

Using EMV with Authorize.Net

Preview Guide

August 2015

Authorize.Net®

Authorize.Net LLC ("Authorize.Net") has made efforts to ensure the accuracy and completeness of the information in this document. However, Authorize.Net disclaims all representations, warranties and conditions, whether express or implied, arising by statute, operation of law, usage of trade, course of dealing or otherwise, with respect to the information contained herein. Authorize.Net assumes no liability to any party for any loss or damage, whether direct, indirect, incidental, consequential, special or exemplary, with respect to (a) the information; and/or (b) the evaluation, application or use of any product or service described herein.

Authorize.Net disclaims any and all representation that its products or services do not infringe upon any existing or future intellectual property rights. Authorize.Net owns and retains all right, title and interest in and to the Authorize.Net intellectual property, including without limitation, its patents, marks, copyrights and technology associated with the Authorize.Net services. No title or ownership of any of the foregoing is granted or otherwise transferred hereunder. Authorize.Net reserves the right to make changes to any information herein without further notice.

Authorize.Net Trademarks

Advanced Fraud Detection Suite™

Authorize.Net®

Authorize.Net Your Gateway to IP Transactions™

Authorize.Net Verified Merchant Seal™

Automated Recurring Billing™

eCheck.Net®

Contents

Recent Revisions to This Document	4
About This Guide	5
Audience and Purpose	5
Conventions	5
Note, Important, and Warning Statements	5
Text and Command Conventions	6
Related Documents	6
Customer Support	6
Implementing EMV	7
Transaction Requests	7
Transaction Request Using an Encrypted Card Reader	7
Transaction Request Using an Unencrypted Card Reader	10
Fallback Payment Data	12
Transaction Responses	14
Successful Response	15
Unsuccessful Response	17
Error Codes	17
Tags	18

Recent Revisions to This Document

Release	Changes
August 2015	First release of this document.

About This Guide

This document enables developers to add EMV payment functionality to a checkout flow using the Authorize.Net XML API.

Audience and Purpose

The audience for this guide are developers who are familiar with how the Authorize.Net payment API works. The purpose is to give developers the information they need to add EMV payment functionality to an existing Authorize.Net API payment implementation.

Conventions

Note, Important, and Warning Statements



A *Note* contains helpful suggestions or references to material not contained in the document.



An *Important* statement contains information essential to successfully completing a task or learning a concept.



A *Warning* contains information or instructions, which, if not heeded, can result in a security risk, irreversible loss of data, or significant cost in time or revenue or both.

Text and Command Conventions

Convention	Usage
bold	■ Field and service names in text; for example: Include the ics_applications field. ■ Items that you are instructed to act upon; for example: Click Save.
<i>italic</i>	■ Filenames and pathnames. For example: Add the filter definition and mapping to your <i>web.xml</i> file. ■ Placeholder variables for which you supply particular values.
monospace	■ XML elements. ■ Code examples and samples. ■ Text that you enter in an API environment; for example: Set the davService_run field to <code>true</code>.

Related Documents

This document assumes a working implementation of the Authorize.Net XML API, also known as the AIM XML API. For more information, including API documentation, see

<http://developer.authorize.net/api/aim/>

Customer Support

For support information about any Authorize.Net service, visit the Support Center:

<https://support.authorize.net/authkb/index?page=home>

Implementing EMV

EMV is a technical standard for smart payment cards and for payment terminals and automated teller machines which can accept them. EMV cards are smart cards (also called chip cards or IC cards) which store their data on integrated circuits rather than magnetic stripes, although many EMV cards also have stripes for backward compatibility. This integration is currently supported using the Walker card reader.

Adding EMV functionality for an existing implementation of the Authorize.Net XML API requires a new subset of XML fields in the standard **createTransactionRequest** and **createTransactionResponse** functions.

Transaction Requests

Transaction requests can be made using either an encrypted card reader or an unencrypted card reader.

Transaction Request Using an Encrypted Card Reader

The table below describes the fields contained in an **createTransactionRequest** function, using an encrypted EMV reader.

Table 1 Elements in an EMV Transaction Using an Encrypted Card Reader

Element	Description	Format
merchantAuthentication	Required. Contains merchant authentication information.	

