | 02 | CHK-NUM | PIC X(11) |
| | | The check number of the check being cashed. If the check number is not applicable or not available, this field is initialized to blanks. Data within this field is right-justified and zero filled. | |
| 86 | 02 | AUTO-PASSBOOK-PRNT | PIC X(1) |
| | | Indicates whether automatic passbook printing is required for this financial transaction. Valid values are as follows:<br>Y = Yes, automatic passbook printing is required.<br>N = No, automatic passbook printing is not required.<br>This field is set by the Device Handler process. | |

# Token T0 Financial Token—ASCII Format

The fields in the ASCII format Financial token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–130 | | FNCL-TKNX | |
| 1–2 | 02 | RVSL-CDE | PIC 9(2) |
| 3–21 | 02 | AMT-1 | PIC X(19) |
| 22–40 | 02 | AMT-2 | PIC X(19) |
| 41–59 | 02 | AMT-3 | PIC X(19) |
| 60–78 | 02 | AMT-4 | PIC X(19) |
| 79–84 | 02 | SAF-DAT | PIC X(6) |
| 85–95 | 02 | FRWD-INST-ID | PIC 9(11) |
| 96–106 | 02 | CRD-ACCPT-ID | PIC 9(11) |
| 107–117 | 02 | CRD-OWNER-ID | PIC 9(11) |
| 118 | 02 | CNTR-CNT | PIC X(1) |
| 119–129 | 02 | CHK-NUM | PIC X(11) |
| 130 | 02 | AUTO-PASSBOOK-PRNT | PIC X(1) |

# Token T1 CAF Inquiry Token—Binary Format

The fields in the binary format CAF Inquiry token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–184 | | CAFI-TKN<br><br>This token contains CAF information. In addition to returning information for CAF inquiry transactions, it is used to return card-related information to the teller for both financial and file inquiry or update transactions when an overridable response code related to cards is encountered. This token also supports multiple account selection for card-initiated transactions when no accounts are entered by the teller. It can appear in the following types of messages:<br><br>• Financial Transaction Response (0210)<br>• Financial Transaction Advice (0220)<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320)<br>• File Inquiry/Update Advice Response (0330) | |
| 1 | 02 | PAGING-IND<br><br>A code indicating whether information on additional accounts is available. Valid values are as follows:<br><br>Y = Yes, more accounts are available.<br>N = No, more accounts are not available.<br><br>If this field contains the value Y, more accounts are identified in the CAF than could be reported on using this token. If a subsequent request for CAF information is sent to the Authorization process with this token, information on up to five additional accounts is returned. This cycle can be repeated until information on all accounts in the CAF (up to 16) has been returned. In the same manner, this token can also be used to request additional account information from the host. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | If this field contains the value N, information for all of the available accounts has been returned. | |
| 2–3 | 02 | CRD-TYP | PIC X(2) |
| | | A code identifying the type of card. The value for this field is taken from the CRD-TYP field in the Base segment of the CAF.<br><br>Codes used in this field are either reserved by BASE24 products or are user-defined. Refer to the ***BASE24 Base Files Maintenance Manual*** for a list of the reserved codes. | |
| 4 | 02 | CRD-STAT | PIC X(1) |
| | | A code indicating the card status from the CAF record. Valid values are as follows:<br><br>0 = Issued but not active<br>1 = Open<br>2 = Lost<br>3 = Stolen<br>4 = Restricted<br>5 = Open<br>6 = BASE24-pos only<br>9 = Closed<br>A = Referral<br>B = Maybe<br>C = Denial<br>D = Signature restricted<br>E = Country club<br>F = Expired card<br>G = Commercial | |
| 5–8 | 02 | EXP-DAT | PIC X(4) |
| | | The card expiration date (YYMM). | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 9–10 | 02 | FROM-IDX | TYPE BINARY 16 |
| | | The index position into the Accounts segment of the CAF for the *from* account, or first account, involved in the transaction. This field provides the ability to continue the search for additional account selection information. If additional information is not needed, this field should be set to 99. This field is only used for multiple account select processing. | |
| 11–12 | 02 | TO-IDX | TYPE BINARY 16 |
| | | The index position into the Accounts segment of the CAF for the *to* account, or second account, involved in the transaction. This field provides the ability to continue the search for additional account selection information. If additional information is not needed, this field should be set to 99. This field is only used for multiple account select processing. | |
| 13–14 | 02 | ACCT-CNT | TYPE BINARY 16 |
| | | A count indicating the number of occurrences of account information in this token. | |
| 15–184 | 02 | ACCT | OCCURS 0 TO 5 TIMES DEPENDING ON ACCT-CNT |
| | | The length of each ACCT occurrence is 34 positions. | |
| | 04 | TYP | PIC X(2) |
| | | A code indicating the type of account. Valid values are as follows:<br>00 = Not applicable<br>01–09 = DDA (checking account)<br>11, 14–19 = Savings<br>12 = IRA<br>13 = Certificate of deposit<br>21 = NOW<br>31, 33–39 = Credit card<br>32 = Credit line<br>41 = Installment loan | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|

42 = Mortgage
43 = Commercial loan
50 = Utility payment
51 = Utility 1 payment
52 = Utility 2 payment
53 = Utility 3 payment
54 = Utility 4 payment
55 = Utility 5 payment

The value in this field is taken from the TYP field in the Accounts segment of the CAF.

04 ACCT-IND PIC X(1)

A code indicating the account number with which this account information is associated. This field is used during multiple account select processing. Valid values are as follows:

F = *From* account
T = *To* account

04 USER-FLD2 PIC X(1)

04 NUM PIC X(19)

The application account number. The value for this field is taken from the NUM field in the Accounts segment of the CAF.

04 STAT PIC X(1)

A code indicating the current status of the application account. Valid values are as follows:

0, A, B, C = Inactive account
1, D, E, F, G, H, I = Open
2, J, K, L = Restricted to deposits
3, M, N, O, P, Q, R = Open primary account
4, S, T, U = Restricted primary account—deposit and inquiry transactions only
9, V, W, X, Y, Z = Closed

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | DESCR<br>The description of the account (for example, Vacation, Household). | PIC X(10) |

# Token T1 CAF Inquiry Token—ASCII Format

The fields in the ASCII format CAF Inquiry token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–184 | | CAFI-TKNX | |
| 1 | 02 | PAGING-IND | PIC X(1) |
| 2–3 | 02 | CRD-TYP | PIC X(2) |
| 4 | 02 | CRD-STAT | PIC X(1) |
| 5–8 | 02 | EXP-DAT | PIC X(4) |
| 9–10 | 02 | FROM-IDX | PIC 9(2) |
| 11–12 | 02 | TO-IDX | PIC 9(2) |
| 13–14 | 02 | ACCT-CNT | PIC 9(2) |
| 15–184 | 02 | ACCT | OCCURS 0 TO 5 TIMES DEPENDING ON ACCT-CNT |
| | 04 | TYP | PIC X(2) |
| | 04 | ACCT-IND | PIC X(1) |
| | 04 | USER-FLD2 | PIC X(1) |
| | 04 | NUM | PIC X(19) |
| | 04 | STAT | PIC X(1) |
| | 04 | DESCR | PIC X(10) |

# Token T2 CAF Update Token

The fields in the CAF Update token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–12 | | CAFU-TKN<br>This token contains CAF update information. It can appear in the following types of messages:<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry Advice (0320)<br>• File Inquiry Advice Response (0330) | |
| 1 | 02 | CRD-STAT<br>A code indicating the card status if the CRD-STAT field in the Base segment of the CAF is being updated. Valid values are as follows:<br>0 = Issued but not active<br>1 = Open<br>2 = Lost<br>3 = Stolen<br>4 = Restricted<br>5 = Open<br>6 = BASE24-pos only<br>9 = Closed<br>A = Referral<br>B = Maybe<br>C = Denial<br>D = Signature restricted<br>E = Country club<br>F = Expired card<br>G = Commercial | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 2 | 02 | ACCT-STAT | PIC X(1) |

A code indicating the account status if the ACCT.STAT field in the Accounts segment of the CAF is being updated. Valid values are as follows:

| | | |
|---|---|---|
| 0, A, B, C | = | No relationship (inactive account) |
| 1, D, E, F, G, H, I | = | Open |
| 2, J, K, L | = | Restricted to deposits |
| 3, M, N, O, P, Q, R | = | Open primary account |
| 4, S, T, U | = | Restricted primary account |
| 9, V, W, X, Y, Z | = | Closed |

| | | | |
|---|---|---|---|
| 3–12 | 02 | ACCT-DESCR | PIC X(10) |

The description of the account defined in the CAF record. Note that the description is defined for informational purposes only.

The description cannot be updated by a CAF Update transaction.

