XML File Method Integration Developer Kit User's Manual

Document Version 2.7.6

Visit us at: www.pivotalpayments.com

Support

Pivotal Payments is committed to providing the highest quality tools and customer support. If you have any questions, comments or suggestions please contact Pivotal Payments at:

USA Canada

Pivotal Payments Inc.
200 Broadhollow Road, Suite 207
Melville, NY 11747

Pivotal Payments Inc.
685 Cathcart Street, Suite 1000
Montreal, QC. H3B 1M7

Support: 1 866 298-9895

Support: setup@pivotalpayments.com
VAR/Integrator Support: integration@pivotalpayments.com
Sales: gateway_sales@pivotalpayments.com

Thank you for choosing Pivotal Payments!

Integration Methods

File Method Integration – A widely used integration method that uses an XML industry standard file format. The required fields will be formatted in XML tags as an “input file” that will be dropped or sent to a specified location. This specified location is then polled by the SmartPayments Client if there are any input files, processed and then replaced with an output response file that contains data from the payment processor.

TCP Socket Integration – This integration method utilizes the same XML tags as the XML file drop but the difference is that this is not dropped as a file but requested by the POS application as a data stream using the winsock OCX component. An IP address and Port specified in the setting configuration of the SmartPayments Admin Client is required to allow the POS application to open a port send the data stream with the required XML tags and wait on that same socket for a response to be returned with the appropriate payment processor information.

SmartPayments OCX – This integration method utilizes a visual Active X component installed by the SmartPayments Admin Client installer in the Pivotal directory. This OCX gives the advantage of allowing integrators built in hardware support to use with their third party POS integrations. This will save time and resources integrating to a supported hardware device.

IP COM – Allows the integrators to develop their POS applications to call specific SmartPayments Admin Client forms to process transaction requests. The advantage of this integration method is that it saves time and resources developing homegrown forms because the POS application can just call the existing forms to handle transaction requests and response.

XML File Method

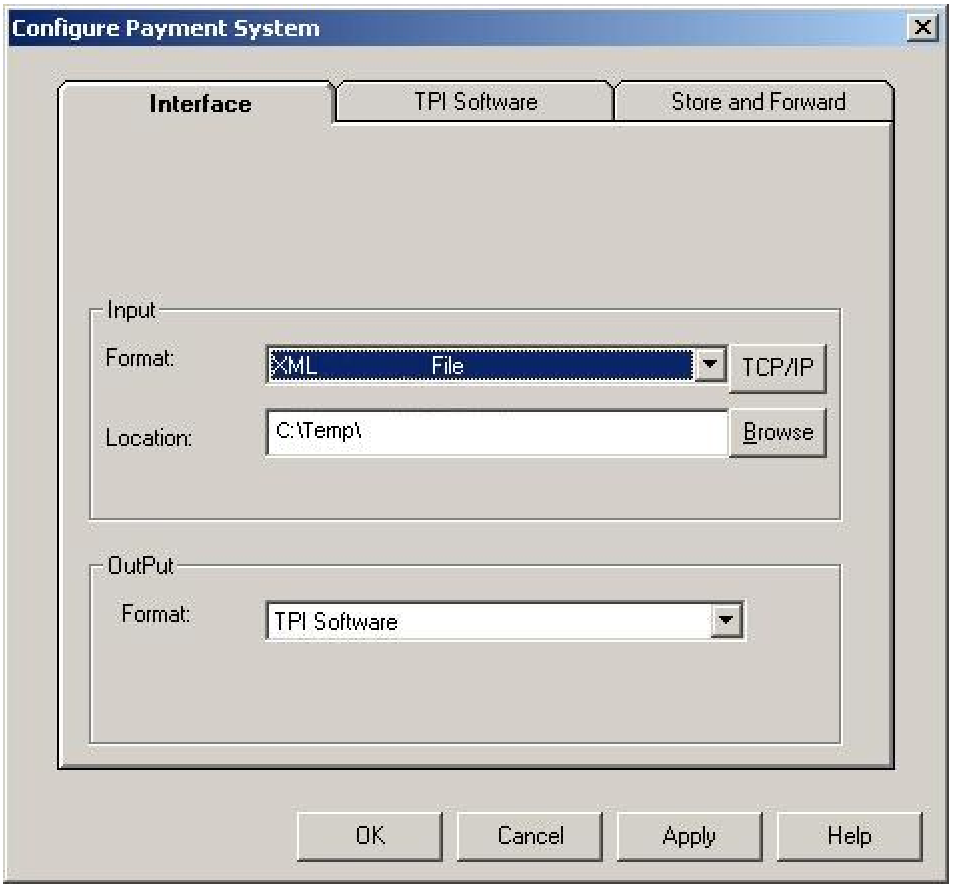
Pivotal Payments Gateway Client XML File Method Integration

Introduction

This XML File format is a widely used format for many applications. It is known to be an industry standard file format. Support for this format was recently added to allow first time developers to the SmartPayments Client Manager to process transactions by sending a named file to a specified location and then retrieve the processed response file from the same location.

Before using this integration method, there are configuration settings that need to be set in the SmartPayments Client User Interface. Please follow these steps in setting up the SmartPayments Client software to be able to accept this XML Format.

1. Launch the Pivotal Payments Gateway Client user interface by double clicking on **smartadmin.exe**
2. Locate the **Set-up** menu and then select **Payments**.
3. Change the Input Format from the default **TCP-XML Pay Format** to **XML Industry Standard File** as shown in the example below.



Format: XML File

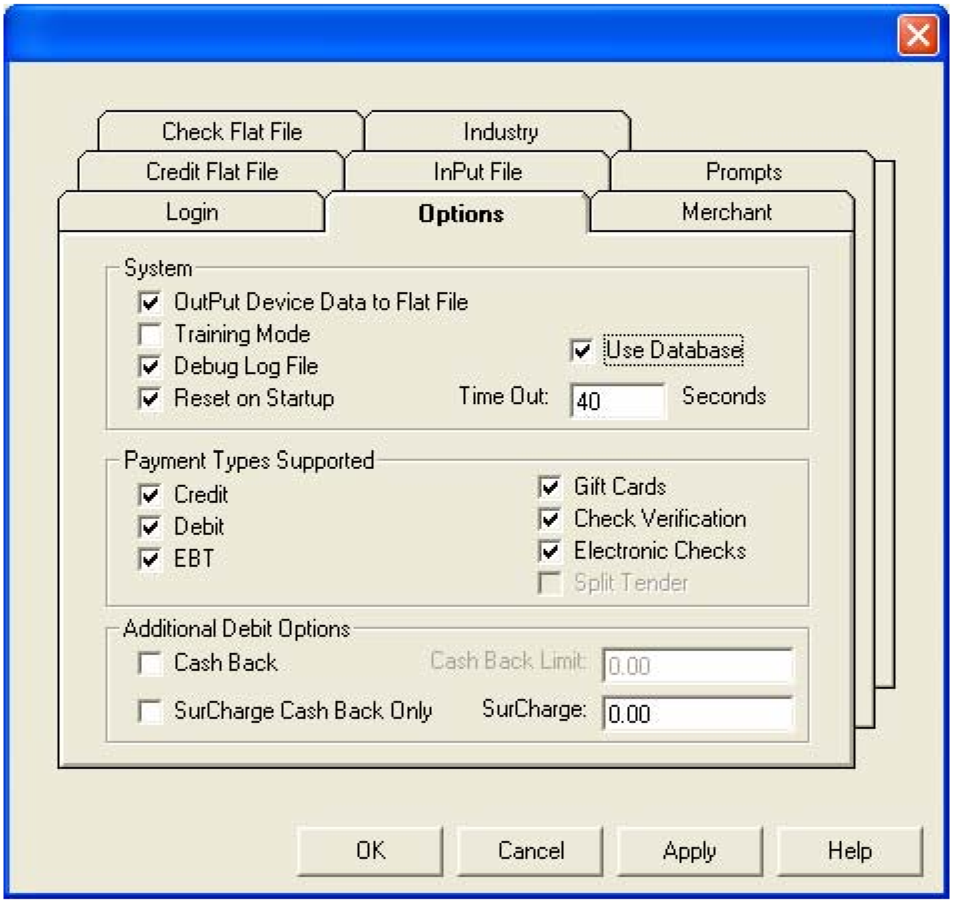
Location: C:\Temp\

Format: TPI Software

4. The L**ocation** field will then be enabled and is used to specify the location that the POS developers will send the input files for processing and retrieve the output response files. The SmartPayments manager will be polling this file location for any files with the extension **.INX** that need to be processed. The folder specified in the **Location** field needs to have the correct permissions set so that the Pivotal Payments Gateway Manager will have full access to it.