Table 1 Elements in an EMV Transaction Using an Encrypted Card Reader (Continued)

Element	Description	Format
name	Required. Merchant's unique API Login ID. The merchant API Login ID is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.	Up to 20 characters.
transactionKey	Required. Merchant's unique Transaction Key. The merchant Transaction Key is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.	16 characters.
transactionRequest	Transaction information. This element includes all of the fields that follow.	
transactionType	Type of transaction. If the value submitted does not match a supported value, the transaction is rejected. If this field is not submitted or the value is blank, the payment gateway will process the transaction as an authCaptureTransaction.	
amount	Required. Amount of the transaction. This is the total amount and must include tax, shipping, and any other charges. The amount can either be hard coded or posted to a script.	Up to 15 digits with a decimal point (no dollar symbol. For example, 8.95.
payment	This section includes payment information.	
- opaqueData	Contains dataDescriptor and dataValue .	
- dataDescriptor	Required. Meta data used to specify how the request should be processed. This should always be the same for EMV.	READER.WALKER.EMV.PAYMENT

Table 1 Elements in an EMV Transaction Using an Encrypted Card Reader (Continued)

Element	Description	Format
• dataValue	Required. Base-64 encoded data that contains encrypted payment data. The payment gateway expects the encrypted payment data and meta data for the encryption keys.	Unmodified encrypted data from the card reader.
■ retail	Contains information related to the retail environment.	
• marketType	2 for retail	Must be set to 2 for EMV.
• deviceType	Contains information about the device used to process the transaction information. 4 = Electronic Cash Register 5 = Personal Computer-Based Terminal 7 = Wireless POS	
• locationCode	Merchant-generated data relating to store locations.	Numeric, merchant input
• terminalId	Merchant-generated data relating to terminals within stores.	Numeric, merchant input

Example 1 Code Example for EMV Transaction Using an Encrypted Card Reader

```

<createTransactionRequest xmlns="AnetApi/xml/v1/schema/
AnetApiSchema.xsd">
  <merchantAuthentication>
    <name>merchantID</name>
    <transactionKey>TransactionKey</transactionKey>
  </merchantAuthentication>
  <refId>123456</refId>
  <transactionRequest>
    <transactionType>authCaptureTransaction</transactionType>
    <amount>5</amount>
    <payment>
      <opaqueData>
        <dataDescriptor>READER.WALKER.EMV.PAYMENT</dataDescriptor>
        <dataValue>eyJkYXRhIjoiQkRQTldTdE1tR2V3UVVXR2c0bzdfXC9qKzFjcTFU
NzhxeVU4NGjelJQUHVGC2ZwSFZzNDhvUEExDXC9rXC8xTU5kNDdrelwvtIQnVZVW5lV0RCTlNE
MnU2amJBRzMiLCJ2ZXJzaW9uWc2VHI4dz09In0=</dataValue>
      </opaqueData>
    </payment>
  </transactionRequest>
  <retail>
    <marketType>2</marketType>
    <deviceType>7</deviceType>
    <locationCode>00000123456</locationCode>
  </retail>
</createTransactionRequest>

```

```

        <terminalId>0000987654</terminalId>
      </retail>
    </transactionRequest>
  </createTransactionRequest>

```

Transaction Request Using an Unencrypted Card Reader

The table below describes the fields contained in a **createTransactionRequest** function using an unencrypted reader.

Table 2 Elements in an EMV Transaction Using an Unencrypted Card Reader

Element	Description	Format
merchantAuthentication	Required. Contains merchant authentication information.	
name	Required. Merchant's unique API Login ID. The merchant API Login ID is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.	Up to 20 characters.
transactionKey	Required. Merchant's unique Transaction Key. The merchant Transaction Key is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.	16 characters.
transactionRequest	Transaction information. This element includes all of the fields that follow.	
transactionType	Type of transaction. If the value submitted does not match a supported value, the transaction is rejected. If this field is not submitted or the value is blank, the payment gateway will process the transaction as an authCaptureTransaction .	