# Token T3 NBF Token—Binary Format

The fields in the binary format NBF token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–420 | | NBF-TKN<br>This token contains NBF information for NBF print, NBF reprint, and NBF inquiry transactions. It is also returned for financial transactions when automatic passbook printing is supported. It can appear in the following types of messages:<br>• Financial Transaction Response (0210)<br>• Financial Transaction Advice (0220)<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320)<br>• File Inquiry/Update Advice Response (0330)<br>• Financial Transaction Reversal Advice (0420)<br>• Financial Transaction Reversal Advice Response (0430) | |
| 1 | 02 | ACCT-IND<br>A code indicating the account number with which this passbook information is associated. Valid values are as follows:<br>F = *From* account<br>T = *To* account | PIC X(1) |
| 2 | 02 | PAGING-IND<br>A code indicating whether there are additional NBF records. Valid values are as follows:<br>Y = Yes, there are additional NBF records.<br>N = No, there are not additional NBF records. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | If this field contains the value Y, more NBF records exist for this account than could be reported on using this token. If a subsequent request for NBF information is sent to the Authorization process with this token, up to five additional records are returned. This cycle can be repeated until all NBF records have been returned. In the same manner, this token can also be used to request no book information from the host.<br>If this field contains the value N, information for all of the records in the NBF for this account have been returned. | |
| 3–10 | 02 | STRT-BAL<br>The starting balance of the passbook prior to the passbook update. | TYPE BINARY 64 SIGNED |
| 11–18 | 02 | END-BAL<br>The ending balance of the passbook after these entries have been made to the passbook. | TYPE BINARY 64 SIGNED |
| 19–20 | 02 | REC-CNT<br>A count indicating the number of passbook entries in this token. This token can contain information from up to five NBF records. | TYPE BINARY 16 |
| 21–420 | 02 | REC<br>The length of each REC occurrence is 80 positions. | OCCURS 0 TO 5 TIMES DEPENDING ON REC-CNT |
| | 04 | TRAN-DAT<br>The date (YYMMDD) the record was added to the NBF. | PIC X(6) |
| | 04 | TRAN-TIM<br>The time (hhmmsstt) the record was added to the NBF. | PIC X(8) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | POST-DAT | PIC X(6) |
| | | The date (YYMMDD) the record was logged to the Teller Transaction Log File (TTLF). | |
| | 04 | TRAN-CDE | PIC X(6) |
| | | The transaction code that identifies the type of transaction that created this record.<br>Positions 1 and 2 identify the transaction type, positions 3 and 4 identify the *from* account type, and positions 5 and 6 identify the *to* account type. | |
| | 04 | TRAN-TYP | PIC X(2) |
| | | Indicates whether the transaction is a debit, a credit, a debit reversal, or a credit reversal. Valid values are as follows:<br>D$\not b$ = Debit ($\not b$ denotes a blank character)<br>C$\not b$ = Credit ($\not b$ denotes a blank character)<br>DR = Debit reversal<br>CR = Credit reversal | |
| | 04 | TRAN-AMT | TYPE BINARY 64 SIGNED |
| | | The amount of the transaction. | |
| | 04 | BOOK-BAL | TYPE BINARY 64 SIGNED |
| | | The passbook balance, after this transaction has been applied. | |
| | 04 | PRNT-IND | PIC X(1) |
| | | A code indicating whether the transaction has been printed in the customer's passbook. Valid values are as follows:<br>Y = Yes, the record has been printed.<br>N = No, the record has not been printed. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | DEV-TRAN-CDE | PIC X(6) |
| | | The device transaction code that defines the type of transaction that the teller performed. | |
| | 04 | TLR-ID | PIC X(8) |
| | | The identification number of the teller who performed the transaction. | |
| | 04 | REGN-ID | PIC X(4) |
| | | The region of the terminal where the transaction was initiated. | |
| | 04 | BRCH-ID | PIC X(4) |
| | | The regional branch of the terminal where the transaction was initiated. | |
| | 04 | CITY | PIC X(13) |
| | | The city in which the terminal where the transaction was initiated is located. | |

# Token T3 NBF Token—ASCII Format

The fields in the ASCII format NBF token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–552 | | NBF-TKNX | |
| 1 | 02 | ACCT-IND | PIC X(1) |
| 2 | 02 | PAGING-IND | PIC X(1) |
| 3–21 | 02 | STRT-BAL | PIC X(19) |
| 22–40 | 02 | END-BAL | PIC X(19) |
| 41–42 | 02 | REC-CNT | PIC 9(2) |
| 43–552 | 02 | REC | OCCURS 0 TO 5 TIMES DEPENDING ON REC-CNT |
| | 04 | TRAN-DAT | PIC X(6) |
| | 04 | TRAN-TIM | PIC X(8) |
| | 04 | POST-DAT | PIC X(6) |
| | 04 | TRAN-CDE | PIC X(6) |
| | 04 | TRAN-TYP | PIC X(2) |
| | 04 | TRAN-AMT | PIC X(19) |
| | 04 | BOOK-BAL | PIC X(19) |
| | 04 | PRNT-IND | PIC X(1) |
| | 04 | DEV-TRAN-CDE | PIC X(6) |
| | 04 | TLR-ID | PIC X(8) |
| | 04 | REGN-ID | PIC X(4) |
| | 04 | BRCH-ID | PIC X(4) |
| | 04 | CITY | PIC X(13) |

# Token T4 PBF Inquiry Token—Binary Format

The fields in the binary format PBF Inquiry token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–308 | | PBFI-TKN<br>This token contains PBF information. In addition to returning information for PBF inquiry transactions, it is used to return account-related information to the teller for both financial and file inquiry or update transactions when an overridable response code related to an account is encountered. It can appear in the following types of messages:<br>• Financial Transaction Response (0210)<br>• Financial Transaction Advice (0220)<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320)<br>• File Inquiry/Update Advice Response (0330) | |
| 1–2 | 02 | PBF-CNT<br>A count indicating the number of occurrences of PBF information in this token. | TYPE BINARY 16 |
| 3–308 | 02 | PBF<br>The length of each PBF occurrence is 102 positions. | OCCURS 0 TO 3 TIMES DEPENDING ON PBF-CNT |
| | 04 | ACCT-IND<br>A code indicating the account number with which this PBF information is associated. Valid values are as follows:<br>C = Credit line/backup account<br>F = *From* account<br>T = *To* account | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | CONFIDENTIAL-FLG | PIC X(1) |

A code indicating whether additional security is to be enforced when accessing the data in this record. This field is currently used for informational purposes only. Valid values are as follows:

0 = Normal (default)
1 = Confidential indicator

| | | | |
|---|---|---|---|
| | 04 | OVRDRFT-LMT | TYPE BINARY 32 SIGNED |

The amount of overdraft protection for this account, if applicable.

The overdraft limit is added to the available balance to obtain the maximum withdrawal amount available to a customer.

| | | | |
|---|---|---|---|
| | 04 | AMT-ON-HLD | TYPE BINARY 64 SIGNED |

The total amount of funds being held and not available to the customer.

An example of held funds is a deposit that is not verified, or a deposit consisting of checks.

Although the funds may be credited to the current account balance, the customer cannot make use of the funds until they are verified or cleared.

| | | | |
|---|---|---|---|
| | 04 | ACCRUED-INTEREST-YTD | TYPE BINARY 64 SIGNED |

The interest accrued on this account since the beginning of the current year.

| | | | |
|---|---|---|---|
| | 04 | LAST-DEP-DAT | PIC X(6) |

The last deposit date (YYMMDD).

| | | | |
|---|---|---|---|
| | 04 | LAST-DEP-AMT | TYPE BINARY 64 SIGNED |

The last deposit amount.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | LAST-WDL-DAT<br>The last withdrawal date (YYMMDD). | PIC X(6) |
| | 04 | LAST-WDL-AMT<br>The last withdrawal amount. | TYPE BINARY 64 SIGNED |
| | 04 | CASH-OUT<br>The total amount of cash paid out during a single business day. | TYPE BINARY 64 SIGNED |
| | 04 | CASH-IN<br>The total amount of cash deposited during a single business day. | TYPE BINARY 64 SIGNED |
| | 04 | AMT-DEP-CR<br>The total amount of credit given to this account for deposits today. | TYPE BINARY 64 SIGNED |
| | 04 | NUM-OF-DEP<br>The total number of deposit transactions for the day. | TYPE BINARY 16 |
| | 04 | STRT-BAL<br>The ledger balance of the account on the host as of the last refresh. This field is informational only, and is not used by the BASE24-teller product except to display on screen. The value in this field is taken from the STRT-BAL field in the BASE24-teller segment of the PBF. | TYPE BINARY 64 SIGNED |
| | 04 | PASSBOOK-BAL<br>The passbook balance from the PBF. | TYPE BINARY 64 SIGNED |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | NBF-REC-CNT | TYPE BINARY 16 |
| | | The number of NBF records on file for this account. | |
| | 04 | PASSBOOK-IND | PIC X(1) |
| | | A code indicating whether the account is a passbook account. | |
| | 04 | CUST-CLASS | PIC X(1) |
| | | A code indicating the customer class. This field is used in determining how much cash is available to the customer after a check deposit. | |
| | | Ten options are available. The options are defined by each institution and are maintained in the IDF. Valid values are 0 through 9. | |
| | 04 | CASHOUT-LMT | TYPE BINARY 16 |
| | | The cash-out transaction limit, in multiples of 1,000. | |
| | | If this field contains zeros, it indicates the account is exempt from limit checking. | |
| | 04 | CASHIN-LMT | TYPE BINARY 16 |
| | | The cash-in transaction limit, in multiples of 1,000. | |
| | | If this field contains zeros, it indicates the account is exempt from limit checking. | |
| | 04 | SP-WARN-STAT | PIC X(1) |
| | | A code indicating whether stop payments or warnings have been placed on the account. Valid values are as follows: | |
| | | 0 = No stops or warnings<br>1 = Stops<br>2 = Warnings<br>3 = Stops and warnings | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | CNTR-CNT | PIC X(1) |
| | | The currency transaction report counter that indicates the number of reports filled out for a given account during a calendar day. | |