5. **Stop** and (re)**Start** the Service to make sure that the configuration changes are committed.

6. If it is the case that the SmartPayments Client Admin will be doing a direct communication to a payment processor instead of a Payment Server, it would be a good practice to enable the local SmartPayments Admin Client database so that the internal sequence numbers/reference numbers will be returned in the output file. The internal sequence or reference numbers are used for follow on transactions such as Voids and Post-Authorizations. The screen shot below shows how this is enabled in the SmartPayments client user interface:



System
☑ OutPut Device Data to Flat File
□ Training Mode
☑ Debug Log File
☑ Reset on Startup
☑ Use Database
Time Out: 40 Seconds

Payment Types Supported
☑ Credit
☑ Debit
☑ EBT
☑ Gift Cards
☑ Check Verification
☑ Electronic Checks
□ Split Tender

Additional Debit Options
□ Cash Back
Cash Back Limit: 0.00
□ SurCharge Cash Back Only
SurCharge: 0.00

Processing Cycle of XML File Drop Integration

1. An input file <*file name*>.inx is dropped in the specified location; i.e. `C:\temp\`.

2. The SmartPayments Manager polls for any files with this extension and once it is found, it performs validation on the file and then changes it into a <*file name*>.pro meaning that this file is in the process of getting routed to the appropriate payment processor.

3. When a response is determined, the SmartPayments manager will then replace this <*file name*>.pro file with the finished <*file name*>.oux file which will be ready for retrieval and parsing by the POS application.

4. After this file has been retrieved or read by the POS application and the information has been processed, it is strongly suggested that this <*file name*>.oux file be destroyed for security purposes.

These are the basic steps that need to be taken for developing a POS application for processing XML files.

1. Write the transaction information to a file in the XML message format (ASCII text) using the file layouts described in this section.

2. Name the file <*file name*>.inx **–** (ex. reg1.inx)

3. Place the <*file name*>.inx file into the **designated** directory. The transaction will now be processed by **SmartPayments**

4. Wait for <*file name*>.oux to appear in the **SmartPayments** directory.

5. Wait for <*file name*>.pro to be deleted from the **SmartPayments** directory.

6. Read the values from <*file name*>.oux. The most important information is returned in the RESULT and AUTH_CODE tags.

7. Delete <*file name*>.oux.

Example
Credit Sale transaction

```
<XML_FILE>
       <XML_REQUEST>
              <COMMAND>1</COMMAND>
              <ACCT_NUM>5439750001500347</ACCT_NUM>
              <EXP_DATE>1208</EXP_DATE>
              <MANUAL_FLAG>0</MANUAL_FLAG>
              <TRANS_AMOUNT>2.00</TRANS_AMOUNT>
       </XML_REQUEST>
</XML_FILE>
```

Credit File Layouts

This section describes the tags required to process credit card transactions.

Credit Input File (.inx)

Tag	Data Type	Description - Credit Input File (.inx)
CARDPRESENT	String	This allows for a value to be passed to Global Payments East to specify if a card was present or not during the transaction. Applicable Values: 0= Not Present 1= Present This property is only applicable to Global Payments EAST.

Tag	Data Type	Description - Credit Input File (.inx)

CVPresence	String	This property describes the status of the CVV value upon submitting a transaction request.
		Applicable Values:
		CVPresence=NOTSUBMITTED
		CVPresence=ILLEGIBLE
		CVPresence=NOT PRESENT
		This property is only applicable to Global Payments EAST.

COMMAND°	String	The action code that identifies what type of transaction will be performed. Consult the **Constants** for a list of valid values.
ACCT_NUM°	String	The credit card number that will be used when processing the transaction. **Max Length:** 20 characters. **Example:** 5439750001500347
EXP_DATE°	String	The expiration date associated with the credit card number that will be processed. Must be exactly four characters long. **Format:** MMYY **Example:** 1208
MANUAL_FLAG°	String	Flag that indicates whether the transaction was manually entered or swiped. If the transaction was swiped, the TRACK_DATA property must also be set. **Valid values:** 0 = manual transaction, 1 = swiped transaction
TRANS_AMOUNT°	String	The amount of the transaction. **Format:** DDDDDD.CC. **Max Length:** 9 characters, including the decimal. The value may not be negative. Do not use commas. **Note:** The amount MUST include the decimal point and the cents even if the amount is a whole dollar amount. **Example:** "3.00", not "3" or "3.". If sending less than one dollar, the zero place holder must be sent as well. **Example:** "0.50".
REFERENCE	String	The reference number from the original transaction (returned by the processor). The REFERENCE property does not need to be set if the Post-Authorization completes a standard Pre-Authorization using the internal sequence value of the Pre-Authorization.
TRACK_DATA**	String	The track II data captured from the magnetic strip of the credit card. The track II data is required to ensure the lowest per-transaction rate from the processing company when performing swiped transactions (Retail and Restaurant). Sending the track II data is not allowed if the merchant's industry type is MOTO or eCommerce. **Max Length:** 40 characters. **Note:** The characters that are appended to the beginning and ending of track II (usually; and ?) should not be passed in. **Example:** 5424180279791765=08121011000001234567
CUSTOMER_CODE	String	Customer code for purchasing/commercial cards. This property must be set for commercial card transactions in order to get the best discount rate. Additionally, the transaction's action code must indicate that the transaction is a commercial card transaction. **Max Length:** 25 characters.

