iTransact Gateway Documentation

iTransact Group, LLC

iTransact Gateway Documentation

iTransact Group, LLC

Publication date Version 4.1 - June 13, 2013

Table of Contents

List of Figures

List of Tables

Chapter 1. General Information

1. About Us

1.1. Our Company

iTransact is a privately held payment acceptance company based near Salt Lake City, Utah. iTransact was founded in 1994 as RediCheck, the first provider of Internet-based check acceptance services. Today, we offer merchant accounts (and related services and equipment) in addition to our proprietary Internet payment gateway. Our services allow Internet, bricks-and-clicks, and traditional brick-and-mortar merchants to accept payments via credit, debit, and gift cards, as well as electronic checks (and/or check conversion with check guarantee).

1.2. iTransact PaymentClearing Gateway

The iTransact payment gateway is compatible with merchant accounts obtained virtually anywhere. It can be integrated into nearly any shopping cart software or used as a stand-alone Virtual Terminal [7]. Run transactions directly on iTransact's secure servers or via a secure server of your own.

2. Payment Processing

2.1. What is a gateway and why do I need one?

A gateway system is software that acts as a secure conduit of transaction information to the credit card and check processing networks. A gateway is much like a physical payment terminal used to swipe cards or read checks at a brick-and-mortar location, but it is designed to communicate through secure Internet connections. Top of the line gateways also provide reporting, automated recurring, swipe options, mobile transactions, and the ability to run transactions for previously entered customers without re-entering transaction data.

The ability to process credit cards or check payment over the Internet requires several pieces:

1. Website, mobile app, or order submission form
2. Gateway account
3. Credit card merchant account or check processing account
4. Deposit account at a merchant's bank

If any of these items are not setup correctly, a business will not be able to successfully charge and/or receive payments from their customers over the Internet.

2.2. Which credit card processing platforms/networks are compatible with the gateway?

The software used by iTransact is certified for credit card processing on most of the main processing networks. Please contact iTransact with any questions regarding those certifications. Any transaction submission method can be used for credit card processing.

The following processing platforms/networks are compatible with the iTransact gateway:

- First Data - Nabanco (aka FDMS Canada/FDMS South) [2]
- First Data - Cardnet [2]
- First Data - Omaha ETC Type 7 [2]
- Paymentech/Gensar [2]

- NDC Global Atlanta East [3]
- Elavon/NOVA [3]
- TSYS (aka Vital/Visanet) [3]
- Heartland Exchange [3]
- Vantiv 610 Tandem [4]

2.3. What information is needed to enable credit card processing?

The following is the information required to activate credit card processing [64] in the gateway for the separate processors.

2.3.1. First Data - Nabanco (aka FDMS Canada/FDMS South)

First Data - Nabanco

Table 1.1. First Data - Nabanco (aka FDMS Canada/FDMS South)

Merchant Number (*Required)*	11 Digits
Terminal ID (*Required*)	2 Digits (Default "99")
Datawire ID (*Required*)	20 Digits
Discover SE (if accepting Discover)	10 Digits
AMEX SE (if accepting AMEX)	10 or 11 Digits

2.3.2. First Data - Cardnet

First Data - Cardnet

Table 1.2. First Data - Cardnet

Merchant Number (*Required*)	12 Digits
Terminal ID (*Required*)	6 Digits
Datawire ID (*Required*)	20 Digits

2.3.3. First Data - Omaha ETC Type 7

First Data - Omaha

Table 1.3. First Data - Omaha ETC Type 7

Merchant Number (*Required*)	12, 15, or 16 Digits
Device ID (*Required*)	4 Digits (Default "0001")
Datawire ID (*Required*)	20 Digits

2.3.4. Paymentech/Gensar

Paymentech/Gensar (Account MUST be Terminal-based)

Table 1.4. Paymentech/Gensar

Merchant Number (*Required*)	12 Digits
Terminal Number (*Required*)	3 Digits

Client/ISO/Bank Number (*Required*)	4 Digits
NetConnect User Name	Up to 30 Characters
NetConnect Password	Up to 30 Characters

2.3.5. NDC Global Atlanta East

NDC Global Atlanta East

Table 1.5. NDC Global Atlanta East

Terminal Number (*Required*)	7-13 Digits
Bank ID (*Required*)	6 Digits

2.3.6. Elavon/NOVA

Elavon/NOVA (VAR Type: iTransact PaymentClearing - Elavon Settlement Type MUST be set to *Manual* [not *Auto*])

Table 1.6. Elavon/NOVA

Terminal Number (*Required*)	16 Digits (Usually ending in "99")
Bank ID (*Required*)	6 Digits

2.3.7. TSYS (aka Vital/Visanet)

TSYS

Table 1.7. TSYS (aka Vital/Visanet)

Merchant Number (*Required*)	12 Digits
Terminal Number (*Required*)	4 Digits
BIN/Bank ID Number (*Required*)	6 Digits
Store Number (*Required*)	4 Digits
Agent Number (*Required*)	6 Digits
Chain Number (*Required*)	6 Digits
Merchant Category/SIC Code (*Required*)	4 Digits
Terminal ID/V#	7 Digits

2.3.8. Heartland Exchange

Heartland Exchange

Table 1.8. Heartland Exchange

Merchant Number (*Required*)	12 Digits
Terminal Number (*Required*)	4 Digits
BIN/Bank ID Number (*Required*)	6 Digits
Store Number (*Required*)	4 Digits
Agent Number (*Required*)	6 Digits
Chain Number (*Required*)	6 Digits
Merchant Category/SIC Code (*Required*)	4 Digits