# Token T4 PBF Inquiry Token—ASCII Format

The fields in the ASCII format PBF Inquiry token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–668 | | PBFI-TKNX | |
| 1–2 | 02 | PBF-CNT | PIC 9(2) |
| 3–668 | 02 | PBF | OCCURS 0 TO 3 TIMES DEPENDING ON PBF-CNT |
| | 04 | ACCT-IND | PIC X(1) |
| | 04 | CONFIDENTIAL-FLG | PIC X(1) |
| | 04 | OVRDRFT-LMT | PIC X(12) |
| | 04 | AMT-ON-HLD | PIC X(19) |
| | 04 | ACCRUED-INTEREST-YTD | PIC X(19) |
| | 04 | LAST-DEP-DAT | PIC X(6) |
| | 04 | LAST-DEP-AMT | PIC X(19) |
| | 04 | LAST-WDL-DAT | PIC X(6) |
| | 04 | LAST-WDL-AMT | PIC X(19) |
| | 04 | CASH-OUT | PIC X(19) |
| | 04 | CASH-IN | PIC X(19) |
| | 04 | AMT-DEP-CR | PIC X(19) |
| | 04 | NUM-OF-DEP | PIC 9(5) |
| | 04 | STRT-BAL | PIC X(19) |
| | 04 | PASSBOOK-BAL | PIC X(19) |
| | 04 | NBF-REC-CNT | PIC 9(5) |
| | 04 | PASSBOOK-IND | PIC X(1) |
| | 04 | CUST-CLASS | PIC X(1) |
| | 04 | CASHOUT-LMT | PIC 9(5) |
| | 04 | CASHIN-LMT | PIC 9(5) |
| | 04 | SP-WARN-STAT | PIC X(1) |
| | 04 | CNTR-CNT | PIC X(1) |
| | 04 | USER-FLD | PIC X(1) |

# Token T5 PBF Update Token

The fields in the PBF Update token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–2 | | PBFU-TKN<br>This token contains PBF update information. It can appear in the following types of messages:<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320)<br>• File Inquiry/Update Advice Response (0330) | |
| 1 | 02 | STAT-UPDT<br>If the RQST.TRAN.CDE field contains the value 82, this field contains a code indicating the new account status for the ACCT-STAT field in the Base segment of the PBF. Valid values are as follows:<br>0, A, B, C = No relationship (inactive account)<br>1, D, E, F, G, H, I = Open<br>2, J, K, L = Restricted to deposits<br>3, M, N, O, P, Q, R = Open primary account<br>4, S, T, U = Restricted primary account<br>9, V, W, X, Y, Z = Closed<br>If the RQST.TRAN.CDE field contains the value 83, this field contains a code indicating the new stop payment/warning status for the SP-STAT field in the BASE24-teller segment of the PBF is being updated. Valid values are as follows:<br>0 = No stops or warnings<br>1 = Stops<br>2 = Warnings<br>3 = Stops and warnings | PIC X(1) |
| 2 | 02 | USER-FLD1 | PIC X(1) |

# Token T6 SPF Inquiry Token—Binary Format

The fields in the binary format SPF Inquiry token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–458 | | SPFI-TKN<br>This token contains SPF information. In addition to returning information for SPF inquiry transactions, it is used to return stop payment information to the teller for cash check transactions when an overridable response code related to stop payments is encountered. It can appear in the following types of messages:<br>• Financial Transaction Response (0210)<br>• Financial Transaction Advice (0220)<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320)<br>• File Inquiry/Update Advice Response (0330) | |
| 1 | 02 | ACCT-IND<br>A code indicating the account number with which this SPF information is associated. Valid values are as follows:<br>F = *From* account<br>T = *To* account | PIC X(1) |
| 2–12 | 02 | SEARCH-HI-CHK-NUM<br>The check number used to search the SPF for an SPF inquiry transaction.<br>For a range of check numbers, this field identifies the highest check number in the range. If a value is not entered by the teller, this field contains blanks and all SPF records for the application account are returned.<br>Data within this field is right-justified and zero filled. | PIC X(11) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 13–23 | 02 | SEARCH-LO-CHK-NUM | PIC X(11) |

For a range of check numbers, this field identifies the lowest check number in the range. This field is used to search the SPF during an SPF inquiry transaction. Data within this field is right-justified and zero filled.

For a single check search, this field contains blanks.

| | | | |
|---|---|---|---|
| 24 | 02 | CHK-FOR-EXPIRED-RECS | PIC X(1) |

A code indicating whether the Authorization process should check the expiration date for each SPF record before it is placed in this token. This field is used to search the SPF during an SPF inquiry transaction. Valid values are as follows:

Y = Yes, do check for expired records. Expired records are not placed in this token.
N = No, do not check for expired records. Expired records are placed in this token along with other records.

| | | | |
|---|---|---|---|
| 25 | 02 | PBF-SP-WARN-STAT | PIC X(1) |

The value retrieved from the SP-STAT field in the BASE24-teller segment of the PBF. Valid values are as follows:

0 = No stops or warnings
1 = Stops
2 = Warnings
3 = Stops and warnings

| | | | |
|---|---|---|---|
| 26 | 02 | PAGING-IND | PIC X(1) |

A code indicating whether there are additional SPF records associated with this account.

Y = Yes, there are additional records.
N = No, there are no additional records.

If this field contains the value Y, more SPF records exist for this account than could be reported on using this token. If a subsequent request for SPF information is sent to the Authorization process with this token, up to five additional

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | records are returned. This cycle can be repeated until all SPF records have been returned. In the same manner, this token can also be used to request stop payment information from the host.<br><br>If this field contains the value N, information for all of the records meeting the search criteria have been returned. | |
| 27–28 | 02 | REC-CNT | TYPE BINARY 16 |
| | | A count indicating the number of SPF records in this token. This token can contain information from up to five SPF records. | |
| 29–458 | 02 | REC | OCCURS 0 TO 5 TIMES DEPENDING ON REC-CNT |
| | | The following fields contain information from the SPF record.<br><br>The length of each REC occurrence is 86 positions. | |
| | 04 | HI-CHK-NUM | PIC X(11) |
| | | The check number of the stop payment. Data within this field is right-justified and zero filled.<br><br>For a range of check numbers, this field contains the highest check number in the range. | |
| | 04 | LO-CHK-NUM | PIC X(11) |
| | | The lowest check number in a range of check numbers. For a single check, this field contains blanks.<br><br>Data within this field is right-justified and zero filled. | |
| | 04 | AMT | TYPE BINARY 64 SIGNED |
| | | The stop pay amount in whole and fractional currency units. For records associated with a range of checks, this field contains 0. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | DAT<br>The date (YYMMDD) of the stop payment. | PIC X(6) |
| | 04 | TIM<br>The time (hhmmsstt) of the stop payment. | PIC X(8) |
| | 04 | EXP-DAT<br>The expiration date (YYMMDD) of the stop payment. The value 000000 indicates the record does not expire. | PIC X(6) |
| | 04 | DESCR<br>The description of the stop payment. This field contains any additional information about the stop payment order. | PIC X(35) |
| | 04 | USER-FLD1 | PIC X(1) |

# Token T6 SPF Inquiry Token—ASCII Format

The fields in the ASCII format SPF Inquiry token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–508 | | SPFI-TKNX | |
| 1 | 02 | ACCT-IND | PIC X(1) |
| 2–12 | 02 | SEARCH-HI-CHK-NUM | PIC X(11) |
| 13–23 | 02 | SEARCH-LO-CHK-NUM | PIC X(11) |
| 24 | 02 | CHK-FOR-EXPIRED-RECS | PIC X(1) |
| 25 | 02 | PBF-SP-WARN-STAT | PIC X(1) |
| 26 | 02 | PAGING-IND | PIC X(1) |
| 27–28 | 02 | REC-CNT | PIC 9(2) |
| 29–508 | 02 | REC | OCCURS 0 TO 5 TIMES DEPENDING ON REC-CNT |
| | 04 | HI-CHK-NUM | PIC X(11) |
| | 04 | LO-CHK-NUM | PIC X(11) |
| | 04 | AMT | PIC X(19) |
| | 04 | DAT | PIC X(6) |
| | 04 | TIM | PIC X(8) |
| | 04 | EXP-DAT | PIC X(6) |
| | 04 | DESCR | PIC X(35) |