Tag	**Data Type**	**Description - Credit Input File (.inx)**
TAX_AMOUNT	String	The tax amount. This is the portion of the amount that is tax. Providing the tax amount is required to obtain the best rate on commercial card transactions. **Format:** DDDDDD.CC. The transaction's action code must indicate that it is a commercial transaction. **Max Length**: 9 characters (including the decimal).
ZIP_CODE	String	The cardholder's zip code. The Zip property is used for address verification. **Max Length:** 9 digits. Address verification can only be performed on non-swiped transactions. **Note:** For manually keyed transactions, the Zip is required to qualify for the lowest transaction rates. **Note:** If submitting the 9-digit zip, do not include the dash.
STREET	String	The cardholder's billing street address. The Street property is used for address verification. Address verification can only be performed on non-swiped transactions. **Note:** For manually keyed transactions, Street is required to qualify for the lowest transaction rates. **Max Length:** 20 characters
TICKET_NUM	String	The ticket or invoice number for internal referencing by merchant. This value is passed to the processor for referencing purposes. **Max Length:** 9 characters. The value can be alphanumeric. **Note:** Concord/EFS (CCRD) supports only a 6 length numeric ticket or invoice number. Transactions will decline if the TICKET_NUM is not submitted with each transaction. **Note:** For manually keyed transactions, TICKET_NUM is required to qualify for the lowest transaction rates.
CARDHOLDER	String	The cardholder’s name. **Max Length:** 20 characters.
CVV2	String	The CVV2 value for the transaction. The card verification value (CVV2 for Visa, CVC2 for MasterCard, and CID for AMEX and Discover) is a 3 or 4 digit number that is embossed in the signature panel for Visa, MasterCard, and Discover and on the front of the card for AMEX. **Max Length:** 4 characters. CVV2 should only be passed on non-swiped transactions.
CMRCL_FLAG	String	The type of commercial card being submitted. See the section **Commercial Card Transactions** for more information. **Max Length**: 1 character **Valid values:** B – Business P,L,G -- Purchase C – Corporate F – Fleet
AUTH_CODE	String	The Authorization code. This value is returned by the issuing bank and should only be set in a transaction request if processing a Post-Authorization and the Post-Authorization is being used to add a Voice-Authorization to the batch or to “store” a Voice-Authorization. (The AuthCode property does not need to be set if the Post-Authorization completes a standard Pre-Authorization using the reference number value of the Pre-Authorization. See the section **Follow On Transactions** for more information.

Tag	Data Type	Description - Credit Input File (.inx)
GRATUITY_AMNT	String	The amount for tip adjustment. This tag is only applicable to restaurant based merchant accounts. Please see list of Action codes.
TAX_EXEMPT	String	Tax Exempt Flag. This flag is used to indicate if the purchase is tax exempt. Used only for Commercial Card Transactions. **Valid Values:** 1 – Purchase is tax exempt; 0 – Purchase is not tax exempt.
TRANS_ID	String	Please use this tag for follow on transactions specifically for a VOID transaction. The information that needs to be passed in this needs to come from a Reference number from the output file of an Original Sale or Pre-Authorization.
CARDPRESENT	String	This allows for a value to be passed to Global Payments to specify if a card was present or not during the transaction. Applicable Values: 0= Not Present 1= Present This property is only applicable to Global Payments.

CVPresence	String	This property describes the status of the CVV value upon submitting a transaction request.
		Applicable Values:
		CVPresence=NOTSUBMITTED
		CVPresence=ILLEGIBLE
		CVPresence=NOT PRESENT
		This property is only applicable to Global Payments.

FORCESNF	Boolean	This property is only applicable to credit card transactions types. The tag <FORCESNF> Value</FORCESNF> when set to True will cause a transaction to be in store and forward mode and will get a response with a store and forward approval code. This is applicable to payment server /gateway users where store and forward can be enabled. Valid values: True or False.
USER_ID	String	This is a new feature that will allow integrators to specify which account they would like their transaction to be associated with if they have multiple merchant accounts set-up on their client. The limitation of this feature is that the merchant needs to be set up with a Payment Server merchant account. The valid values of this USER_ID are the index value of the profile representing that merchant account. For example, if there are two profiles on the account, one for VITAL and a second profile for Global Payments (Vital Profile Index= 0 ; Global Payments Index = 1. The integrator can then choose to pass "0" to represent the profile index of VITAL or "1" for Global to send transactions to the respective account.

° These properties are the minimum required to process a Sale or Pre-Authorization transaction.
** This field is required when MANUAL_FLAG value is set to 1

Credit Output File (.oux)

Tag	**Data Type**	**Description - Charge Output File (.oux)**
RESULT	String	Returns the result, which indicates the transaction's status upon completion. Refer to the **Transaction Result Constants** section for a list of valid values and descriptions.
AUTH_CODE	String	For approved transactions, returns the authorization code from the issuing bank. For declined transactions, returns the reason why the transaction was declined (if the issuing bank provides one) or why the transaction was rejected.
CARD	String	This returns the last 4 digits of the card number used by the cardholder to perform this payment transaction.
CARD_TYPE	String	This field displays the card type used in the payment transaction. Please see CARD_TYPE list for valid values.
REFERENCE	String	Returns the reference number associated with the transaction. The reference number is assigned by the card associations. The reference number is used to help identify the transaction and is useful for the cardholder and merchant when doing research. This value is not returned with all transactions.
AVS_CODE	String	Returns the AVS response code from the issuing bank. If performing Address Verification on card-not-present transactions, this code indicates how well the AVS information passed in matches what the issuing bank has on file for the cardholder. Consult the section **Constants** for a description of values that may be returned
TRANS_DATE	String	Returns the date that the transaction was processed. This value is not returned by all processing companies.
TICKET	String	Returns the ticket number or invoice of the transaction.
INTRN_SEQ_NUM	String	Returns the Internal Sequence Number, which is an-assigned unique number for each transaction.
TRANS_ITEM_NUM	String	Returns the Transaction Item number or the number that is associated with the transaction in the settlement file. This value is not returned by all processing companies.
TBATCH	String	Returns the active batch number for the transaction. This value is not returned by all processing companies.
TRANS_ID	String	Returns the Transaction Identifier that is returned from the processor. This value is not returned by all processing companies.
TICODE	String	Returns the Transaction Indicator Code that is returned from the processor. The Transaction Indicator Code is a Validation code for VISA / MasterCard. This value is not returned by all processing companies.
IND	String	Returns the IND code. The IND code is a transaction description code and an Interchange compliance field. This value is not returned by all processing companies.
MSI	String	Returns the Market Specific Indicator. This value indicates the transaction's market segment. This value is assigned by the card associations and is not returned with all transactions.
RET	String	Returns the Retrieval reference number. This value is not returned by all processing companies

Tag	Data Type	Description - Charge Output File (.oux)
PEM	String	Returns the POS entry mode that is associated with the transaction. This value is not returned by all processing companies.
TIM	String	Returns the Time of the transaction. This value is not returned by all processing companies.
ACI	String	Returns the Authorization Characteristics Indicator is that is provided by the card associations. This value is stored for settlement.
PROC_RESP_CODE	String	Returns the response code that is provided by the processor. This value is not returned by all processing companies.
REC	String	Returns the record number of the transaction in the reversal file. Will return -1 if the processor doesn't support reversals. This value is not returned by all processing companies.
CMRCL_TYPE	String	Returns the type of commercial card that was used for the transaction. This value is not returned by all processing companies.
CVV2_CODE	String	Returns the CVV2/CVC2/CID response code from the issuing bank. If performing CVV2/CVC2/CID validation on card-not-present transactions, this code indicates if the CVV2/CVC2/CID code passed in matches what the issuing bank has on file for the cardholder. Consult the section **Constants** for a description of values that may be returned
ERRORCODE	String	Returns the error code applicable to the transaction. It will only return a value if the error has occurred.
ERRORDESC	String	Returns the description of the error code and will only return a value if an error has occurred.
MERCHRECEIPT	String	Return this tag if and only if the receipt was successfully generated and if this option is enabled in the Smart Client user interface.
CUSTRECEIPT	String	Returns several carriage return new line to this tag. Please note that if the integrator is expected to properly parse this data through their own point of sale if they are to use this properly. If this is not the case, then do not enable this option in the Smart Client user interface.
PURCH_CARD_TYPE	String	Returns a flag indicating whether the processor indicated whether the card was a Purchasing Card or not. This value is not returned by all processing companies. **Valid values:** 1 = Purchasing Card, 0 = Otherwise
GRATUITY_AMNT	String	Returns the gratuity amount if one is associated with the transaction. This value is not returned by all processing companies.
RESULT_CODE	String	Returns a numerical representation of the result of the transaction. Currently, this field is only used for a batch transaction.
CMRCL_FLAG	String	Returns the Commercial Card Flag. This indicates what type of Commercial card was used for the transaction. This value is not returned by all processing companies.
NET_ID	String	Returns a one character identification code that identifies the network on which the transaction was approved. This value is not returned by all processing companies.