Terminal ID/V#	7 Digits

2.3.9. Vantiv 610 Tandem

Vantiv 610 Tandem

Table 1.9. Vantiv 610 Tandem

Merchant ID (*Required*)	12 Digits (add zeros if needed)
Bank ID (*Required*)	4 Digits
Terminal Number (*Required*)	3 Digits (add zeros if needed)

2.4. What check processing options are there?

iTransact's system offers three methods for accepting check payments online. It is not necessary to accept credit card payments to accept checks. A merchant can, however, accept both credit cards and checks. Check transactions can be submitted to the gateway through all of the normal transaction methods (Virtual Terminal, form, and or XML/API requests) but not currently through any of the mobile apps. The following methods can be used to accept checks through the gateway:

- RediCheck check printing
- EFT processing (through CheckGateway)
- NACHA formatted file creation

Please keep in mind that for any of these methods, check funds can not be verified in real-time. There is a potential that a check transaction can still bounce (be rejected) even after it is submitted through the gateway.

2.5. What is needed to enable check processing?

There are different requirements to be met and things to be considered when using any of the methods for check processing.

RediCheck check printing

iTransact's RediCheck service was the world's first method for businesses to securely accept check payments over the Internet. Merchants who opt to use this method to process check payments will submit transactions through forms integrated with the gateway, and once each week, our RediCheck department will create check drafts (using the submitted customer banking information). These drafts are printed and mailed to the merchant. The merchant needs to take those drafts and deposit them at their bank. Remember, there is a potential for those check payments to bounce in the same way that a hand-written check can bounce. Keep that in mind when fulfilling orders. iTransact is never notified whether a printed check clears or bounces. Merchants need to track that in conjunction with their deposit bank. Please contact your sales agent for pricing.

EFT processing

Merchants who opt to use this method to process check payments will submit transactions through forms integrated with the gateway, and the gateway will connect with the electronic funds transfer processor (CheckGateway) for validation and verification. If the customer information is validated, a check payment begins the EFT process. This process normally takes 5-7 days. If the EFT is successful, the money will deposit. iTransact is never notified whether an EFT clears or bounces. The processor provides details and reports concerning each transaction (in addition to the limited information available in the gateway's Transaction Listing [32] and Transaction Detail window).

To use this method, a merchant must apply and be approved by the EFT processor - and then follow any special instructions imposed by the EFT provider. The provider will issue a six digit EFT account ID. Provide this to the *iTransact deployment team* when you are ready to activate EFT processing in your gateway. Please contact your sales agent for pricing.

NACHA formatted file creation

NACHA file creation is a method that allows a merchant to process check transactions through the gateway without the need of a third party processor. Transactions are submitted through the gateway, a NACHA formatted file is generated, and then the merchant must download the file from the gateway and upload that file to their deposit bank's NACHA/Treasury account system.

Prior to real transactions being allowed through the gateway, a merchant must

1. be approved for and have activated a NACHA processing account (often called a Treasury Services account) at their deposit bank
2. provide the *iTransact deployment team* with all of the necessary values (provided by their bank) for these required fields:
   - **Company ID** - 10-digit company number
   - **Company Name** - company's name
   - **Immediate Destination** - bank's transit routing number
   - **Immediate Destination Name** - bank's name
   - **Immediate Origin** - 10-digit company number
   - **Immediate Origin Name** - company's name
   - **Originating DFI** - bank's transit routing number
3. run a test transaction batch and verify deposit

Most banks provide detailed reporting for Treasury Services, including rejected payments. Unfortunately, iTransact is never notified whether a check payment clears or bounces. Please remember that because there is no validation of funds in real time, NACHA transactions can still bounce or be rejected. Not all deposit banks offer Treasury Services. iTransact does not have any say in what fees or charges a deposit bank assesses for NACHA transactions. Please work with your bank representative to negotiate those rates. Please contact your iTransact sales agent regarding gateway pricing.

2.6. How long do deposits take?

Credit Cards

Funds from credit card transactions are deposited into a merchant's checking account approximately 2-3 business days after the credit card transaction settles.

Check Payments

- **RediCheck payments**: Since these are printed check drafts, they are dealt with by your deposit bank in the same manner that your bank would deal with a hand-written check. Please remember that you need to deposit the check drafts when thy are mailed to you. Those are sent by mail once a week.
- **EFT payments**: Funds are deposited into a merchant's checking account approximately 5-7 business days after it is transacted by the processor.
- **NACHA file payments**: Please check with your bank representative concerning their deposit time frame.

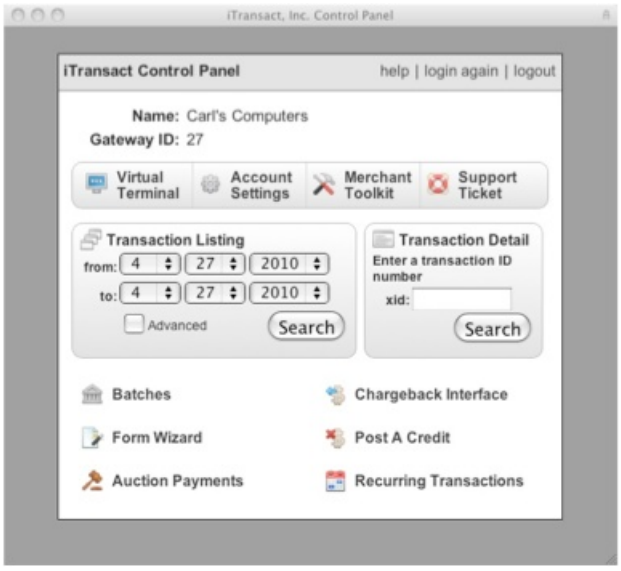
Chapter 2. User Information

1. The Control Panel

1.1. What is the Control Panel?

The Control Panel is the main user interface for the iTransact gateway. All gateway merchants have access to this tool. It includes access to reporting, running transactions, recurring tools, integration information, settlement tools, and gateway settings. This is the standard layout of the Control Panel.