# Token T7 SPF Update Token—Binary Format

The fields in the binary format SPF Update token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–86 | | SPFU-TKN<br>This token contains SPF update information. It can appear in the following types of messages:<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320)<br>• File Inquiry/Update Advice Response (0330) | |
| 1–11 | 02 | HI-CHK-NUM<br>The check number for the stop payment that is being added or deleted. Data within this field is right-justified and zero filled.<br>For a range of check numbers, this field contains the highest check number in the range. | PIC X(11) |
| 12–22 | 02 | LO-CHK-NUM<br>The lowest check number in a range of check numbers. For a single check, this field contains blanks. Data within this field is right-justified and zero filled. | PIC X(11) |
| 23–30 | 02 | AMT<br>The stop pay amount in whole and fractional currency units (if entered by the teller) for the stop payment that is being added or deleted. | TYPE BINARY 64 SIGNED |
| 31–36 | 02 | DAT<br>The date (YYMMDD) for the stop payment that is being added or deleted. If the date is not entered by the teller, this field defaults to the current date. | PIC X(6) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 37–44 | 02 | TIM | PIC X(8) |
| | | The time (hhmmsstt) for the stop payment that is being added or deleted.<br><br>If the time is not entered by the teller, this field defaults to the current time. | |
| 45–50 | 02 | EXP-DAT | PIC X(6) |
| | | The expiration date (YYMMDD) for the stop payment that is being added or deleted. | |
| 51–85 | 02 | DESCR | PIC X(35) |
| | | The description for the stop payment that is being added or deleted. This field contains any additional information about the stop payment order. | |
| 86 | 02 | PBF-SP-WARN-STAT | PIC X(1) |
| | | A code indicating the new value for the SP-STAT field in the BASE24-teller segment of the PBF (if entered by the teller). Valid values are as follows:<br><br>0 = No stops or warnings<br>1 = Stops<br>2 = Warnings<br>3 = Stops and warnings<br><br>If this field is left blank, the BASE24-teller product calculates the new value. | |

# Token T7 SPF Update Token—ASCII Format

The fields in the ASCII format SPF Update token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–98 | | SPFU-TKNX | |
| 1–11 | 02 | HI-CHK-NUM | PIC X(11) |
| 12–22 | 02 | LO-CHK-NUM | PIC X(11) |
| 23–41 | 02 | AMT | PIC X(19) |
| 42–47 | 02 | DAT | PIC X(6) |
| 48–55 | 02 | TIM | PIC X(8) |
| 56–61 | 02 | EXP-DAT | PIC X(6) |
| 62–96 | 02 | DESCR | PIC X(35) |
| 97 | 02 | PBF-SP-WARN-STAT | PIC X(1) |
| 98 | 02 | USER-FLD1 | PIC X(1) |

# Token T8 WHFF Inquiry Token—Binary Format

The fields in the binary format WHFF Inquiry token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–1038 | | WHFFI-TKN<br>This token contains WHFF information. In addition to returning information for WHFF inquiry transactions, it is used to return warning, hold, and float information to the teller for financial transactions when an overridable response code related to warnings, holds, or floats is encountered. It can appear in the following types of messages:<br>• Financial Transaction Response (0210)<br>• Financial Transaction Advice (0220)<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320)<br>• File Inquiry/Update Advice Response (0330) | |
| 1–2 | 02 | SEARCH-REC-TYP<br>The record type used to search the WHFF for a WHFF inquiry transaction.<br>If this field contains blanks, all WHFF records for the application account are returned. Valid values are as follows:<br>b b = All records (b denotes a blank character).<br>01 = Warning<br>02 = Hold<br>03 = Deposit float | PIC X(2) |
| 3 | 02 | AMT-SEARCH-REQ<br>A code indicating whether a search for a specific amount is required for a WHFF inquiry transaction. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | If a value is not entered by the teller, all WHFF records for the application account of the above record type are returned. Valid values are as follows:<br>Y = Yes, search for a specific amount.<br>N = No, do not search for a specific amount.<br>If this field contains the value Y, the Authorization process uses the values in the SEARCH-AMT field to locate a WHFF record. | |
| 4 | 02 | USER-FLD1 | PIC X(1) |
| 5–12 | 02 | SEARCH-AMT<br>The amount used to search the WHFF for a WHFF inquiry transaction. This field is only used when the AMT-SEARCH-REQ field contains the value Y. | TYPE BINARY 64 SIGNED |
| 13–18 | 02 | SEARCH-DAT<br>The date (YYMMDD) used to search the WHFF for a WHFF inquiry transaction. This field is only used when the AMT-SEARCH-REQ field contains the value Y and this field does not contain spaces. | PIC X(6) |
| 19–26 | 02 | SEARCH-TIM<br>The time (hhmmsstt) used to search the WHFF for a WHFF inquiry transaction. This field is only used when the AMT-SEARCH-REQ field contains the value Y, the SEARCH-DAT field does not contain spaces, and this field does not contain spaces. | PIC X(8) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 27 | 02 | CHK-FOR-EXPIRED-RECS | PIC X(1) |
| | | A code indicating whether the Authorization process should check the expiration date for each WHFF record before it is placed in this token. Valid values are as follows:<br>Y = Yes, check for expired records. Expired records are not placed in this token.<br>N = No, do not check for expired records. Expired records are placed in this token along with other records.<br>This field can also be used to indicate that the host should check the expiration date of its warning, hold, or float information before placing the information in this token. | |
| 28 | 02 | USER-FLD2 | PIC X(1) |
| 29–30 | 02 | WHFF-CNT | TYPE BINARY 16 |
| | | A count indicating the number of occurrences of WHFF information in this token. | |
| 31–1038 | 02 | WHFF | OCCURS 0 TO 3 TIMES DEPENDING ON WHFF-CNT |
| | | The length of each WHFF occurrence is 336 positions. | |
| | 04 | ACCT-IND | PIC X(1) |
| | | A code indicating the account number with which this WHFF information is associated. Valid values are as follows:<br>C = Credit line/backup account<br>F = *From* account<br>T = *To* account | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 04 | PBF-SP-WARN-STAT | PIC X(1) |

A code indicating the value retrieved from the SP-STAT field in the BASE24-teller segment of the PBF. Valid values are as follows:

0 = No stops or warnings
1 = Stops
2 = Warnings
3 = Stops and warnings

| | 04 | PAGING-IND | PIC X(1) |
|---|---|---|---|

A code indicating whether there are additional WHFF records. Valid values are as follows:

Y = Yes, there are additional records.
N = No, there are no additional records.

If this field contains the value Y, more WHFF records exist for this account than could be reported on using this token. If a subsequent request for WHFF information is sent to the Authorization process with this token, up to five additional records are returned. This cycle can be repeated until all WHFF records have been returned. In the same manner, this token can also be used to request warning, hold, and float information from the host.

If this field contains the value N, information for all of the records in the WHFF for this account have been returned.

| | 04 | USER-FLD3 | PIC X(1) |
|---|---|---|---|

| | 04 | REC-CNT | TYPE BINARY 16 |
|---|---|---|---|

A count indicating the number of WHFF records in this token.

| | 04 | REC | OCCURS 5 TIMES DEPENDING ON REC-CNT |
|---|---|---|---|

Data within this field is right-justified and zero filled.