Tag	Data Type	Description - Charge Output File (.oux)
AUTH_SRC_CODE	String	Returns the Authorization Source Code. The authorization source code indicates to the processor who authorized the transaction. This value is not returned by all processing companies. Values Returned: @ = no card reader D - Track 2 T - Keyed track 2 capable
CARD_ID_CODE	String	Returns a code that is used to verify the identity of the cardholder. Value s Returned: M - card present unable to read Magnetic data N - Card not present Z - card signature terminal has a pin pad @ - card signature no pin pad
ACCT_DATA_SRC	String	Returns the entry method of the transaction.
ECOMM_GOODS_IND	String	Returns a value indicating whether the goods sold were digital or physical in an e-commerce environment.

Debit File Layouts

This section describes the tags required to process debit transactions.

When processing debit cards, a PIN pad is required to allow the customer to enter their PIN (Personal Identification Number). In addition, debit card information is always collected via a card swipe device, never via keyboard entry. Because of this, a card reader is also required.

When processing U.S. debit card transactions, merchants have the option of allowing the customer to receive cash back on a transaction. For instance, the customer purchases $50 of products and wants $25 cash back, set the Amount to 50.00 and Cash Back to 25.00. This will withdraw a total of $75 from the debit card account, $50 for the products and $25 for cash to give to the customer.

Debit Input File (.inx)

Tag	Data Type	Description - Debit Input File (.inx)
ACTION°	String	The action code that identifies what type of transaction will be performed. Consult the section **Constants** for a list of valid values.
ACCT_NUM°	String	The Debit card number that will be used when processing the transaction. **Max Length:** 20 characters. **Example:** 5439750001500347

Tag	Data Type	Description - Debit Input File (.inx)
USER_ID	String	This is a new feature that will allow integrators to specify which account they would like their transaction to be associated with if they have multiple merchant accounts set-up on their client. The limitation of this feature is that the merchant needs to be set up with a Payment Server merchant account. The valid values of this USER_ID are the index value of the profile representing that merchant account. For example, if there are two profiles on the account, one for VITAL and a second profile for Global Payments (Vital Profile Index= 0 ; Global Payments Index = 1. The integrator can then choose to pass "0" to represent the profile index of VITAL or "1" for Global to send transactions to the respective account.
EXP_DATE°	String	The expiration date associated with the Debit card number that will be processed. Must be exactly four characters long. **Format:** MMYY **Example:** 1208 Set this property if there is an expiration date associated with the Debit card.
MANUAL_FLAG°	String	Flag that indicates whether the transaction was swiped or manually entered. This property must be set to 1 (swiped) and the TRACK_DATA property must also be set.
TRANS_AMOUNT°	String	The amount of the transaction. **Format:** DDDDDD.CC. **Max Length:** 9 characters, including the decimal. The value may not be negative. Do not use commas. **Note:** The amount <u>MUST</u> include the decimal point and the cents even if the amount is a whole dollar amount. **Example:** "3.00", not "3" or "3.". If sending less than one dollar, the zero place holder must be sent as well. **Example:** "0.50".
TRACK_DATA°	String	The track II data captured from the magnetic strip of the card. The track II data is required. **Max Length:** 40 characters. **Note:** The characters that are appended to the beginning and ending of track II (usually ; and ?) should not be passed in. **Example:** 5439750001500347=08121011000001234567
TICKET_NUM	String	The ticket or invoice number for internal referencing by merchant. This value is passed to the processor for referencing purposes. **Max Length:** 9 characters. The value can be alphanumeric. **Note:** Concord/EFS (CCRD) supports only a 6 length numeric ticket or invoice number
CARDHOLDER	String	The cardholder's name. **Max Length:** 20 characters.
KEY_SERIAL_NUM	String	If a Key Serial Number is obtained from the PIN pad, this property should be populated with that number. **If processing transactions with a PIN pad using DUKPT encryption**, this value is sixteen or twenty characters long (depending on the processor's encryption). **The SmartPayments Client Service will automatically obtain this information if not provided with the transaction.**

Tag	Data Type	Description - Debit Input File (.inx)
CASHBACK_AMNT	String	The amount of cash back that the customer will receive. This amount is in addition to value entered in TRANS_AMOUNT property. For example, if the total amount of the sale is $10 and the customer has requested $5 cash back, TRANS_AMOUNT should be set to $15 and CASHBACK_AMNT should be set to $5. The CASHBACK_AMNT property should be formatted the same the TRANS_AMOUNT property. **Max Length:** 9 characters. **Note:** Some debit processors do not support the cash back feature.
PIN_BLOCK	String	The encrypted PIN block that is retrieved from the PIN pad. The PIN is provided to the processor for verification. **Max Length:** 16 characters.

° These properties are required to process a Debit Sale transaction.

Debit Output File (.oux)

Tag	Data Type	Description - Debit Output File (.oux)
RESULT	String	Returns the result, which indicates the transaction's status upon completion. Refer to the **Transaction Result Constants** section for a list of valid values and descriptions.
AUTH_CODE	String	For approved transactions, returns the authorization code from the issuing bank. For declined transactions, returns the reason why the transaction was declined (if the issuing bank provides one) or why the transaction was rejected.
CARD	String	This returns the last 4 digits of the card number used by the cardholder to perform this payment transaction. Note: Now Applicable to Global Canada.
CARD_TYPE	String	This field displays the card type used in the payment transaction. Please see CARD_TYPE list for valid values. Note: Now Applicable to Global Canada.
REFERENCE	String	Returns the reference number associated with the transaction. The reference number is used to help identify the transaction and is useful for the cardholder and merchant when doing research. This value is not returned with all transactions.
TRANS_DATE	String	Returns the date that the transaction was processed. This value is not returned by all processing companies.
TICKET	String	Returns the ticket number or invoice of the transaction. This value is echoed back from the original transaction.
INTRN_SEQ_NUM	String	Returns the Internal Sequence Number, which is an-assigned unique number for each transaction.
TRANS_ITEM_NUM	String	Returns the Transaction Item number or the number that is associated with the transaction in the settlement file. This value is not returned by all processing companies.
TBATCH	String	Returns the active batch number for the transaction. This value is not returned by all processing companies.
TICODE	String	The TICODE field contains the last byte of the Working key that is provided by the processor.

Tag	Data Type	Description - Debit Output File (.oux)
TIM	String	Returns the Time of the transaction. This value is not returned by all processing companies.
NET_ID	String	Returns a one-character identification code that identifies the network on which the transaction was approved.

Check File Layouts

This section provides information on the tags that are required to process check transactions.