Figure 2.1. Control Panel



When your account is activated, you will be given the login location, the five digit gateway ID, and the password used to open this interface. If you have lost or forgotten your password, please contact the *support team*. A session will expire after 10 minutes of inactivity.

1.2. Blue Control Panel Information

The Blue Control Panel is an alternative version of the Control Panel interface, but is not necessary for all merchants. If your sales agent has activated the alternate interface for your account, your gray Control Panel will remain available. The instructions for use of the Blue Control Panel are available in the documentation area of the *Merchant Toolkit*. If you have questions about this interface, please contact the *support team*.

NOTE: *The Blue Control Panel is currently in beta testing. As such, it occasionally does not function the way that it would if it was a full production release. Please be aware of this and exercise patience with any issues you run into. Please contact our support team to make them aware of what you are experiencing. Ideally, we would like to be able to fix issues as they are discovered.*

1.3. Control Panel Features

The Control Panel offers several helpful tools to merchants. The following is a list and short explanation of each tool.

- **Virtual Terminal** - This tool is used much like a physical payment terminal. Customer information can be entered to attempt a transaction on a credit card or check. This feature can also be used with conjunction with a USB swipe reader. [7]

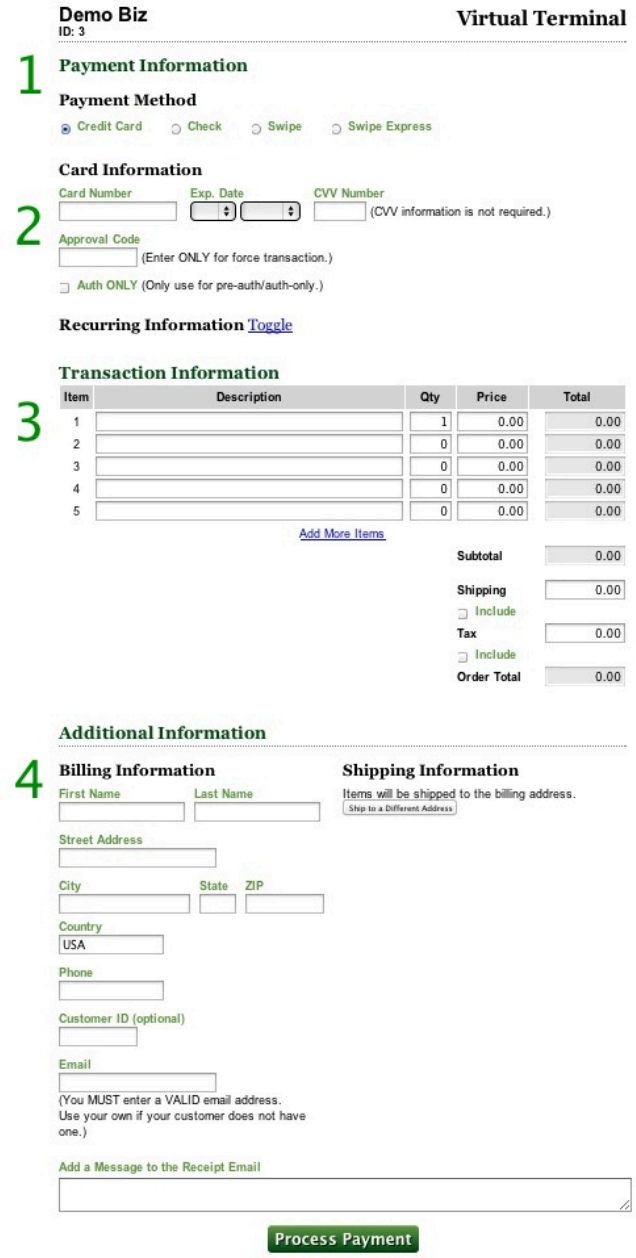
- **Account Settings** - This feature is used to view and/or update contact information, anti-fraud features, email settings, and general transaction functions. [19]
- **Merchant Toolkit** - This contains full integration instructions for the gateway and information that will simplify the process of integrating the gateway system with a website. The toolkit includes examples of the simplest and most advanced methods for linking a website with the gateway. It also includes downloads of the support documentation. [31]
- **Support Ticket** - Use this to submit questions and support requests concerning your account.
- **Transaction Listing** - This interface displays transaction information in several formats according to a date range or search criteria, and allows for report downloads. [32]
- **Transaction Detail** - Use this to search for the details of a specific transaction using the unique XID number assigned by the gateway. [40]
- **Batches** - Using this window, a merchant can view a batch history, the data from the current open batch, download NACHA files, and generate a manual settlement. EFT data does not list in this interface. This only displays information for Credit Card, NACHA, and RediCheck transactions. [46]
- **Chargeback Interface** - This feature is used to locate a transaction if a credit card number and the approximate date of the transaction are known. This is normally used by merchants in a chargeback situation, when a cardholder has disputed a charge. [48]
- **Form Wizard** - This tool is designed to allow a merchant to create forms quickly and easily for use with the Split Form, Standard Form, and BuyNow formats. This simple interface walks a merchant through several questions and then builds a functioning HTML form. [48]
- **Post A Credit** - This feature should only be used to generate a refund or payment to a cardholder's account that was not originally charged through the gateway account with the permission of your merchant service provider. By default, this functionality is disabled and the feature does not display in the *Account Settings*. Contact the support team for additional information. [57]
- **Auction Payments** - This can be used by merchants who sell items in online auctions to accept check and credit card payments. [59]
- **Recurring Transactions** - All information and tools that can be used to generate, track, and update recurring transactions can be accessed in this area. [60]
- **Card Setup (Not Pictured)** - When an account is first activated, your merchant account data must be submitted through this tool. In many cases, this will be performed by the sales agent for the account. [64]

2. Virtual Terminal

This tool is used much like a physical payment terminal. Customer information can be entered to attempt a transaction on a credit card or check. This feature can also be used with conjunction with a USB swipe reader.