The length of each REC occurrence is 66 positions.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | 06 | REC-TYP | PIC X(2) |
| | | A code indicating the type of record. Valid values are as follows:<br>01 = Warning<br>02 = Hold<br>03 = Deposit float | |
| | 06 | AMT | TYPE BINARY 64 SIGNED |
| | | The warning, hold, or deposit float amount in whole and fractional currency units. | |
| | 06 | DAT | PIC X(6) |
| | | The date (YYMMDD) of the WHFF record. | |
| | 06 | TIM | PIC X(8) |
| | | The time (hhmmsstt) of the WHFF record. | |
| | 06 | EXP-DAT | PIC X(6) |
| | | The expiration date (YYMMDD) of the WHFF record. The value 000000 indicates that the record has no expiration date. | |
| | 06 | DESCR | PIC X(35) |
| | | The description of the WHFF record.<br>This field contains any additional information about the WHFF record. | |
| | 06 | USER-FLD4 | PIC X(1) |

# Token T8 WHFF Inquiry Token—ASCII Format

The fields in the ASCII format WHFF Inquiry token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–1192 | | WHFFI-TKNX | |
| 1–2 | 02 | SEARCH-REC-TYP | PIC X(2) |
| 3 | 02 | AMT-SEARCH-REQ | PIC X(1) |
| 4–22 | 02 | SEARCH-AMT | PIC X(19) |
| 23–28 | 02 | SEARCH-DAT | PIC X(6) |
| 29–36 | 02 | SEARCH-TIM | PIC X(8) |
| 37 | 02 | CHK-FOR-EXPIRED-RECS | PIC X(1) |
| 38 | 02 | USER-FLD2 | PIC X(1) |
| 39–40 | 02 | WHFF-CNT | PIC 9(2) |
| 41–1192 | 02 | WHFF | OCCURS 0 TO 3 TIMES DEPENDING ON WHIFF-CNT |
| | 04 | ACCT-IND | PIC X(1) |
| | 04 | PBF-SP-WARN-STAT | PIC X(1) |
| | 04 | PAGING-IND | PIC X(1) |
| | 04 | REC-CNT | PIC 9(1) |
| | 04 | REC | OCCURS 0 TO 5 TIMES DEPENDING ON REC-CNT |
| | 06 | REC-TYP | PIC X(2) |
| | 06 | AMT | PIC X(19) |
| | 06 | DAT | PIC X(6) |
| | 06 | TIM | PIC X(8) |
| | 06 | EXP-DAT | PIC X(6) |
| | 06 | DESCR | PIC X(35) |

# Token T9 WHFF Update Token—Binary Format

The fields in the binary format WHFF Update token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–66 | | WHFFU-TKN<br>This token contains WHFF update information. It can appear in the following types of messages:<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320)<br>• File Inquiry/Update Advice Response (0330) | |
| 1–2 | 02 | REC-TYP<br>A code indicating the type of WHFF record that is being added or deleted. Valid values are as follows:<br>01 = Warning<br>02 = Hold<br>03 = Deposit float | PIC X(2) |
| 3–10 | 02 | AMT<br>The warning, hold, or deposit float amount in whole and fractional currency units. | TYPE BINARY 64 SIGNED |
| 11–16 | 02 | DAT<br>The date (YYMMDD) for the WHFF record that is being added or deleted. If the date is not entered by the teller, this field defaults to the current date. | PIC X(6) |
| 17–24 | 02 | TIM<br>The time (hhmmsstt) for the WHFF record that is being added or deleted. If the time is not entered by the teller, this field defaults to the current time. | PIC X(8) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 25–30 | 02 | EXP-DAT | PIC X(6) |
| | | The expiration date (YYMMDD) for the WHFF record that is being added or deleted. The value 000000 indicates that the record has no expiration date. | |
| 31–65 | 02 | DESCR | PIC X(35) |
| | | The description for the WHFF record that is being added or deleted. This field contains any additional information about the WHFF record. | |
| 66 | 02 | PBF-SP-WARN-STAT | PIC X(1) |
| | | A code indicating the new value for the SP-STAT field in the BASE24-teller segment of the PBF (if entered by the teller). Valid values are as follows:<br>0 = No stops or warnings<br>1 = Stops<br>2 = Warnings<br>3 = Stops and warnings<br>If this field is left blank, the BASE24-teller product calculates the new value. | |

# Token T9 WHFF Update Token—ASCII Format

The fields in the ASCII format WHFF Update token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–78 | | WHFFU-TKNX | |
| 1–2 | 02 | REC-TYP | PIC X(2) |
| 3–21 | 02 | AMT | PIC X(19) |
| 22–27 | 02 | DAT | PIC X(6) |
| 28–35 | 02 | TIM | PIC X(8) |
| 36–41 | 02 | EXP-DAT | PIC X(6) |
| 42–76 | 02 | DESCR | PIC X(35) |
| 77 | 02 | PBF-SP-WARN-STAT | PIC X(1) |
| 78 | 02 | USER-FLD1 | PIC X(1) |

# Token TA Administrative Token

The Administrative token contains user-defined administrative data. The Administrative token is not used by the BASE24-teller product for processing, but can be passed in the internal and external message, logged to the TTLF, or extracted from the TTLF. The basic structure of the Administrative token is shown below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–148 | 02 | ADMIN-DATA | PIC X(148) |

# Token TB Account Token—Binary Format

The fields in the binary format Account token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–80 | | ACCT-TKN | |
| 1–2 | 02 | ACCT-CNT<br>A count indicating the number of occurrences of account information in this token. | TYPE BINARY 16 |
| 3–80 | 02 | ACCT<br>The following fields contain account information for a maximum of three accounts.<br>The length of each ACCT occurrence is 26. | OCCURS 0 TO 3 TIMES DEPENDING ON ACCT-CNT |
| | 04 | IND<br>A code indicating the account number with which this PBF information is associated. Valid values are as follows:<br>C = Credit line/backup account<br>F = *From* account<br>T = *To* account | PIC X(1) |
| | 04 | STAT<br>A code indicating the current status of the application account. Valid values are as follows:<br>0, A, B, C = Inactive account<br>1, D, E, F, G, H, I = Open<br>2, J, K, L = Restricted to deposits<br>3, M, N, O, P, Q, R = Open primary account<br>4, S, T, U = Restricted primary account—deposits and inquiries only<br>9, V, W, X, Y, Z = Closed | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | The value in this field is taken from the ACCT-STAT field in the Base segment of the PBF. | |
| | 04 | LEDG-BAL | TYPE BINARY 64 SIGNED |
| | | The ledger balance for the account. | |
| | 04 | AVAIL-BAL | TYPE BINARY 64 SIGNED |
| | | The available balance for the account. | |
| | 04 | TTL-PRE-AUTH-HLD-AMT | TYPE BINARY 64 SIGNED |
| | | The total amount of all preauthorization holds found in the Preauthorized Holds segment of the PBF. | |

# Token TB Account Token—ASCII Format

The fields in the ASCII format Account token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–182 | | ACCT-TKNX | |
| 1–2 | 02 | ACCT-CNT | PIC 9(2) |
| 3–182 | 02 | ACCT | OCCURS 0 TO 3 TIMES DEPENDING ON ACCT-CNT |
| | 04 | IND | PIC X(1) |
| | 04 | STAT | PIC X(1) |
| | 04 | USER-FLD | PIC X(1) |
| | 04 | LEDG-BAL | PIC X(19) |
| | 04 | AVAIL-BAL | PIC X(19) |
| | 04 | TTL-PRE-AUTH-HLD-AMT | PIC X(19) |

# Token TC Override Token—Binary Format

The fields in the binary format Override token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–80 | | OVRRD-TKN<br>This token contains override processing information. It can appear in the following types of messages:<br>• Financial Transaction Response (0210)<br>• Financial Transaction Advice (0220)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry/Update Advice (0320) | |
| 1–2 | 02 | OVRRD-CNT<br>A count indicating the number of occurrences of override information in this token. | TYPE BINARY 16 |
| 3–80 | 02 | OVRRD<br>The length of each OVRRD occurrence is 26 positions. | OCCURS 0 TO 3 TIMES DEPENDING ON OVRRD-CNT |
| | 04 | ACCT-IND<br>A code indicating the account number with which this override information is associated. Valid values are as follows:<br>C = Credit/Backup Line Account<br>F = *From* account<br>T = *To* account<br>Any response codes returned in this token that are not related to a particular account (for example, card or PIN related response codes) are included with the response codes for the first account associated with the transaction. That is, if the transaction includes a *from* account, non-account-related response codes are included with the response codes for the | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | *from* account. If the transaction does not include a *from* account, non-account-related response codes are included with the response codes for the *to* account. | |
| | 04 | USER-FLD1 | PIC X(1) |
| | 04 | TLR-RESP-CDE-BIT-MAP | TYPE BINARY 64 SIGNED |
| | | Indicates the presence of a response code that can be overridden by a teller. | |
| | | The Authorization process initializes this field to zeros before setting the appropriate bits to 1. | |
| | 04 | TLR-RESP-CDE-BIT-MAP-I REDEFINES TLR-RESP-CDE-BIT-MAP | OCCURS 4 TIMES TYPE BINARY 16 SIGNED |
| | 04 | SPRVSR-RESP-CDE-BIT-MAP | TYPE BINARY 64 SIGNED |
| | | The presence of a response code that can be overridden by a supervisor. | |
| | | The Authorization process initializes this field to zeros before setting the appropriate bits to 1. | |
| | 04 | SPRVSR-RESP-CDE-BIT-MAP-I REDEFINES SPRVSR-RESP-CDE-BIT-MAP | OCCURS 4 TIMES TYPE BINARY 16 SIGNED |
| | 04 | MNGR-RESP-CDE-BIT-MAP | TYPE BINARY 64 SIGNED |
| | | The presence of a response code that can be overridden by a manager. | |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| | | The Authorization process initializes this field to zeros before setting the appropriate bits to 1. | |
| | 04 | MNGR-RESP-CDE-BIT-MAP-I<br>REDEFINES MNGR-RESP-CDE-BIT-MAP<br>OCCURS 4 TIMES | TYPE BINARY 16 SIGNED |