Check Input File (.INX)

Tag	Data Type	Description - Check Input File (.inx)
ACTION°	String	The action code that identifies what type of transaction will be performed. Consult the **Constants** for a list of valid values.
ACCT_NUM°	String	For Check, MICR, or Double ID: The account number that will be used when processing the transaction. **Max Length:** 20 characters.
USER_ID	String	This is a new feature that will allow integrators to specify which account they would like their transaction to be associated with if they have multiple merchant accounts set-up on their client. The limitation of this feature is that the merchant needs to be set up with a Payment Server merchant account. The valid values of this USER_ID are the index value of the profile representing that merchant account. For example, if there are two profiles on the account, one for VITAL and a second profile for Global Payments (Vital Profile Index= 0 ; Global Payments Index = 1. The integrator can then choose to pass "0" to represent the profile index of VITAL or "1" for Global to send transactions to the respective account.
MANUAL_FLAG°	String	Flag that indicates whether the transaction was manually entered or swiped. If the transaction was swiped, the TRACK_DATA property must also be set. **Valid values:** 0 = manual transaction, 1 = swiped transaction
TRANS_AMOUNT°	String	The amount of the transaction. **Format:** DDDDDD.CC. **Max Length:** 9 characters, including the decimal. The value may not be negative. Do not use commas. **Note:** The amount MUST include the decimal point and the cents even if the amount is a whole dollar amount. **Example:** "4.00", not "4" or "4.". If sending less than one dollar, the zero place holder must be sent as well. **Example:** "0.65". This has to be passed in the INX file.
ZIP_CODE	String	The cardholder's zip code. The Zip property is used for address verification. **Max Length:** 9 digits. Address verification can only be performed on non-swiped transactions. **Note:** For manually keyed transactions, the Zip is required to qualify for the lowest transaction rates. **Note:** If submitting the 9-digit zip, do not include the dash.

Tag	Data Type	Description - Check Input File (.inx)
TICKET_NUM	String	The ticket or invoice number for internal referencing by merchant. This value is passed to the processor for referencing purposes. **Max Length:** 9 characters. The value can be alphanumeric. **Note:** Concord/EFS (CCRD) support only a 6 length numeric ticket or invoice number. Transactions will decline if the TICKET_NUM is not submitted with each transaction. **Note:** For manually keyed transactions, TICKET_NUM is required to qualify for the lowest transaction rates.
MICR	String	The raw MICR data from the bottom of the check. Used for conversion transaction.
STATE°°	String	The state code of the state that issued the check writer's driver's license. The state code is required for DL (Driver's License). **Format:** 2 characters.
LICENSE°°	String	The driver's license number of the individual writing the check. **Max Length:**20 characters. The driver's license is required for DL (Driver's License) transactions and when performing Double ID transactions.
ABA_NUM°°	String	The Transit Routing Number / ABA number that will be used when processing the transaction. This value indicates which bank issued the check. **Max Length:** 9 characters. This value is required for MICR transactions and when performing Double ID transactions.
PHONE_NUM°°	String	The phone number of the individual writing the check. **Format:** digits only. The phone number is required for COD (Checks On Delivery). **Max Length:** 7 digits.
DOB°°	String	The date of birth of the check writer. **Format:** MMDDYY. The birth date is required for DL (Driver's License) check transactions. **Max Length:** Exactly six characters.
CHECK_NUM°	String	The check number of the check that will be used when the transaction. **Max Length:** 10 characters.

Note: To perform Double ID, both the MICR and LICENSE fields must be populated.
° These properties are required, regardless of service type.
** The user name is used to keep the transaction associated with the correct terminal
°° COD – required for Checks-On-Delivery
DL – required for Driver's License
MICR – required for MICR

Check Output File (.OUX)

Tag	Data Type	Description - Check Output File (.oux)
RESULT	String	Returns the result, which indicates the transaction's status upon completion. Refer to the **Transaction Result Constants** section for a list of valid values and descriptions.

Tag	Data Type	Description - Check Output File (.oux)
AUTH_CODE	String	For approved transactions, returns the authorization code from the issuing bank. For declined transactions, returns the reason why the transaction was declined (if the issuing bank provides one) or why the transaction was rejected.
REFERENCE	String	Returns the reference number associated with the transaction. The reference number is assigned by the card associations. The reference number is used to help identify the transaction and is useful for the cardholder and merchant when doing research. This value is not returned with all transactions.
TRANS_DATE	String	Returns the date that the transaction was processed. This value is not returned by all processing companies.
TICKET	String	Returns the ticket number or invoice of the transaction.
INTRN_SEQ_NUM	String	Returns the Internal Sequence Number, which is an-assigned unique number for each transaction.
PROC_RESP_CODE	String	Returns the response code that is provided by the processor. This value is not returned by all processing companies.

Gift File layouts

This section describes the XML tags required to process Gift/Loyalty transactions.

Gift Card Input File (.INX)

Tag	Data Type	Description – Gift Input File (.INX)
ACTION°	String	The action code that identifies what type of transaction will be performed. Consult the **Constants** for a list of valid values.
ACCT_NUM°	String	The gift card number that will be used when processing the transaction. **Max Length:** 20 characters. **Example:** 5439750001500347
USER_ID	String	This is a new feature that will allow integrators to specify which account they would like their transaction to be associated with if they have multiple merchant accounts set-up on their client. The limitation of this feature is that the merchant needs to be set up with a Payment Server merchant account. The valid values of this USER_ID are the index value of the profile representing that merchant account. For example, if there are two profiles on the account, one for VITAL and a second profile for Global Payments (Vital Profile Index= 0 ; Global Payments Index = 1. The integrator can then choose to pass "0" to represent the profile index of VITAL or "1" for Global to send transactions to the respective account.
EXP_DATE°	String	The expiration date associated with the credit card number that will be processed. Must be exactly four characters long. **Format:** MMYY **Note: Most gift cards have no expiration date.** **Example:** 1249

Tag	Data Type	Description – Gift Input File (.INX)
MANUAL_FLAG°	String	Flag that indicates whether the transaction was manually entered or swiped. If the transaction was swiped, the TRACK_DATA property must also be set. **Valid values:** 0 = manual transaction, 1 = swiped transaction
TRANS_AMOUNT°	String	The amount of the transaction. **Format:** DDDDDD.CC. **Max Length:** 9 characters, including the decimal. The value may not be negative. Do not use commas. **Note:** The amount MUST include the decimal point and the cents even if the amount is a whole dollar amount. **Example:** "3.00", not "3" or "3.". If sending less than one dollar, the zero place holder must be sent as well. **Example:** "0.50".
REFERENCE	String	The reference number from the original transaction (returned by the processor). The REFERENCE property does not need to be set if the Post-Authorization completes a standard Pre-Authorization using the internal sequence value of the Pre-Authorization.
TRACK_DATA**	String	The track II data captured from the magnetic strip of the credit card. The track II data is required to ensure the lowest per-transaction rate from the processing company when performing swiped transactions (Retail and Restaurant). Sending the track II data is not allowed if the merchant's industry type is MOTO or eCommerce. **Max Length:** 40 characters. **Note:** The characters that are appended to the beginning and ending of track II (usually ; and ?) should not be passed in. **Example:** 5424180279791765=0812101100001234567
AUTH_CODE	String	The Authorization code. This value is returned by the issuing bank and should only be set in a transaction request if processing a Post-Authorization and the Post-Authorization is being used to add a Voice-Authorization to the batch or to "store" a Voice-Authorization. (The AuthCode property does not need to be set if the Post-Authorization completes a standard Pre-Authorization using the internal sequence value of the Pre-Authorization. See the section **Follow On Transactions** for more information.
FORCE_FLAG	String	Flag Indicating whether the transaction should be forced for Paymentech, Valid Values: 1- force, 0 – don't force

° These properties are the minimum required to process a Sale or Pre-Authorization transaction.
** This field is required when MANUAL_FLAG value is set to 1

Gift Card Output File (.OUX)

Tag	Data Type	Description – Gift Output File (.OUX)
RESULT	String	Returns the result, which indicates the transaction's status upon completion. Refer to the **Transaction Result Constants** section for a list of valid values and descriptions.