2.1. Standard Credit Card Interface

Figure 2.2. Virtual Terminal - Standard Credit Card Interface



The standard credit card Virtual Terminal Interface is the page that opens by default when a merchant clicks on the `Virtual Terminal` link from the Control Panel [6]. This interface allows for a merchant to type in several different items for purchase, as well as separate shipping and tax charges for the entire purchase. Recurring transactions can also be entered here. The interface can be used for check/EFT/NACHA or credit card payments.

1. **Welcome Section** - When the Virtual Terminal opens, the merchant is greeted with their business name, gateway ID number and the format options at the top of the interface. A merchant using the standard Virtual Terminal does not need to click on any of the optional interfaces in the `Payment Method` section.

2. **Card Information Section** - This displays by default or will be re-enabled when a merchant selects the `Credit Card` radio button.

   - **Card Number** - The customer's credit card number should be entered into this field without any dashes or spaces.

- **Exp. Date** - The drop down menus must be selected for the account's expiration month and year.
- **CVV Number** - The value for this field is the CVV, CVV2, or CID code listed on the credit card. This three or four digit numeric code is used as a fraud deterrent.
- **Approval Code** - The value for this field can only be obtained directly from the credit card merchant account processor's voice approval phone service. This feature should only be used if a "call authorization center" error response was received during a previous authorization attempt. The approval code will be a numeric or alpha-numeric code provided by the voice approval service. The gateway does not provide voice approval codes. Those codes must be obtained directly from the merchant account processor.
- **Auth Only Checkbox** - A merchant should never check this box, unless they do not desire to actually charge a customer's card. When this is selected, the transaction will only run a pre-authorization which verifies the card account and a freezes the amount in the account, but it does not actually charge the card. A pre-authorized transaction can be converted to a full transaction by running a post-authorization from the Transaction Listing or through the API. If no post-authorization is run, the money is eventually released back to the customer's account - and never paid to the merchant.

3. **Transaction Information Section** - Some of the entry fields in this area are required and others are optional. A merchant can choose to enter up to ten separate items plus shipping and tax amounts, or can submit a single item which is a total of the amount to be billed to the customer. To access items 6-10, please use the `Add More Items` option.

   - **Item Description** - A merchant should enter the name of the product that a customer is purchasing in this field. This information will be recorded in the merchant's Transaction Details in the Control Panel and in the Merchant [66]/Customer [66] confirmation emails. Some merchants choose to enter all of the items in a single line item - either with each item detailed, or with a generic description like “Purchased Items”. This can be done as long as the value for the `Item Qty` is “1” and the total price of the purchase is entered into the `Item Price` field.
   - **Item Qty** - This value will be multiplied by the amount listed in the `Item Price` field to provide the value for the `Item Total`. This value can be “1”, even if you are selling multiple quantities - as long as the `Item Price` amount is the cost of all of the products combined.
   - **Item Price** - The amount listed here will be multiplied by the value listed in the `Item Qty` field to provide the value for the `Item Total`.
   - **Item Total** - This value is calculated automatically when the Virtual Terminal multiplies the entered value of the `Item Qty` and the `Item Price` for a single item.
   - **Subtotal** - This amount is the sum of the `Item Totals` for all items purchased.
   - **Include Shipping Checkbox** - This should be selected if a merchant would like shipping to be a separate line item. This must be used in conjunction with an entry in the `Shipping Amount` field.
   - **Shipping Amount** - This value should be the amount of shipping for the entire purchase. The Virtual Terminal does not calculate shipping. A merchant will need to calculate that prior to entering the amount in this interface. If the `Include Shipping` checkbox is selected, there must be a value in this field.
   - **Include Tax Checkbox** - This should be selected if a merchant would like tax to be a separate line item. This must be used in conjunction with an entry in the `Tax Amount` field.
   - **Tax Amount** - This value should be the amount of tax for the entire purchase. The Virtual Terminal does not calculate tax rates. A merchant will need to calculate that prior to entering the transaction in this interface. If the `Include Tax` checkbox is selected, there must be a value in this field.
   - **Order Total** - This value is the sum of the `Total`, the `Shipping` amount, and the `Tax` amount. This is the amount that will be charged to the customer's card. If this value is zero for a credit card, the transaction will run as an AVSOnly transaction (if supported by your processor).

- **Email Text** - This field allows a merchant to enter a message up to 255 characters which will display on both the merchant confirmation email and on the customer confirmation email.

4. **Additional Information Section** - The account holder and shipping data is entered in this area.

- **Billing Information** - All fields here are required unless otherwise indicated.
  - First Name - This should be the customer's first name.
  - Last Name - This should be the customer's last name.
  - Address - This should be the account holder's street address as listed with the account issuer.
  - City - This should be the account holder's city as listed with the account issuer.
  - State - This should be the state abbreviation of the account holder as listed with the account issuer.
  - ZIP - This should be the account holder's postal code as listed with the account issuer.
  - Country - This should be the account holder's country as listed with the account issuer.
  - Phone - This should be a contact phone number for the customer.
  - Customer ID - This is an optional field that allows a merchant to enter a tracking code for the transaction.
  - Email - This should be the customer's email address. The transaction confirmation email will be sent to this address.
- **Optional Shipping Information** - These fields can be accessed by clicking the `Ship to a Different Address` button. Each of these fields are optional and can be populated with alternative shipping address data. The processing banks are unable to verify this information.
- **Email Text** - This field allows a merchant to enter a message up to 255 characters which will display on both the merchant confirmation email and on the customer confirmation email.