# Token TC Override Token—ASCII Format

The fields in the ASCII format Override token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–152 | | OVRRD-TKNX | |
| 1–2 | 02 | OVRRD-CNT | PIC 9(2) |
| 3–152 | 02 | OVRRD | OCCURS 0 TO 3 TIMES DEPENDING ON OVRRD-CNT |
| | 04 | ACCT-IND | PIC X(1) |
| | 04 | USER-FLD1 | PIC X(1) |
| | 04 | TLR-RESP-CDE-BIT-MAP | PIC X(16) |
| | 04 | SPRVSR-RESP-CDE-BIT-MAP | PIC X(16) |
| | 04 | MNGR-RESP-CDE-BIT-MAP | PIC X(16) |

# Token TD PIN Token—Binary Format

The fields in the binary format PIN token are described below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–56 | | PIN-TKN<br>This token contains PIN processing information. It can appear in the following types of messages:<br>• Financial Transaction Request (0200)<br>• Financial Transaction Response (0210)<br>• Financial Transaction Advice (0220)<br>• File Inquiry/Update Request (0300)<br>• File Inquiry/Update Response (0310)<br>• File Inquiry Advice (0320)<br>• Financial Transaction Reversal Advice (0420)<br>• Financial Transaction Reversal Advice Response (0430) | |
| 1–16 | 02 | PIN<br>The Personal Identification Number (PIN) block in PIN/PAD or PIN/PAN format. This field can also contain the clear (unencrypted) PIN. | PIC X(16) |
| 17 | 02 | PIN-FRMT<br>The format of the PIN field. Valid values are as follows:<br>0 = Clear PIN<br>1 = PIN/PAN PIN Block<br>3 = PIN/PAD PIN Block | PIC 9(1) |
| 18 | 02 | PIN-PAD-CHAR<br>The PAD character utilized in the formation of the external PIN/PAD PIN block. Valid values are 0 through 9 and A through F. | PIC X(1) |

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 19–20 | 02 | PIN-SIZE | PIC 9(2) |
| | | Length of the PIN in digits.<br>If the PIN is encrypted, the PIN-SIZE field is set to 16.<br>If the PIN is not present, this field is set to 00. | |
| 21 | 02 | PIN-TRIES | PIC X(1) |
| | | The number of PIN tries against this card. This field is initialized by the Device Handler process and updated by the Authorization process on each unsuccessful PIN entry. Once the PIN has been verified, it is set to a value of Z by the Authorization process. | |
| 22 | 02 | USER-FLD1 | PIC X(1) |
| 23–24 | 02 | ANSI-OFST | TYPE BINARY 16 |
| | | For PIN/PAN PIN blocks, this field contains the starting position of the 12 PAN digits on the card. The first digit of the PAN has a positional value of 0. | |
| 25–40 | 02 | PIN-KEY | PIC X(16) |
| | | The key under which the PIN is encrypted.<br>If the PIN-ENCRYPT-TYP field in the TTDF contains 00, this field contains zeros.<br>If the PIN-ENCRYPT-TYP field in the TTDF contains 01 or 02, this field contains the clear version of the key.<br>If the PIN-ENCRYPT-TYP field in the TTDF contains 03 or 04, this field contains the encrypted version of the key. | |
| 41–56 | 02 | USER-KEY | PIC X(16) |
| | | The previous key used for encrypting PINs. If PIN verification fails using the key in the PIN-KEY field, verification is attempted with the key in this field. | |

# Token TD PIN Token—ASCII Format

The fields in the ASCII format PIN token are shown below. For descriptions of these fields, refer to the documentation of the binary format version of this token.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–56 | | PIN-TKNX | |
| 1–16 | 02 | PIN | PIC X(16) |
| 17 | 02 | PIN-FRMT | PIC 9(1) |
| 18 | 02 | PIN-PAD-CHAR | PIC X(1) |
| 19–20 | 02 | PIN-SIZE | PIC 9(2) |
| 21 | 02 | PIN-TRIES | PIC X(1) |
| 22 | 02 | USER-FLD1 | PIC X(1) |
| 23–24 | 02 | ANSI-OFST | PIC 9(2) |
| 25–40 | 02 | PIN-KEY | PIC X(16) |
| 41–56 | 02 | USER-KEY | PIC X(16) |

# Token TE Native Message Token

The Native Message token contains the native message sent from the teller device. The Native Message token is not used by the BASE24-teller product for processing, but can be passed in the internal and external message, logged to the TTLF, or extracted from the TTLF. The basic structure of the Native Message token is shown below.

| Position | Level | Field Name and Description | Data Type |
|---|---|---|---|
| 1–148 | 02 | NATIVE-DATA | PIC X(148) |

# A: Token Impacts on Log File Disk Space Requirements

BASE24 users can indicate which tokens should be logged to the BASE24 transaction log files—the BASE24-atm Transaction Log File (TLF), POS Transaction Log File (PTLF), Teller Transaction Log File (TTLF), ITS Transaction Log File (ITLF), and Interchange Log Files (ILFs).

When deciding whether to log a token to a transaction log file, institutions should consider how adding that token will affect the length of the log record and how that added length will affect disk space requirements for the transaction log file. If logging a specific token to a transaction log file increases the length of the log file record so that fewer records can fit in a HP NonStop data block, more data blocks (that is, more disk space) will be required to hold the same number of records. This will in turn increase the institution's disk space requirements for the log file.

This appendix describes how users can assess the impacts of logging additional token data, and provides examples of how adding token data can affect disk space requirements.

# Information Needed to Determine Log File Impacts

To determine the disk space impacts of logging a token to a transaction log file, you need to have the following information:

- The format being used for the file. Beginning with the D46.00 release, HP NonStop supports a new larger Enscribe disk file partition format called format 2. Files with format 2 partitions are often referred to as *big files*. Files with format 2 partitions can contain much more data than the existing format 1 files. With format 2 files, as the disk size increases, partitions can increase to 1 terabyte. Currently, format 2 files are limited only by the physical disk size. BASE24 products support format 2 for the following entry-sequenced log files: Interchange Log File (ILF), POS Transaction Log File (PTLF), Transaction Log File (TLF), and ITS Transaction Log File (ITLF). Format 1, with a 2-gigabyte limit, continues to be supported by BASE24 products for these files and is the default.
- The block size being used for the file. The maximum HP NonStop data block for Enscribe files is 4096 bytes. This maximum block size is recommended for all BASE24 transaction log files. Of the 4096 bytes for transaction log file blocks, 22 bytes per block are reserved for HP NonStop file system use for format 1 files, while 44 bytes per block are reserved for HP NonStop file system use for format 2 files.

  **Note:** Log files are entry-sequenced, which is why 22 or 44 bytes are reserved for system use. Different file types require different numbers of reserved bytes.
- The current record length and number of records stored in each block. The HP NonStop file system uses an additional 2 bytes per record for format 1 records, while the HP NonStop file system uses an additional 4 bytes per record for format 2 files.
- The length of the token data that you want to add. The first token logged to a transaction log file requires the length of the token data plus 12 bytes, to log the Header token and the token header. Each subsequent token requires the length of the token data plus 6 bytes for the token header.

Once you have the above information, you can use it to determine whether logging tokens will affect the amount of disk space required for a particular log file. Some examples are provided on the following page to illustrate how adding token data can affect disk space requirements.

# How Logging Tokens Can Affect Disk Space Requirements

The following examples illustrate how logging tokens can affect the disk space requirements for a particular log file.

## Example 1

Assume you have a format 1 transaction log file with a maximum block size of 4096 bytes and a record length of 2000 bytes with no token data. Given these assumptions, you would currently log two records per block, and use 4026 bytes of the block (22 bytes for file system use and 2000 bytes plus 2 bytes for each of the records).

Now assume you have a token that is 20 bytes long, and you want to know the impact of logging the token. Since the record currently does not include token data, the actual length added to each record is 32 bytes—6 bytes for the Header token, 6 bytes for the token header, and 20 bytes for the token data. The new length of each log file record would be 2032 bytes. This length would still allow two records per block, using 4090 bytes of the block (22 bytes for file system use and 2032 bytes plus 2 bytes for each of the records), with no impact to the disk space required.

However, if the token data were 30 bytes long, the adjusted record size would be 2042 bytes. This record length would allow only one record to be logged per block, using 2066 bytes of the block (22 bytes for file system use and 2042 bytes plus 2 bytes for the record), which would double the amount of disk space required to hold the same number of records.

## Example 2

Assume you have a format 2 transaction log file with the maximum block size of 4096 bytes and a record length of 800 bytes. Given these assumptions, you would currently log five records per block and use 4064 bytes of the block (44 bytes for file system use and 800 bytes plus 4 bytes for each of the records).

You could add tokens with a total length of 6 bytes (including the Header token, if required, and a token header for each token) without impacting the disk requirements—with 6 bytes of token data per record, five records would use 4094 bytes of the block (44 bytes for file system use and 806 bytes plus 4 bytes for each of the records).