Table 2 Elements in an EMV Transaction Using an Unencrypted Card Reader (Continued)

Element	Description	Format
■ amount	Required. Amount of the transaction. This is the total amount and must include tax, shipping, and any other charges. The amount can either be hard coded or posted to a script.	Up to 15 digits with a decimal point (no dollar symbol. For example, 8.95.
■ payment	This section includes payment information.	
● emvData	Contains the tags element.	
● tags	Contains unmodified tag information sent by the reader.	
■ retail	Contains information related to the retail environment.	
● marketType	0 for ecommerce 1 for moto 2 for retail	Must be set to 2 for EMV.
● deviceType	Contains information about the device used to process the transaction information.	Must be set to 7 for EMV.
● locationCode	Merchant-generated data relating to store locations.	String
● terminalId	Merchant-generated data relating to terminals within stores.	String

Example 2 Code Sample of an EMV Transaction Using an Unencrypted Card Reader

```

<createTransactionRequest xmlns="AnetApi/xml/v1/schema/
AnetApiSchema.xsd">
  <merchantAuthentication>
    <name>merchantName</name>
    <transactionKey>transactionKey</transactionKey>
  </merchantAuthentication>
  <transactionRequest>
    <transactionType>authCaptureTransaction</transactionType>
    <amount>5</amount>
    <payment>
      <emvData>
        <tags>9F3403204000950500000000009F3704518823719F100706011103A
000009F26081E1756ED0E2134E29F36020015820200009C01009F1A0208409A030006219F
02060000000020005F2A0208409F0306000000000000</tags>
      </emvData>
    </payment>
    <retail>

```

```

    <marketType>2</marketType>
    <deviceType>7</deviceType>
    <locationCode>00000123456</locationCode>
    <terminalId>0000987654</terminalId>
  </retail>
</transactionRequest>
</createTransactionRequest>

```

Fallback Payment Data

Some terminals have the ability to use magstripe data if the EMV functionality fails. This is known as a fallback, and you can do this in the Authorize.Net API by using the **emvFallback** field.

Table 3 Elements in an EMV Transaction Using Fallback

Element	Description	Format
merchantAuthentication	Required. Contains merchant authentication information.	
name	Required. Merchant's unique API Login ID. The merchant API Login ID is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.	Up to 20 characters.
transactionKey	Required. Merchant's unique Transaction Key. The merchant Transaction Key is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.	16 characters.
transactionRequest	Transaction information. This element includes all of the fields that follow.	

Table 3 Elements in an EMV Transaction Using Fallback (Continued)

Element	Description	Format
■ transactionType	Type of transaction. If the value submitted does not match a supported value, the transaction is rejected. If this field is not submitted or the value is blank, the payment gateway will process the transaction as an authCaptureTransaction .	
■ amount	Required. Amount of the transaction. This is the total amount and must include tax, shipping, and any other charges. The amount can either be hard coded or posted to a script.	Up to 15 digits with a decimal point (no dollar symbol. For example, 8.95.
■ payment	This section includes payment information.	
● trackData	Contains the track1 element.	
● track1	Contains Track 1 data derived from the magnetic card reader. You can send track1 or track2 , but not both.	String
● track2	Contains Track 2 data derived from the magnetic card reader. You can send track1 or track2 , but not both.	String
● emvFallBack	Required. This field is a boolean flag that indicates whether the transaction is using an EMV fallback.	true or false
■ retail	Contains information related to the retail environment.	
● marketType	0 for ecommerce 1 for moto 2 for retail	Must be set to 2 for EMV.
● deviceType	Contains information about the device used to process the transaction information.	Must be set to 7 for EMV.

Example 3

```
<createTransactionRequest xmlns="AnetApi/xml/v1/schema/
AnetApiSchema.xsd">
```

```

<merchantAuthentication>
  <name>merchantName</name>
  <transactionKey>transactionKey</transactionKey>
</merchantAuthentication>
<transactionRequest>
  <transactionType>authCaptureTransaction</transactionType>
  <amount>5</amount>
  <payment>
    <trackData>
      <track1>%BXXXXXXXXXXXX5675^3/
TEST.^15121010000000439000000?</track1>
    </trackData>
    <emvFallback>true</emvFallback>
  </payment>
  <retail>
    <marketType>2</marketType>
    <deviceType>1</deviceType>
  </retail>
</transactionRequest>
</createTransactionRequest>

```

Transaction Responses

The complete list of tags is returned for both successful and unsuccessful transactions. However, we recommend as a best practice that you display only the following five tags for a successful response:

- AID
- TVR
- IAD
- TSI
- ARC

We recommend that you send the full list of tags for an unsuccessful response.