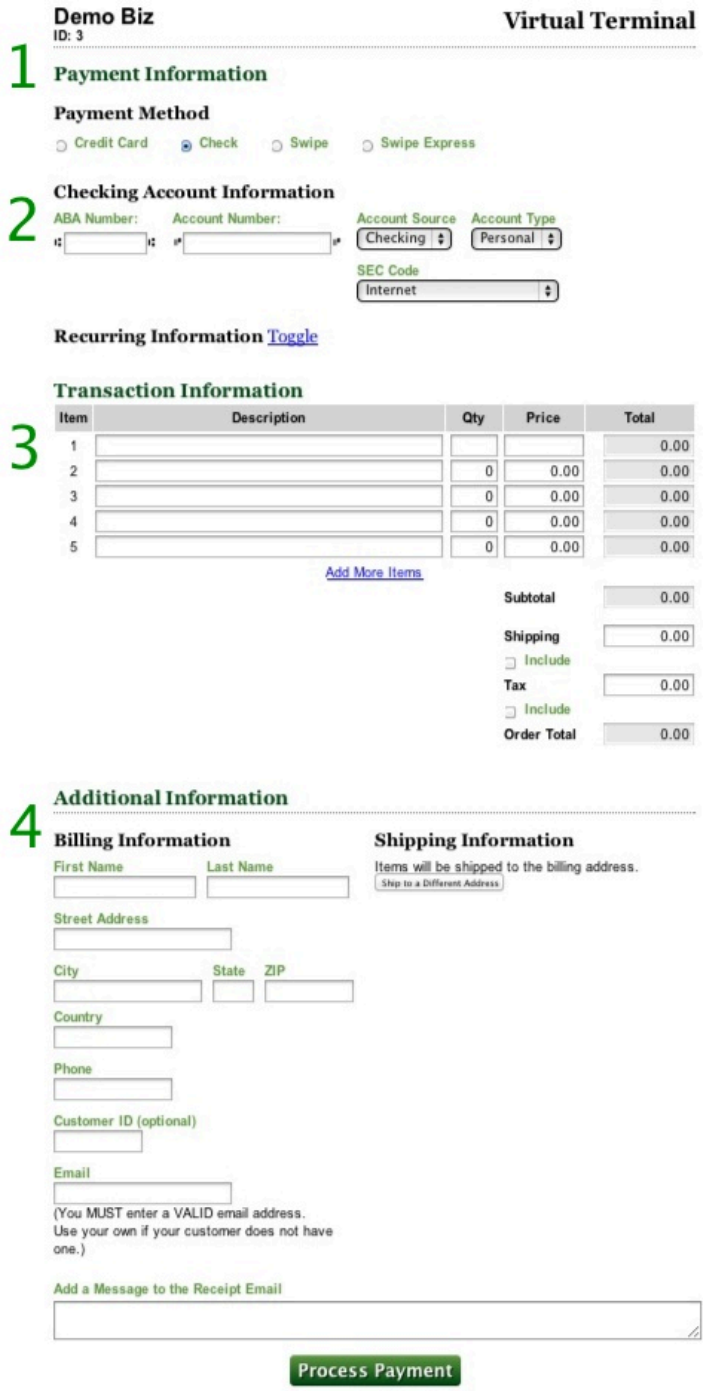
Processing Transactions

Submit a charge through this interface by entering correct data into the required fields and any of the desired optional fields. Be sure to double check the account information as well as the amount of the transaction. Click the `Process Payment` button. The transaction will be attempted in real-time. The gateway will either display an approval screen or a failure screen [17]. The failure screen will list the reason for the failure. An approval page will display information specific to the successful transaction.

2.2. Check/EFT Interface

The Virtual Terminal will display checking account fields when the Check radio button in the Payment Method section. Areas 1, 3, and 4 are the same as the standard credit card interface [8], but area 2 displays the checking account entry fields for check and EFT transactions.

Figure 2.3. Virtual Terminal - Check/EFT Interface



1. **Welcome Section** - Select `Check`

2. **Checking Account Information** - The following fields will be enabled when a merchant selects the `Check Payment Method` radio button.

   - **ABA Number** - This is the nine digit ABA Routing number for a customer's bank. These are generally the first nine numbers listed in the line of numbers across the bottom of a check.

   - **Account Number** - This is the account holder’s checking account number as it appears on a check.

   - **Account Type** - (For EFT Transactions Only) A merchant needs to use the selection tool to indicate whether the customer's checking account is a personal or a business checking account.

- **Account Source** - Use the selection tool to indicate whether the customer's checking account is a checking or a savings account.
- **SEC Code** - (For EFT Transactions Only) Depending upon the nature of the EFT processing account, a merchant may be required to designate the three letter Standard Entry Category for the transaction. Potential values are:
  - PPD - Prearranged payment and deposit
  - CCD - Corporate credit or debit
  - ARC - Accounts receivable entry
  - BOC - Back office conversion
  - POP - Point of purchase
  - RCK - Returned check entry
  - WEB - Internet initiated entry
  - TEL - Telephone initiated entry

3. **Transaction Information Section -** See details

4. **Additional Information Section -** See details

Processing Transactions

Submit a charge through this interface by swiping the card and entering correct data into the required fields and any of the desired optional fields. Be sure to double check the amount of the transaction. Click the `Process Payment` button. The transaction will be attempted in real-time. The gateway will either display an approval screen [17] or a failure screen [17]. The failure screen will list the reason for the failure. An approval page will display information specific to the successful transaction.