However, if you added 8 bytes of token data (or more, up to 208 bytes), five records would no longer fit in a block—with 8 bytes of token data per record, the length required for five records would be 4104 bytes (44 bytes for file system use and 808 bytes plus 4 bytes for each of the records). The number of records per block would be decreased to four, which would in turn increase the number of blocks required to log the same number of records. In this case, your disk space requirements would be increased by 25 percent (e.g., the number of blocks required to log 40 records would increase from eight to ten).

# Index


## A

Account Qualifier token, 5-13
Account token
  ASCII format, 8-46
  binary format, 8-44
ACH Debit token, 7-21
ACI Proactive Risk Manager token, 5-20
Acquirer Routing token, 5-211
Additional Hopper token
  ASCII format, 6-18
  binary format, 6-16
Address Verification token, 7-4
Administrative token, 8-43
Alternate Merchant ID token, 7-22
American Express Additional Data token, 7-123
American Express Private Use Data Token, 7-130
American Express token
  ASCII format, 7-19
  binary format, 7-11
ATM Balances token
  ASCII format, 6-36
  binary format, 6-33
ATM Data 1 token, 6-42
ATM Flag 1 token, 6-19
ATM Interchange Compliance token, 6-22
ATM Preferred Transaction Token, 6-57
ATM Preferred Transaction token
  ASCII format, 6-57
  binary format, 6-55
Authentication Data token, 7-74
Authorization processes, token usage, 2-5
Auto Substantiation Data token, 7-131
Automated Clearing House (ACH) Debit token, 7-21

## B

Bag Deposit token, 6-25
BASE24-atm Release 5.0 token, 6-4
BASE24-pos Release 5.0 token, 7-5
BASE24-pos Release 5.1 token, 7-34
BASE24-telebanking, and tokens, 1-7
BIC ISO Interface processes, token usage, 2-7
Bill Payment Confirmation token, 6-41
Bill Payment Payee List token
  ASCII format, 6-40
  binary format, 6-38
BNA Multiple Currency token
  ASCII format, 6-67
  binary format, 6-65
Bulk Check Amount token
  ASCII format, 6-73, 6-76
  binary format, 6-72
Bulk Check Disposition token
  ASCII format, 6-78
  binary format, 6-77
Bulk Check MICR token
  ASCII format, 6-71
  binary format, 6-70
Bulk Check SSB token
  binary format, 6-74

## C

CAF Inquiry token
  ASCII format, 8-11
  binary format, 8-6
CAF Update token, 8-12
Cardholder Postal Code token, 5-19
Cardholder Serial Number token, 7-66
Cash Acceptor Terminal Settlement token, 6-37
Certificate token, 7-59
Check Bundle token
  ASCII format, 6-69
  binary format, 6-68
Check Callback token, 7-33
Check Guarantee/Verification 2 token, 7-31
Check Guarantee/Verification token, 7-9
COBTKN, 3-4
Completion Required token, 5-245
Credit Line token
  ASCII format, 5-12
  binary format, 5-11
Custom Response Code token, 6-61
Customer Name token
  ASCII format, 5-9
  binary format, 5-8

## D

Data Encryption Key token, 5-239
Device Handler processes, token usage, 2-4
Diebold BNA Counts token
  ASCII format, 6-60
  binary format, 6-58, 6-62, 6-65
Digital Wallet token, 7-139
Disk space, token impacts on log files, A-1
DUKPT Data token, 7-70
Dynamic Currency Conversion Processing token
  ASCII format, 5-287
  binary format, 5-283
Dynamic Currency Conversion Status token, 5-281

## E

EBT Available Balance token
  ASCII format, 7-145
  binary format, 7-143
EBT Voucher token, 7-142
E-commerce Additional Data token, 7-135
EMV Discretionary Data token
  ASCII format, 5-183
  binary format, 5-174
EMV Issuer Script Results token, 5-231
EMV Request Data token
  ASCII format, 5-172
  binary format, 5-130
EMV Response Data token
  ASCII format, 5-202
  binary format, 5-194
EMV Script Data token
  ASCII format, 5-204
  binary format, 5-203
EMV Status token
  ASCII format, 5-193
  binary format, 5-184
EMV Supplementary Data token
  ASCII format, 5-289
  binary format, 5-288
Encrypted Balance token
  ASCII format, 5-241
  binary format, 5-240
Enhanced Reversal Routing token, 7-108
External message
  specifying tokens sent to the co-network, 4-40
  specifying tokens sent to the host, 4-35

## F

Financial token
  ASCII format, 8-5
  binary format, 8-2
Format 1 files, A-2
Format 2 files, A-2

## G

Gateway Info token, 5-278
Generic Data token, 5-295

## H

Header token
  ASCII format, 5-5
  binary format, 5-4
  functional description, 1-9
Healthcare Service token
  ASCII format, 7-129
  binary format, 7-127
Healthcare token, 7-97
Healthcare/Transit token
  ASCII format, 7-126
  binary format, 7-124
Hold token
  ASCII format, 6-54
  binary format, 6-53

## I

Increased Optional Data token, 7-64
Industry Data token, 7-91
Installment Payment Data token, 7-136
Integrated Authorization Server processes, token usage, 2-5
Interchange Compliance token, 7-29
Interchange Interface processes, token usage, 2-6
Interchange Log File (ILF)
  specifying tokens extracted from the, 4-20
  specifying tokens logged to the, 4-7
Interface FIID
  ASCII format, 5-105
  binary format, 5-28
Interface version ID
  ASCII format, 5-105
  binary format, 5-28
Interim Statement Processing token
  ASCII format, 6-49
  binary format, 6-48
Interim Statement/Passbook Data token, 6-45
Internal Transaction Data (ITD), 1-7
International Address Verification Service Data token, 7-76
Intra Country Data token, 5-277
Inventory Voucher token
  ASCII format, 5-276
  binary format, 5-274
ISO Host Interface processes, token usage, 2-5
Issuer Fee Rebate token
  ASCII format, 5-25
  binary format, 5-23

ITS Transaction Log File (ITLF)
specifying tokens extracted from the, 4-25
specifying tokens logged to the, 4-10

## L

Log file perusal subsystems, token usage, 2-9
Log files, token impacts on disk space, A-1

## M

Merchant Banking Center Settlement token
ASCII format, 6-32
binary format, 6-29
Merchant Serial Number token, 7-67
Messages
Interchange Interface processes, 1-3
internal message structure, 1-4
tokens, 1-5
translating between internal and external formats, 1-2
MHI Additional Data token
ASCII format, 7-69
binary format, 7-68
MICR Data token, 5-10
Migration ATM Data1 token, 5-269
Migration Customer Data token, 5-272
Migration EPS HISO token, 5-271
Migration POS Data1 token, 5-270
Migration process, token usage, 2-8
Money Exchange token
ASCII format, 6-28
binary format, 6-26
Multiple Account Inquiry token
ASCII format, 6-44
binary format, 6-43
Multiple Account token, 6-23
Multiple Currency token
ASCII format, 5-218
binary format, 5-215
Multiple Logical Network token
ASCII format, 5-234
binary format, 5-233

## N

Native Message token, 8-54
NBF token
ASCII format, 8-18
binary format, 8-14
newlink CH binary, 7-77
Non-Currency Dispense token, 6-21

## O

Original Currency Release 6.0 token
ASCII format, 5-221
binary format, 5-219
Override token
ASCII format, 8-50
binary format, 8-47

## P

PAN Mapping token, 5-293, 5-294
Passbook Processing token
ASCII format, 6-52
binary format, 6-50
PBF Inquiry token
ASCII format, 8-24
binary format, 8-19
PBF Update token, 8-25
Person-to-Person Transaction 2 token, 5-291
Person-to-Person Transaction token, 5-242
PIN Change token, 6-7
PIN token
ASCII format, 8-53
binary format, 8-51
Point of Service Data token, 7-60
POS Balances token
ASCII format, 7-73
binary format, 7-71
POS Data 2 token
ASCII format, 7-134
binary format, 7-132
POS Data1 token
ASCII format, 7-83
binary format, 7-77
POS Merchant token, 7-84
POS Split Transaction Routing token, 7-107
POS Transaction Log File (PTLF)
specifying tokens extracted from the, 4-23
specifying tokens logged to the, 4-8
Pre-Pay Generic Receipt token
ASCII format, 5-213
binary format, 5-212
Pre-Pay Merchant token
ASCII format, 7-90
binary format, 7-88
Pre-Pay Online Receipt token
ASCII format, 5-264
binary format, 5-263
Pre-Pay Original Data token, 5-265
Pre-Pay Receipt token
ASCII format, 5-223
binary format, 5-222
Pre-Pay Response token
ASCII format, 5-255
binary format, 5-253

Pre-Pay Selection token
ASCII format, 5-259
binary format, 5-256
Pre-Pay Switch token
ASCII format, 5-252
binary format, 5-250
Pre-Pay Top-Up token, 5-227
Pre-Pay Voucher Receipt token
ASCII format, 5-262
binary format, 5-260
PRM Real Time token, 5-26
PS2000 ATM token, 6-15
Purchase Card and Fleet Card token, 7-40