Tag	Data Type	Description – Gift Output File (.OUX)
AUTH_CODE	String	For approved transactions, returns the authorization code from the issuing bank. For declined transactions, returns the reason why the transaction was declined (if the issuing bank provides one) or why the transaction was rejected.
CARD	String	This returns the last 4 digits of the card number used by the cardholder to perform this payment transaction.
CARD_TYPE	String	This field displays the card type used in the payment transaction. Please see CARD_TYPE list for valid values.
REFERENCE	String	Returns the reference number associated with the transaction. The reference number is assigned by the card associations. The reference number is used to help identify the transaction and is useful for the cardholder and merchant when doing research. This value is not returned with all transactions.
TRANS_DATE	String	Returns the date that the transaction was processed. This value is not returned by all processing companies.
TICKET	String	Returns the ticket number or invoice of the transaction.
INTRN_SEQ_NUM	String	Returns the Internal Sequence Number, which is an-assigned unique number for each transaction.
TRANS_ITEM_NUM	String	Returns the Transaction Item number or the number that is associated with the transaction in the settlement file. This value is not returned by all processing companies. This will contain the PNREF information which is a SmartPayments Client Unique Identifier.
TRANS_ID	String	Returns the Transaction Identifier that is returned from the processor. This value is not returned by all processing companies.
RET	String	Returns the Retrieval reference number. This value is not returned by all processing companies
PEM	String	Returns the POS entry mode that is associated with the transaction. This value is not returned by all processing companies.
TIM	String	Returns the Time of the transaction. This value is not returned by all processing companies.
ACI	String	Returns the Authorization Characteristics Indicator is that is provided by the card associations. This value is stored for settlement.
PROC_RESP_CODE	String	Returns the response code that is provided by the processor. This value is not returned by all processing companies.
CARD_ID_CODE	String	Returns a code that is used to verify the identity of the cardholder.
ACCT_DATA_SRC	String	Returns the entry method of the transaction.

EBT File Layouts

This section contains the tags required to process EBT transactions.

For processing EBT Cards, a pin pad is required to allow for the customer to enter their PIN. The EBT information is usually collected via a card swipe device instead of keyboard entry so it is a requirement as well to have this card swipe reader. (**Note:** some EBT processors would allow for manually keyed in transactions).

EBT Input File (.INX)

Tag	**Data Type**	**Description – EBT Input File (.INX)**
ACTION°	String	The action code that identifies what type of transaction will be performed. Consult the **Constants** for a list of valid values.
ACCT_NUM°	String	The credit card number that will be used when processing the transaction. **Max Length:** 20 characters. **Example:** 5439750001500347
EXP_DATE°	String	The expiration date associated with the credit card number that will be processed. Must be exactly four characters long. **Format:** MMYY **Example:** 1208
MANUAL_FLAG°	String	Flag that indicates whether the transaction was manually entered or swiped. If the transaction was swiped, the TRACK_DATA property must also be set. **Valid values:** 0 = manual transaction, 1 = swiped transaction
REFERENCE	String	The reference number from the original transaction (returned by the processor). The REFERENCE property does not need to be set if the Post-Authorization completes a standard Pre-Authorization using the internal sequence number value of the Pre-Authorization.
TRANS_AMOUNT°	String	The amount of the transaction. **Format:** DDDDDD.CC. **Max Length:** 9 characters, including the decimal. The value may not be negative. Do not use commas. **Note:** The amount MUST include the decimal point and the cents even if the amount is a whole dollar amount. **Example:** “4.00”, not “4” or “4.”. If sending less than one dollar, the zero place holder must be sent as well. **Example:** “0.80”.
TRACK_DATA**	String	The track II data captured from the magnetic strip of the credit card. The track II data is required to ensure the lowest per-transaction rate from the processing company when performing swiped transactions (Retail and Restaurant). Sending the track II data is not allowed if the merchant's industry type is MOTO or eCommerce. **Max Length:** 40 characters. **Example:** 5424180279791765=0812101100000 1234567 **Note:** The characters that are appended to the beginning and ending of track II (usually ; and ?) should not be passed in.
KEY_SERIAL_NUM	String	If a Key Serial Number is obtained from the PIN pad, this property should be populated with that number. **If processing transactions with a PIN pad using DUKPT encryption**, this value is sixteen or twenty characters long (depending on the processor’s encryption). **The SmartPayments Client Service will automatically obtain this information if not provided with the transaction.**

Tag	Data Type	Description – EBT Input File (.INX)
PIN_BLOCK	String	The encrypted PIN block that is retrieved from the PIN pad. The PIN is provided to the processor for verification. **Max Length:** 16 characters
ZIP_CODE	String	The cardholder's zip code. The Zip property is used for address verification. **Max Length:** 9 digits. Address verification can only be performed on non-swiped transactions. **Note:** For manually keyed transactions, the Zip is required to qualify for the lowest transaction rates. **Note:** If submitting the 9-digit zip, do not include the dash.
TICKET_NUM	String	The ticket or invoice number for internal referencing by merchant. This value is passed to the processor for referencing purposes. **Max Length:** 9 characters. The value can be alphanumeric. **Note:** Concord/EFS (CCRD) supports only a 6 length numeric ticket or invoice number. Transactions will decline if the TICKET_NUM is not submitted with each transaction. **Note:** For manually keyed transactions, TICKET_NUM is required to qualify for the lowest transaction rates.
CARDHOLDER	String	The cardholder’s name. **Max Length:** 20 characters.
CASHBACK_AMNT	String	The amount of cash back that the customer will receive. This amount is in addition to value entered in TRANS_AMOUNT property. For example, if the total amount of the sale is $10 and the customer has requested $5 cash back, TRANS_AMOUNT should be set to $15 and CASHBACK_AMNT should be set to $5. The CASHBACK_AMNT property should be formatted the same the TRANS_AMOUNT property. **Max Length:** 9 characters. **Note:** Some debit processors do not support the cash back feature.
EBT_TYPE	String	EBT_TYPE Indicates what type of EBT transaction will be performed. Valid values: F- Food Stamp Transaction; C-Cash Benefits transaction.
AUTH_CODE	String	For an EBT Forced Authorization, the original authorization code would need to be passed in this tag from the Pre_Auth transaction.

° These properties are the minimum required to process a Sale or Pre-Authorization transaction.
** This field is required when MANUAL_FLAG value is set to 1

EBT Output File (.OUX)

Tag	Data Type	Description – EBT Output File (.OUX)
RESULT	String	Returns the result, which indicates the transaction's status upon completion. Refer to the **Transaction Result Constants** section for a list of valid values and descriptions.
AUTH_CODE	String	For approved transactions, returns the authorization code from the issuing bank. For declined transactions, returns the reason why the transaction was declined (if the issuing bank provides one) or why the transaction was rejected.
CARD	String	This returns the last 4 digits of the card number used by the cardholder to perform this payment transaction. Note: Now Applicable to Global Canada
CARD_TYPE	String	This field displays the card type used in the payment transaction. Please see CARD_TYPE list for valid values. Note: Now Applicable to Global Canada.
REFERENCE	String	Returns the reference number associated with the transaction. The reference number is assigned by the card associations. The reference number is used to help identify the transaction and is useful for the cardholder and merchant when doing research. This value is not returned with all transactions.
TRANS_DATE	String	Returns the date that the transaction was processed. This value is not returned by all processing companies.
TICKET	String	Returns the ticket number or invoice of the transaction.
INTRN_SEQ_NUM	String	Returns the Internal Sequence Number, which is an-assigned unique number for each transaction.
TRANS_ITEM_NUM	String	Returns the Transaction Item number or the number that is associated with the transaction in the settlement file. This value is not returned by all processing companies.
TBATCH	String	Returns the active batch number for the transaction. This value is not returned by all processing companies.
TRANS_ID	String	Returns the Transaction Identifier that is returned from the processor. This value is not returned by all processing companies.
TICODE	String	Returns the Transaction Indicator Code that is returned from the processor. The Transaction Indicator Code is a Validation code for VISA / MasterCard. This value is not returned by all processing companies.
MSI	String	Returns the Market Specific Indicator. This value indicates the transaction's market segment. This value is assigned by the card associations and is not returned with all transactions.
RET	String	Returns the Retrieval reference number. This value is not returned by all processing companies
PEM	String	Returns the POS entry mode that is associated with the transaction. This value is not returned by all processing companies.
TIM	String	Returns the Time of the transaction. This value is not returned by all processing companies.
NET_ID	String	Returns a one character identification code that identifies the network on which transactions approved.