2.3. Recurring Transactions

If your transaction needs to be set as a recurring transaction, click the `Recurring Information Toggle` button to display the appropriate entry fields. When clicked for a standard credit card or check transaction the following section will display.

Figure 2.4. Virtual Terminal - Recurring Transactions

Recurring Information Section

- **Recipe Name** - This drop down menu displays all of a merchant's pre-built recipes [61]. A recipe provides the rules and schedule by which a transaction will re-bill when set with at least one remaining repetition.

- **Number of Repetitions** - This is the numeric value for the amount of times the recipe needs to cycle. Each successful repetition will cycle down the number of remaining repetitions by one until it reaches zero (or until the transaction is set to zero repetitions [150]).

- **Recurring Total** - This is optional. If the amount that is to recur is the same as the total amount listed in the initial transaction information, please leave this blank. This feature can be used in conjunction with recipes designated by the merchant as a Split Amount [63]recipe. When an amount is entered into this field, that will be the amount billed when the transaction recurs. For example, merchants who bill a one time setup fee and then a different amount for monthly service fees would put the amount of the monthly service fee in the `Recurring Total` field.

- **Recurring Description** - This is optional. If the `Item Description` used in the initial transaction information is a sufficient explanation for both the initial payment and any subsequent recurring transactions, please leave this blank. If, however, the merchant would like this to display differently on subsequent recurring billings, please enter an adequate description in this field.

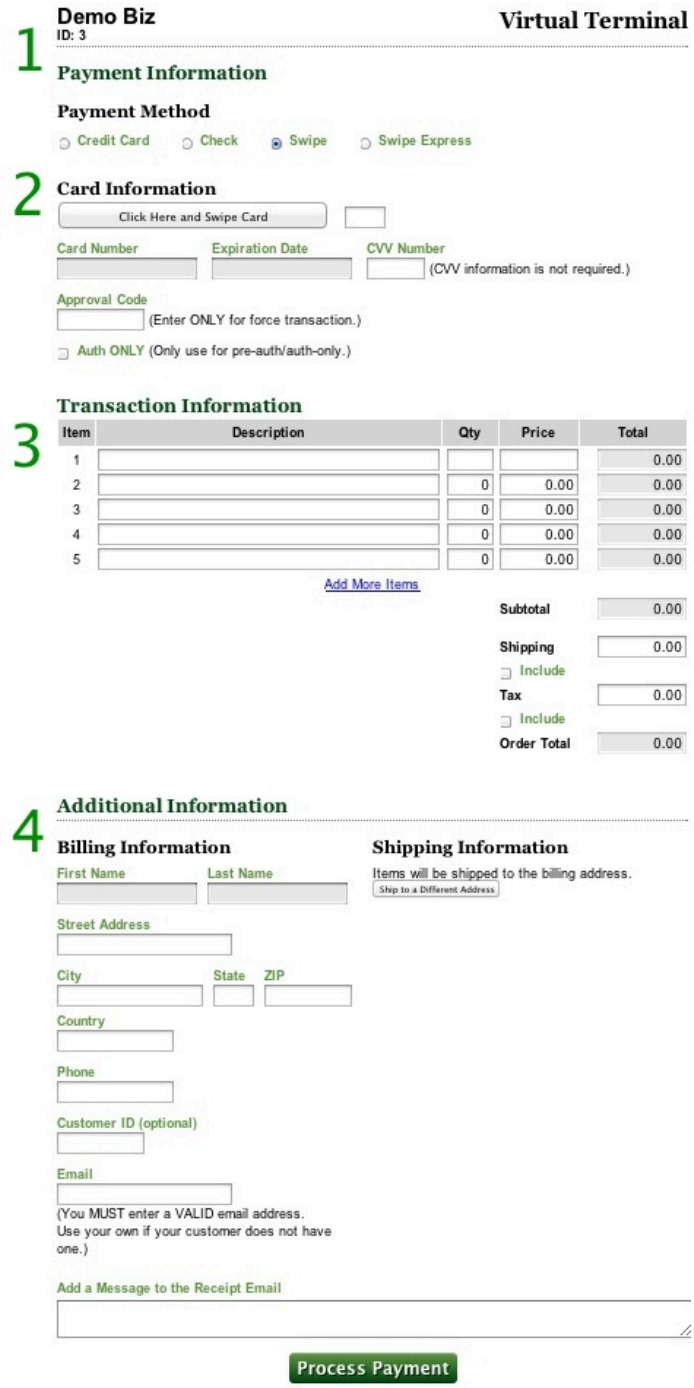
Processing Transactions

Submit a charge through this interface by swiping the card and entering correct data into the required fields and any of the desired optional fields. Be sure to double check the amount of the transaction. Click the `Process Payment` button. The transaction will be attempted in real-time. The gateway will either display an approval screen or a failure screen [17]. The failure screen will list the reason for the failure. An approval page will display information specific to the successful transaction.

2.4. Swipe Card Interface

The standard Swipe Card interface can be accessed by clicking on the `Swipe Card` link in the Standard interface.

Figure 2.5. Virtual Terminal - Swipe Card Interface



This interface is to be used by merchants who are using a compatible USB-Connected magnetic card swipe reader. The Magtek Stripe Reader-USB and the IDTech MiniMag USB Swipe Reader have been tested with the gateway and found to be compatible. Human Input Device (HID) and encrypted swipers are not compatible with the gateway. If you need help purchasing one, please contact our *support team*.

Using the interface, a card may be swiped through the reader and it will populate the payment fields. The card holder billing information must be entered on this form. This will allow an Address Verification and a CVV check to take place on any transaction attempt. Areas 1, 3, and 4 are the same as the standard credit card interface, but area 2 displays the swipe entry section.