## R

Refresh process, token usage, 2-9
Release 5.0 token
BASE24-atm, 6-4
BASE24-pos, 7-5
Report programs, token usage, 2-9
Reversal Date and Time token, 5-225
Reward Program token, 7-106

## S

Self-Service Banking Base token
ASCII format, 6-10
binary format, 6-9
Self-Service Banking Check Terminal Settlement token
ASCII format, 6-14
binary format, 6-13
Self-Service Banking Check token, 6-11
Settlement processes, token usage, 2-8
Shared BNA Counts token
ASCII format, 6-64
binary format, 6-62
SPF Inquiry token
ASCII format, 8-30
binary format, 8-26
SPF Update token
ASCII format, 8-33
binary format, 8-31
Split Transaction Routing token, 5-246
Statement Print token, 6-3
Station ID token, 7-39
Stored Value token
ASCII format, 7-149
binary format, 7-146
Super Extract process, token usage, 2-8
Surcharge Data token
ASCII format, 5-18
binary format, 5-15
Switch Common Data Token, 5-266
Switch token, acquirer
ASCII format, 5-105
binary format, 5-28
Switch token, interface FIID
ASCII format, 5-105
binary format, 5-28
Switch token, interface version ID
ASCII format, 5-105
binary format, 5-28
Switch token, issuer
ASCII format, 5-105
binary format, 5-28

## T

Teller Transaction Log File (TTLF)
default TKN records for extracts, 4-29
default TKN records for logging, 4-13
specifying tokens extracted from the, 4-26
specifying tokens logged to the, 4-11
TKN^ADD^INFO utility
general information, 2-24
parameters, 2-24
processing, 2-28
status codes, 2-28
TKN^DEL^INFO utility
general information, 2-55
parameters, 2-55
processing, 2-58
status codes, 2-58
TKN^GET^IDS utility
general information, 2-18
parameters, 2-18
processing, 2-22
status codes, 2-21
TKN^GET^INFO utility
general information, 2-13
parameters, 2-14
processing, 2-16
status codes, 2-16
TKN^LOG^INFO utility
general information, 2-49
parameters, 2-49
processing, 2-52
status codes, 2-51
TKN^MAIN^CONVERT utility
general information, 2-61
parameters, 2-62
processing, 2-66
status codes, 2-65
TKN^SORT^INFO utility
general information, 2-37
parameters, 2-37
processing, 2-45
status codes, 2-44

TKN^UPDT^INFO utility
general information, 2-31
parameters, 2-31
processing, 2-35
status codes, 2-34
TLF token
ASCII format, 5-208
binary format, 5-205
Token AS Shared BNA Counts Token, 6-62
Token AT BNA Multiple Currency Token, 6-65
Token buffers, 2-11
Token File (TKN)
adding extract records, 4-31
adding logging records, 4-15
default BASE24-teller extract records, 4-29
default BASE24-teller logging records, 4-13
updating extract records, 4-33
updating logging records, 4-18
Token header
ASCII format, 5-7
binary format, 5-6
functional description, 1-10
Tokens
adding to a buffer, 2-24
converting to another data format, 2-61
defining new tokens, 3-2
definition, 1-5
deleting, 2-55
determining which are present, 2-13
example of message processing, 1-19
impacts on log file disk space requirements, A-1
in buffers, 2-11
limitations, 2-3
locations in BASE24, 1-6
moving to another buffer, 2-37
process usage, 2-4
specifying for extract, 4-19
specifying token logging, 4-3
specifying tokens sent in external messages, 4-35
storing in files, 2-2
updating information in, 2-31
utilities, 2-12
when ASCII format is used, 1-18
when binary format is used, 1-18
writing to a file, 2-49

Tokens, by ID
token 01, 7-4
token 02, 6-3
token 03, 6-4
token 04, 7-5
token 05, 7-9
token 06, 6-7
token 07, 6-9, 6-10
token 08, 5-8, 5-9
token 10, 7-11, 7-19
token 11, 7-21
token 12, 5-10
token 13, 5-11, 5-12
token 14, 6-11
token 15, 6-13, 6-14
token 16, 7-22
token 17, 7-23
token 18, 5-13
token 19, 7-26
token 20, 7-29
token 21, 6-15
token 22, 6-16, 6-18
token 23, 5-14
token 24, 6-19
token 25, 5-15, 5-18
token 27, 5-19
token 28, 5-20
token 29, 7-31
token 30, 5-23, 5-25
token 31, 7-33
token 32, 5-26
token A5, 6-21
token A6, 6-22
token A7, 6-23
token A8, 6-25
token A9, 6-26, 6-28
token AA, 6-29, 6-32
token AB, 6-33, 6-36
token AD, 6-37
token AE, 6-38, 6-40
token AF, 6-41
token AG, 6-42
token AH, 6-43, 6-44
token AI, 6-45
token AJ, 6-48, 6-49
token AK, 6-50, 6-52
token AL, 6-53, 6-54
token AM, 6-55, 6-57
token AO, 6-58, 6-60
token AR, 6-61
token AS, 6-62, 6-64
token AT, 6-65, 6-67
token AU, 6-68, 6-69
token AV, 6-71
token AW, 6-73
token AX, 6-76
token AY, 6-78
token B0, 5-28, 5-105
token B1, 5-28, 5-105
token B2, 5-130, 5-172
token B3, 5-174, 5-183
token B4, 5-184, 5-193
token B5, 5-194, 5-202
token B6, 5-203, 5-204
token B7, 5-205, 5-208
token B8, 5-209
token B9, 5-210
token BA, 5-211
token BB, 5-212
token BC, 5-214
token BD, 5-215, 5-218
token BE, 5-219, 5-221
token BF, 5-222, 5-223
token BG, 5-224
token BH, 5-225
token BI, 5-227
token BJ, 5-231
token BK, 5-233, 5-234
token BL, 5-235
token BM, 5-236
token BN, 5-239
token BO, 5-240, 5-241
token BP, 5-242
token BQ, 5-245
token BR, 5-246
token BS, 5-250, 5-252
token BT, 5-253, 5-255
token BU, 5-256, 5-259
token BV, 5-260, 5-262
token BW, 5-263, 5-264
token BX, 5-265
token BY, 5-266
token C0, 7-34
token C1, 7-39
token C2, 7-40
token C3, 7-59
token C4, 7-60
token C5, 7-64
token C6, 7-65
token C7, 7-66
token C8, 7-67
token C9, 7-68, 7-69
token CA, 7-70
token CB, 7-71, 7-73
token CE, 7-74
token CF, 7-76
token CH, 7-77, 7-83
token CI, 7-84
token CJ, 7-88, 7-90
token CK, 7-91
token CP, 7-97
token CQ, 7-106
token CR, 7-107
token CS, 7-108
token CT, 7-109, 7-113
token CU, 7-123
token CV, 7-124, 7-126
token CW, 7-127, 7-129
token CX, 7-130
token CY, 7-131
token CZ, 7-132, 7-134
token F1, 7-135
token F2, 7-136
token F3, 7-137, 7-139

token M1, 5-269
token M2, 5-270
token M4, 5-271
token M5, 5-272
token N8, 5-274, 5-276
token S0, 5-277
token S1, 5-278
token S2, 5-281
token S3, 5-283, 5-287
token S4, 5-288, 5-289
token S6, 5-290
token S7, 5-291
token S8, 5-293, 5-294
token SA, 5-295
token T0, 8-2, 8-5
token T1, 8-6, 8-11
token T2, 8-12
token T3, 8-14, 8-18
token T4, 8-19, 8-24
token T5, 8-25
token T6, 8-26, 8-30
token T7, 8-33
token T8, 8-34, 8-39
token T9, 8-40, 8-42
token TA, 8-43
token TB, 8-44, 8-46
token TC, 8-47, 8-50
token TD, 8-51, 8-53
token TE, 8-54
token U0, 7-142
token U1, 7-143, 7-145
token U2, 7-146, 7-149
Track 1 token, 5-14
Track 2 token, 5-290
Track 3 token, 5-224
Trans Stain XID token, 7-65
Transaction Description token, 5-210
Transaction Log File (TLF)
specifying tokens extracted from the, 4-21
specifying tokens logged to the, 4-4
Transaction Profile token, 5-209
Transaction Specific Data token using expanded datasets, 7-113
Transaction Specific Data token using Redefines, 7-109
Transaction Subtype token, 5-236
Transit Transaction token, 7-137
TSS Index token, 5-214

## V

Virtual Primary Account Number token, 5-235
Visa Payment Service 2000 Offline token, 7-26
Visa Payment Service 2000 token, 7-23

## W

WHFF Inquiry token
ASCII format, 8-39
binary format, 8-34
WHFF Update token
ASCII format, 8-42
binary format, 8-40

*ACI Worldwide, Inc.*