Tag	Data Type	Description – EBT Output File (.OUX)
EBT_FOOD_BALANCE	String	Returns the remaining Balance on a Food Stamp card. This value is not returned by all processing companies.
EBT_CASH_BALANCE	String	Returns the remaining balance on a Cash Benefits card. This value is not returned by all processing companies.

° These properties are required to process transactions.

Notes: These XML tags follow the industry standard formats and please be advised that these formats. This format has its own ERROR CODE Definition:

XML File Sample

Input File Structure (For a swiped transaction)

```
<XML_FILE>
      <XML_REQUEST>
            <COMMAND>1</COMMAND>
            <ACCT_NUM>5439750001500347</ACCT_NUM>
            <EXP_DATE>1208</EXP_DATE>
            <MANUAL_FLAG>1</MANUAL_FLAG>
            <TRANS_AMOUNT>1.00</TRANS_AMOUNT>
            <TRACK_DATA>5439750001500347=0812</TRACK_DATA>
            <ZIP_CODE>85284</ZIP_CODE>
            <STREET>8320</STREET>
            <CARDHOLDER>John Doe</CARDHOLDER>
      </XML_REQUEST>
</XML_FILE>
<XML_FILE>
      <XML_REQUEST>
            <RESULT>CAPTURED</RESULT>
            <AUTH_CODE>QAV1MC</AUTH_CODE>
            <REFERENCE>13</REFERENCE>
            <AVS_CODE>R</AVS_CODE>
            <TRANS_DATE>1125</TRANS_DATE>
            <TICKET></TICKET>
            <TRANS_ID>13</TRANS_ID>
            <MSI></MSI>
            <PEM></PEM>
            <TIM>160226</TIM>
            <RET></RET>
            <ACI>Y</ACI>
            <PROC_RESP_CODE>00</PROC_RESP_CODE>
            <CMRCL_TYPE>0</CMRCL_TYPE>
            <PURCH_CARD_TYPE>0</PURCH_CARD_TYPE>
            <CVV2_CODE></CVV2_CODE>
            <INTRN_SEQ_NUM>13</INTRN_SEQ_NUM>
            <TRANS_ITEM_NUM>008</TRANS_ITEM_NUM>
            <TICODE></TICODE>
            <CARD_ID_CODE>@</CARD_ID_CODE>
            <ACCT_DATA_SRC>D</ACCT_DATA_SRC>
      </XML_REQUEST>
</XML_FILE>
```

Note: If you wish to force a duplicate transaction using the same credit card and amount, just place an “F” at the beginning of the Account Number string. This action will force the transaction to be sent for authorization.

Appendices:
A. Test Card Information

Integrators can use these following test cards for manually keyed in transaction testing. If a full track II is required for the test, please contact our Pivotal support team at setup@pivotalpayments.com

Card Type	**Card Number**	**Expiration Date**
VISA	4387755555555550	1209
Mastercard	5454545454545454	1210
AMEX	371449635398431	1210
Discover	6011000995500000	1210
Procurement Card (Mastercard)	5405010100000016	1210
Procurement card (VISA)	4275330012345675	1210

Notes on Testing:

Please secure a testing account with the payment processor of your choice or Pivotal support will assist you in testing against our test hosted gateway. Contact us for this information so you may begin testing against a simulated live merchant account setting for those integrators that will be routing to a Payment Server gateway.

Debit Transactions for PIN based debit will require an encryption scheme injected pin pad by a specific processor and either a test plastic card or a live card for testing. This is required because all true debit transactions need to be swiped only accompanied by a personal identification number (PIN).

For those that want the debug log files turned on, you may enable this by launching the SmartPayments Admin / Client executable and proceed to Setup|Configure|Options and select the Debug log file. This debug log file when enabled will log transactions processed through the SmartPayments Admin Client and is located in the Pivotal directory as SmartPayments.log.

B. Action/Command Codes

Credit Card

Command Code	Description
1	Sale
2	Credit
3	Void Sale
4	Pre-Authorization
5	Post-Authorization
6	Void Credit
7	Void Post-Authorization
8	Commercial Card Sale
9	Commercial Card Credit
10	Commercial Card Post-Authorization
13	Gratuity

Debit

Action Code	Description
41	Sale
42	Return
43	Void
46	Void Return

Gift

Action Code	Description
18	Balance
25	Redemption/ Sale
26	Refund/Credit
27	Increment
28	Activate
29	Cancel/Deactivate
30	Void

Check

Action Code	Description
20	MICR
21	Phone
22	Driver’s License (Verify)
23	Double ID
24	Void Check
51	Sale (MICR)
52	Void(MICR)
53	Force(MICR)

EBT

Action Code	Description
60	Account Inquiry
61	Cash Withdrawal
62	Food Stamp Purchase
63	Food Stamp Credit

Batch

Action Code	Description
31	Batch Settlement
85	Batch Inquiry

Store and Forward

Action Code	Description
98	Store and Forward

C. Result and Response Codes

Address Verification Response Codes

Response Code	Address Match
A	Address matches, ZIP code does not
B	Address matches, postal code does not
C	No match on address or postal code
D	Street address and postal code matches
E	AVS error
G	Service not supported by non-US issuer
I	Address not verified for international transaction
M	Street address and postal code matches
N	No match on address or ZIP code
P	Postal code matches, address does not
R	Retry, system is unavailable or timed out
S	Service not supported by issuer (card type does not support AVS)
U	Address information is unavailable
W	9-digit ZIP code matches, address does not
X	Exact match
Y	Address and 5-digit ZIP code match
Z	5-digit ZIP code matches, address does not
0	No response sent

CVV2/CVC2/CID Response Codes

Response Code	Address Match
M	CVV2/CVC2/CID match
N	CVV2/CVC2/CID mismatch
P	Not processed -- Either the expiration date was not provided, or the card does not have a CVV2/CVC2/CID value. If the expiration date was left blank, resubmit as a zero dollar amount for the transaction so the customer's credit line won't be affected by the second CVV2/CVC2/CID request.
S	Issuer indicates that the CVV2/CVC2/CID data should be present on the card, but the merchant has indicated that the CVV2/CVC2/CID data is not present on the card.
U	Issuer has not certified for CVV2/CVC2/CID or issuer has not provided Visa/MasterCard with the CVV2/CVC2/CID encryption keys.

D. Constants and Definitions

Transaction Result Constants

Result	Transaction Type	Description
CAPTURED	Monetary	Successful online transaction now ready for settlement
NOT CAPTURED	Varies	Unsuccessful online transaction
APPROVED	Non-Monetary	Successful offline transaction for Terminal based processors, or successful Pre-Authorization for Host based processors)
NOT APPROVED	Varies	Unsuccessful offline transaction or unsuccessful Pre-Authorization for Host based processors
CANCELLED	Any	Transaction canceled by operator or modem never connected

Credit Card Transactions Definitions

There are several types of credit card transactions. The different types of transactions are referred to as actions. The following is a list of the various actions with general descriptions.