Successful Response

The table and code sample below show the fields that you receive as part of a successful EMV transaction response. These elements are contained within the `createTransactionResponse` reply.

Table 4 Successful EMV Transaction Response

Element	Description	Format
refId	Merchant-assigned reference ID for the request. If included in the request, this value is included in the response. This feature might be especially useful for multi-threaded applications.	Up to 20 characters.
messages	This section contains information about the results of the request.	
■ resultCode	Contains additional information about the status of the request.	Ok or Error.
■ message	Contains specific message information.	
● code	Code number for message.	Examples: I00001 E000001
● text	Text for the error message.	
transactionResponse	Contains response information.	
■ responseCode	Overall status of the transaction.	1 = Approved 2 = Declined 3 = Error 4 = Held for Review
■ authCode	Authorization or approval code.	6 characters.
■ transId	The payment gateway assigned identification number for transaction. This value must be used for any follow-on transactions such as a credit, prior auth capture, or void.	When testRequest is set to a positive response, or when Test Mode is enabled on the payment gateway, this value will be 0.
■ transHash	Payment gateway-generated MD5 hash value that can be used to authenticate the transaction response. Because transaction responses are returned using an SSL connection, this feature is not necessary for AIM.	Alphanumeric.
■ testRequest	Indicates whether or not to treat this request as a test transaction.	true, false, 1, 0

Table 4 Successful EMV Transaction Response

Element	Description	Format
■ accountNumber	Card number. All but the last 4 digits are masked.	
■ accountType	Card type. For example, Visa or MasterCard.	
■ emvResponse	Contains EMV response information.	
• mode		
• AID	Required by FDC and TSYS. Application Identifier Card. Identifies the application as described in ISO/IEC 7816-5.	B, Length 5-16
• TVR	Terminal Verification Results. Status of the different functions as seen from the terminal.	B, Length 5
• IAD	Issuer Application Data. Contains proprietary application data for transmission to the issuer in an online transaction.	B, Length <=32
• TSI	Transaction Status Indicator. Indicates the functions performed in a transaction.	B, Length 2
• ARC	Application Response Code. Code that defines the disposition of a message.	AN 2, Length 2
• tags	Contains tags. For more information about tags, see "Tags," page 18 .	
• tag	Contains tag information.	
• name	Name of the tag.	
• value	Value of the tag.	
■ code	Response code that represents the status.	
■ description	Text description of the status.	

Example 4 Successful EMV Transaction Response

```

<createTransactionResponse>
  <refId>123456</refId>
  <messages>
    <resultCode>Ok</resultCode>
    <message>
      <code>I00001</code>
      <text>Successful.</text>
    </message>
  </messages>
  <transactionResponse>
    <responseCode>1</responseCode>
    <authCode>UGELQC</authCode>
    <transId>2148061808</transId>
  </transactionResponse>
</createTransactionResponse>

```

```

<transHash>0B428D8A928AAC61121AF2F6EAC5FF3F</transHash>
<testRequest>0</testRequest>
<accountNumber>XXXX0015</accountNumber>
<accountType>MasterCard</accountType>
<emvResponse>
  <Mode>Issuer</Mode>
  <AID>A0000000031010</AID>
  <TVR>0000008000</TVR>
  <IAD>06010A03A40002</IAD>
  <TSI>F800</TSI>
  <ARC>00</ARC>
  <tags>
    <tag>
      <name>5F34</name>
      <value>12</value>
    </tag>
    <tag>
      <name>95</name>
      <value>0010008000</value>
    </tag>
  </tags>
</emvResponse>
<message>
  <code>1</code>
  <description>This transaction has been approved.</description>
</message>
</transactionResponse>
</createTransactionResponse>

```

Unsuccessful Response

An unsuccessful transaction response is the same as a successful response except that it contains a block of XML that begins with `<errors>`, which looks like this:

```

<errors>
  <error>
    <errorCode>A numeric value</errorCode>
    <errorText>A description of the error.</errorText>
  </error>
</errors>

```

Error Codes

Error codes will be developed soon.

Tags

EMV tags uniquely identify data objects from the list of tags defined by the EMV standard. The following EMV tags are used by Authorize.Net.