1. **Welcome Section** - Select `Swipe`

2. **Card Information Section** - Click on the `Swipe` button and swipe the credit card through a compatible USB-Connected magnetic card swipe reader, this section will be automatically populated by the encrypted data embedded in the magnetic strip on the back of the card.

3. **Transaction Information Section** - See details

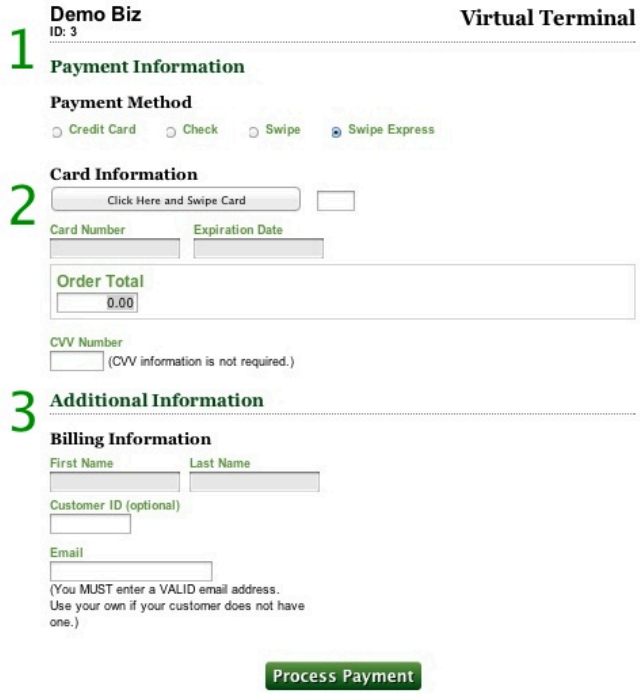
4. **Additional Information Section** - See details

Processing Transactions

Submit a charge through this interface by swiping the card and entering correct data into the required fields and any of the desired optional fields. Be sure to double check the amount of the transaction. Click the `Process Payment` button. The transaction will be attempted in real-time. The gateway will either display an approval screen or a failure screen [17]. The failure screen will list the reason for the failure. An approval page will display information specific to the successful transaction.

2.5. Swipe Card Express Interface

Figure 2.6. Virtual Terminal - Swipe Card Express Interface



The Express Swipe Card entry area is accessed by clicking on the `Swipe Card Express` link in the Standard interface. It allows a merchant to enter minimal information and swipe a credit card to attempt a transaction. This interface is to be used by merchants who are using a compatible USB-Connected magnetic card swipe reader. The Magtek Stripe Reader-USB and the IDTech MiniMag USB Swipe Reader have been tested with the gateway and found to be compatible. When in this interface, a card may be swiped through the reader and it will populate the required payment fields. There are no address fields, so Address Verification is unavailable.

1. **Welcome Section** - Select `Swipe Express`

2. **Card Information Section** - Click on the button and then swiping the credit card through a compatible USB-Connected magnetic card swipe reader, this section will be automatically populated by the encrypted data embedded in the magnetic strip on the back of the card.

   - **Order Total** - This is the amount that will be charged to the customer's card. If this value is zero, the transaction will run as an AVSOnly transaction (if supported by your processor).

- **CVV Number** - The value for this optional field is the CVV, CVV2, or CID code listed on the credit card. This three or four digit numeric code is used as a fraud deterrent.

3. **Additional Information** - These fields will display after a successful swipe entry.

- **Email** - This is optional, but should be the customer's email address. The transaction confirmation email will be sent to this address.

- **Custom ID** - This is an optional field that allows a merchant to enter a tracking code for the transaction.

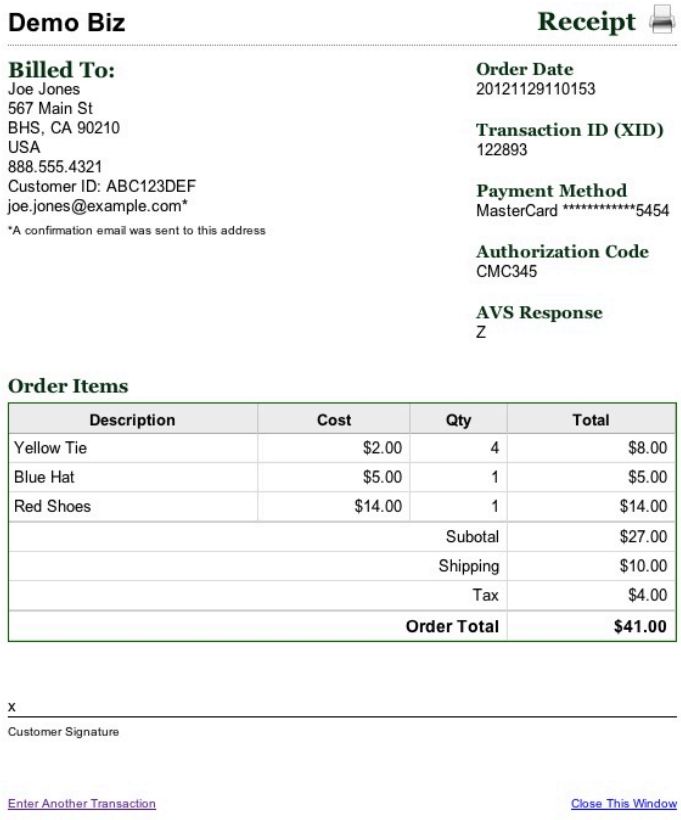
Processing Transactions

Submit a charge through this interface by swiping the card and entering correct data into the required fields and any of the desired optional fields. Be sure to double check the amount of the transaction. Click the `Process Payment` button. The transaction will be attempted in real-time. The gateway will either display an approval screen or a failure screen [17]. The failure screen will list the reason for the failure. An approval page will display information specific to the successful transaction.