Term	Definition
Sale	Diminishes the cardholder's open to buy amount, and places the transaction in the open batch. This action is commonly used in retail environments.
Void Sale	Eliminates a sale transaction from the open batch. The funds will not be deposited into the merchant's bank account at settlement/close. Void Sale is typically used for returns (usually same day) or to correct cashier mistakes. A Void Sale can only be done before settling/closing of the batch (this usually means the action has to be performed on the same day as the sale).
Credit	Used to refund money to the cardholder. This action is typically used after the batch that contains the Sale (or Post-Authorization) transaction has been settled or closed. This action will increase the cardholder's open to buy amount once the batch containing the credit has been settled.
Void Credit	Removes a credit transaction from the batch. This can only be done before the batch is settled/closed (this usually means the action has to be performed on the same day as the credit). This transaction is not available with all processing companies. If the Void Credit action is not available, use the Void Sale action.
Pre-Authorization	Diminishes the cardholder's open to buy amount. It does not place a transaction in the open batch. A Pre-Authorization can be considered one of two steps of a sale. A Pre-Authorization reduces the limit to buy for only a set amount of time, usually 7-10 days, which is determined by the credit card's issuing bank. To place the transaction in the open batch, a follow-on transaction (called a Post-Authorization) must occur. This action is commonly used in MOTO (Mail Order / Telephone Order) and eCommerce environments.
Commercial Card Sale	Diminishes the cardholder's open to buy amount, and places the transaction in the open batch. This action is similar to a standard credit card sale, but is typically used if the card tendered is a procurement, purchasing, business, government, or commercial card. Two additional values, the Tax amount and customer code, must be passed with this type of card in order for the merchant to qualify for the lowest transaction rate.

Term	Definition
Commercial Card Credit	Typically used after the batch that contains the Procurement Card Sale (or Procurement Card Post-Authorization) transaction has been settled or closed. This action is similar to a standard credit card credit, but is typically used if the card tendered is a procurement, purchasing, business, government, or commercial card. This action will increase the cardholder's limit to buy once the batch containing the credit has been settled. Two additional values, the Tax amount and customer code, must be passed with this type of card in order for the merchant to qualify for the lowest transaction rate.
Commercial Card Post-Authorization	Places an approved Pre-Authorization transaction into the open batch. This action is similar to a standard credit card Post-Authorization, but is typically used if the card tendered is a procurement, purchasing, business, government, or commercial card. This action can be considered the second half of a sale. This follow-on transaction must occur before a Pre-Authorization can be settled/closed. This Post-Authorization may also be used to place an approved Voice-Authorization in the batch. Two additional values, the Tax amount and customer code, must be passed with this type of card in order for the merchant to qualify for the lowest transaction rate.
Sale with Gratuity	Used only in a restaurant environment, this action allows a server to authorize the amount of the meal plus the gratuity and place the entire amount in the batch. When settled, the amount plus the gratuity is deducted from the cardholder’s account.

Note: Pre-Authorization cannot be voided.

Check Transaction Definitions

Check Guarantee and Check Verification are transactions that do not involve an electronic transfer of funds. To determine that the check writer has an existing account and does not have current outstanding checks, check verification can be performed by the merchant. The Check Guarantee service extends the guarantee that the merchant will get a percentage of his money even if the check bounces. There is a higher fee associated with this check guarantee service depending on the check processor the merchant is associated with.

Check Verification and Check Guarantee are single step processes and there are three types of check transactions.

Check Verification – This transaction permits the merchant to confirm that the check writer has a checking account and does not have any outstanding checks.

Check Guarantee – This transaction allows the merchant to verify that the check writer has an existing account and that the amount for that transaction can guaranteed.
Note: The guaranteed amount may vary, so please consult with your check processor.

Check Conversion – This action allows the merchant to transmit the MICR information from the check to the issuing bank and receive electronically deposited funds, thus eliminating the need for paper checks.

Check Transaction types as supported by SmartPayments Client.

Verify	Enables the merchant to confirm that the customer has an existing checking account and assures the transaction amount is available. This action also enables the merchant to perform the first of two steps in the check 'sale' transaction.
Sale	Diminishes the checking account balance of the customer. A sale executes two functions; verification of the check and finalization the transaction for settlement.
Void	Eliminates a Sale or Force transaction for the settlement information. The transaction will be deleted and no funds will be received from this transaction. The Void Sale action is used to edit and correct mistakes on the same day before the settlement process is finalized. This can only be performed before the settlement is closed.
Force	Finalizes a verified check transaction in the settlement batch. A Verify transaction is followed by a Force, which is equal to a Check Sale.

Debit Transaction Definitions

The items listed in the table are the supported debit card commands with their definitions.

Sale	Diminishes the checking account balance of the customer.
Void	Removes a Sale transaction from the open Batch or can also reverse a Return transaction. The debit void is not supported by all payment processors.
Return	This action allows a merchant to credit funds back into the card holder's debit account.

Debit Cards are also known as check cards and their use has proliferated greatly in the last few years. Debit Cards are more akin to checks due to the fact that when used for a purchase, this action immediately deducts the balance of the purchase from the checking account of the card holder. ATM debit cards can also be used to withdraw cash from the banks.

***Notes to Integrators:**

1. If you are planning to integrate debit transactions: please be aware that true debit transactions are “in person”, requiring a card swipe and entry of the card holder’s PIN. This is true unless the merchant is enrolled with a “PIN-less” Debit Account. Please consult your Merchant Service Providers for details about this program.

2. Please be aware that a pin pad is required for Debit card transactions and make sure that these have been injected with encryption keys by your merchant’s payment processor network. These strong encryption keys are specific to the payment processor.

3. Please consult the properties specific to debit card transactions supported by the different integration methods. TRACK, Key Serial Number, PIN Block data elements are required in addition to the basic sale information transmitted for normal credit card transactions.

4. Debit CHECK cards can be used as credit cards as well without the PIN and these are often referred to as ‘offline’ debit cards. These are debit cards that are branded as check cards with a credit card association logo.

Gift Card Definitions

Gift Card programs have become popular over the last few years. Gift Cards are plastic cards similar to a credit or debit card that can be both hand keyed or swiped through a card reader. These are basically designed as an alternative to the paper loyalty cards or gift certificates. When a gift card is purchased either through cash or credit cards, the gift card will then have to be activated with either a fixed or variable amount tied to the gift card. Some gift card programs allow for reloading more funds to that gift card that can be used to purchase items from a store.

When a purchase is done using a gift card, the request is sent to a gift card processor and the amount will be debited from the card value. A balance on the gift card is usually returned along with the response from the gift card sale. Since gift card processors are host based, settlement close is usually not applicable.

EBT Definitions

EBT stands for electronic benefits transfer wherein the government of the United States can issue benefits electronically. These benefits cover food stamps, cash benefits issued as a social security payment and disability payments. The processing of an EBT transaction is very similar to a debit card transaction. It is required for this type of transaction to be swiped and face to face in a retail environment such as a grocery store. A pin pad is usually required as well for terminals that will be processing EBT transactions.

EBT transactions are usually host based but it is advised to consult with the merchant service provider or payment processor representative on the topic of settlement close.

There are 4 basic tender types associated with EBT transactions.

Food Stamp Sale	Processes a food stamp sale that deducts from the previous balance of the EBT card.
Food Stamp Return	This action allows a merchant to run a credit back funds to the EBT card.
Cash Benefit	This action allows transaction requests in relation to government program cash benefits.
Balance Inquiry	Allows the merchant to view the balance on the EBT account.