Table 5 EMV Tags

Tag	Description	Format	Required By
50	Application Label	FDC	Walker
	Mnemonic associated with the AID according to ISO/IEC 7816-5.	AN 1-16, Length 1-16	FDC
		TSYS	TSYS
		AN 1-16, Length 1-16	
4F	Application Identifier Card (ICC)	FDC	Walker
	Identifies the application as described in ISO/IEC 7816-5.	B, Length 5-16	FDC
		TSYS	TSYS
		B, Length 5-16	
5A	Application PAN	TSYS	Walker
	Valid cardholder account number.	CN <=19, Length <=10	TSYS
	Truncate any trailing F and any mask.		
5F2A	Transaction Currency Code	Normalized	Walker
	Indicates the currency code of the transaction according to ISO 4217.	Numeric (4)	GPN
		GPN	FDC
		Fixed length 3 N, 4 bit BCD, 2 bytes	TSYS
		FDC	
		N 3, Length 2	
		TSYS	
		N 3, Length 2	
5F34	PAN Sequence Number	Normalized	Walker
	Identifies and differentiates cards with the same PAN.	Numeric (2)	FDC
		FDC	TSYS
	Blank if not present.	N 2, Length 1	
		TSYS	
		N 2, Length 1	

Table 5 EMV Tags (Continued)

Tag	Description	Format	Required By
82	Application Interchange Profile Indicates the capabilities of the card to support specific functions in the application.	Normalized	Walker
		Numeric (4)	GPN
		GPN	FDC
		Fixed length 16 bit string; 2 bytes	TSYS
		FDC	
		B, Length 2	
		TSYS	
84	Dedicated File Name Identifies the name of the DF as described in ISO/IEC 7816-4. Usually the same as the AID.	Normalized	Walker
		Numeric (14)	GPN
		GPN	
		Variable length 5-16 bytes	
95	Terminal Verification Results Status of the different functions as seen from the terminal.	Normalized	Walker
		Numeric (8)	GPN
		GPN	FDC
		Fixed length 40 bit string; 5 bytes	TSYS
		FDC	
		B, Length 5	
		TSYS	
8A	Application Response Code Code that defines the disposition of a message.	FDC	Walker
		AN 2, Length 2	FDC
		TSYS	TSYS
		AN 2, Length 2	
9A	Transaction Date Local date that the transaction was authorized.	Normalized	Walker
		Numeric (6)	GPN
		GPN	FDC
		Fixed length 6N, 4 bit BCD; 3 bytes	TSYS
		FDC	
		N 6, Length 3	
		TSYS	
		N 6, Length 3	

Table 5 EMV Tags (Continued)

Tag	Description	Format	Required By
9B	Transaction Status Indicator Indicates the functions performed in a transaction.	FDC	Walker
		B, Length 2	FDC
		TSYS	TSYS
		B, Length 2	
9F21	Transaction Time Local time that the transaction was authorized.	FDC	FDC
		N 6, Length 3	
		TSYS	
		N 6, Length 3	
9C	Transaction Type Indicates the type of financial transaction, represented by the first two digits of ISO 8583: 1987 Processing Code.	Normalized	Walker
		Numeric (2)	GPN
		GPN	FDC
		Fixed length 2N, 4 bit BCD (unsigned packed); 1 byte	TSYS
		FDC	
		N 2, Length 1	
		TSYS	
		N 2, Length 1	
9F1A	Terminal Country Code Indicates the country of the terminal, represented according to ISO 3166.	Normalized	Walker
		Numeric (4)	GPN
		GPN	FDC
		Fixed length 2N, 4 bit BCD; 2 bytes	TSYS
		FDC	
		N 3, Length 2	
		TSYS	
		N 3, Length 2	
9F1E	Interface Device Serial Number Unique and permanent serial number assigned to the IFD by the manufacturer.	Normalized	Walker
		Numeric (10)	GPN
		GPN	
		Fixed length 8 AN; 8 bytes	

Table 5 EMV Tags (Continued)

Tag	Description	Format	Required By
9F02	Amount Authorized Authorized amount of the transaction (excluding adjustments).	Normalized	Walker
		Numeric (12)	GPN
		GPN	FDC
		Fixed length 12 N, 4 bit BCD (unsigned packed); 6 bytes	TSYS
		FDC	
		N 12 Length 6	
		TSYS N 12, Length 12	
9F03	Other Amount Secondary amount associated with the transaction representing a cashback amount.	Normalized	Walker
		Numeric (9)	GPN
		GPN	FDC
		Fixed length 12 N, 4 bit BCD (unsigned packed); 6 bytes	TSYS
		FDC	
		N 12 Length 6	
		TSYS B, Length 4	
9F07	Application Usage Control Indicates issuer's specified restrictions on the geographic usage and services allowed for the application.	FDC	Walker
		B, Length 2	FDC
		TSYS	TSYS
		B, Length 2	
9F09	Application Version Number (Terminal) Version number assigned by the payment system for the application.	Normalized	Walker
		Numeric (4)	GPN
		GPN	
		Fixed Length 4 hex digits, 2 byte binary value	
9F0D	Issuer Action Code - Default Specifies the issuer's conditions that cause a transaction to be rejected if it might have been approved online, but the terminal is unable to process the transaction online.	FDC	Walker
		B, Length 5	FDC
		TSYS	TSYS
		B, Length 5	

Table 5 EMV Tags (Continued)

Tag	Description	Format	Required By
9F0E	Issuer Action Code - Denial Specifies the issuer's conditions that cause the denial of a transaction without attempt to go online.	FDC	Walker
		B, Length 5	FDC
		TSYS	TSYS
		B, Length 5	
9F0F	Issuer Action Code - Online Specifies the issuer's conditions that cause a transaction to be transmitted online.	FDC	Walker
		B, Length 5	FDC
		TSYS	TSYS
		B, Length 5	
9F10	Issuer Application Data Contrains proprietary application data for transmission to the issuer in an online transaction.	Normalized	Walker
		Numeric (28)	GPN
		GPN	TSYS
		Variable length 1 byte binary + up to 32 bytes; maximum 33 bytes	
		TSYS	
		B, Length <=32	
9F12	Application Preferred Name Preferred mnemonic associated with the AID.	Normalized	Walker
		Numeric (16)	GPN
		FDC	FDC
		AN 1-16, Length 1-16	TSYS
		TSYS	
		AN 1-16, Length 1-16	
9F26	Application Cryptogram Cryptogram returned by the ICC in response of the GENERATE AC command.	Normalized	Walker
		Numeric (12)	GPN
		GPN	FDC
		Fixed Length 16 Hex Digits; 8 bytes	TSYS
		FDC	
		B, Length 8	
		TSYS	
		B, Length 8	

Table 5 EMV Tags (Continued)

Tag	Description	Format	Required By
9F27	Cryptogram Information Data Indicates the type of cryptogram and the actions to be performed by the terminal.	Normalized	Walker
		Numeric (2)	GPN
		GPN	FDC
		Fixed Length, 1 byte	TSYS
		FDC	
		B, Length 1	
		TSYS	
9F33	Terminal Capabilities Indicates the card data input, CVM and security capabilities of the terminal.	Normalized	Walker
		Numeric (6)	GPN
		GPN	
		Fixed Length 24 bit string, 3 bytes	
9F34	Cardholder Verification Method (CVM) Results Indicates the results of the last CVM performed.	Normalized	Walker
		Numeric (6)	GPN
		GPN	FDC
		Fixed Length, 3 bytes	TSYS
		FDC	
		B, Length 3	
		TSYS	
9F35	Terminal Type Indicates the environment of the terminal, its communications capability and its operational control.	Normalized	Walker
		Numeric (2)	GPN
		GPN	
		Fixed Length, 1 byte	
		TSYS	
		N 2, Length 1	

Table 5 EMV Tags (Continued)

Tag	Description	Format	Required By
9F36	Application Transaction Counter Counter maintained by the application in the ICC (incrementing the ATC is managed by the ICC).	Normalized	Walker
		Numeric (4)	GPN
		GPN	FDC
		Fixed Length 4 hex digits, 2 byte binary value	TSYS
		FDC	
		B, Length 2	
		TSYS	
9F37	Unpredictable Number Value to provide variability and uniqueness to the generation of a cryptogram.	Normalized	Walker
		Numeric (8)	GPN
		GPN	FDC
		Fixed Length 8 hex digits; 4 bytes	TSYS
		FDC	
		B, Length 4	
		TSYS	
9F39	Card Entry Mode	B, Length 4	