2.6. Virtual Terminal Response Pages

All transactions submitted through any of the Virtual Terminal interfaces will either be successful or not. For your reference, here are examples of the response page for a successful transaction and one that failed.

Figure 2.7. Success: Receipt Page - Credit Card



Demo Biz — **Receipt**

Billed To:
Joe Jones
567 Main St
BHS, CA 90210
USA
888.555.4321
Customer ID: ABC123DEF
joe.jones@example.com*
*A confirmation email was sent to this address

Order Date
20121129110153

Transaction ID (XID)
122893

Payment Method
MasterCard **********5454

Authorization Code
CMC345

AVS Response
Z

Order Items

Description	Cost	Qty	Total
Yellow Tie	$2.00	4	$8.00
Blue Hat	$5.00	1	$5.00
Red Shoes	$14.00	1	$14.00
		Subotal	$27.00
		Shipping	$10.00
		Tax	$4.00
		Order Total	**$41.00**

x _____
Customer Signature

Enter Another Transaction Close This Window

- **Order Date** - This is a timestamp of the transaction.

- **Transaction ID** (XID) - This unique numeric code is assigned by the gateway as an identifier of the transaction. XID numbers can be used for simple searches in the transaction reporting tools.

- **Authorization Code** - This identifier can be a six to eight digit numeric or alpha-numeric value. This is assigned by the card issuing bank and passed back through the merchant account processor.

- **AVS Response** - This code is returned to indicate which pieces of an address match (or do not match) what is on file with the issuing bank's address verification database.

This page can be printed and used as a receipt (in addition to the emailed confirmation [66]) for your customers. A signature line is included if you need to capture your customer's signature.

Figure 2.8. Success: Receipt Page - Checks

Carl's Computers **Receipt**

Billed To:
Jane Johnson
4141 Rodeo Dr
BHS, CA 90210
USA
312.555.5825
Customer ID: QAZ678PLM
jane@example.com*
*A confirmation email was sent to this address

Order Date
20121129115433

Transaction ID (XID)
122909

Payment Method
Checking Account:
**********4365

Authorization Code
Unspecified

AVS Response
Unspecified

Order Items

Description	Cost	Qty	Total
Plaid Socks	$5.00	1	$5.00
		Subotal	$5.00
		Order Total	**$5.00**

x
Customer Signature

Enter Another Transaction Close This Window

A similar page will display when a transaction is successful. Many pieces of data that were entered in the interface will be shown here. This also shows the Transaction ID (XID) assigned by the gateway as the unique identifier of the transaction. XID numbers can be used for simple searches in the transaction reporting tools.

Figure 2.9. Failure - Credit Card and Check

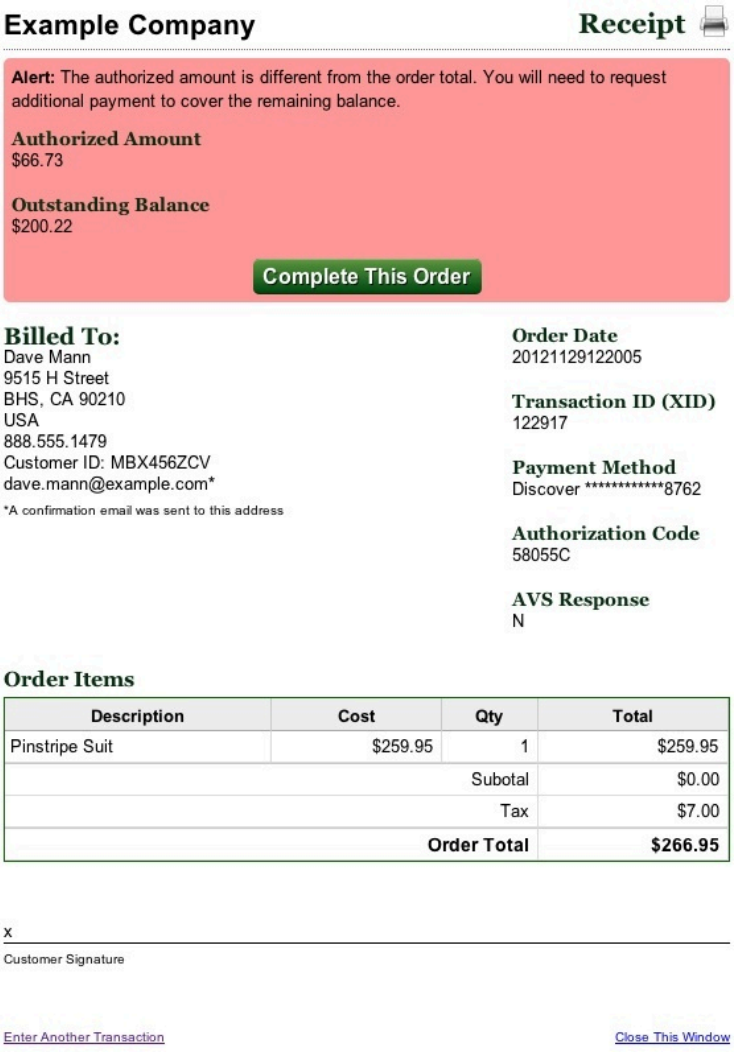
YOUR TRANSACTION COULD NOT BE PROCESSED AND HAS BEEN CANCELED.
THE FOLLOWING ERROR OCCURRED:

Code: VCC003 The credit card number entered has the wrong number of digits. Please verify.

If applicable, you may return to the previous page and correct the error.
To return to the previous page, press the BACK button on your browser (ALT+Left Arrow) or click here.

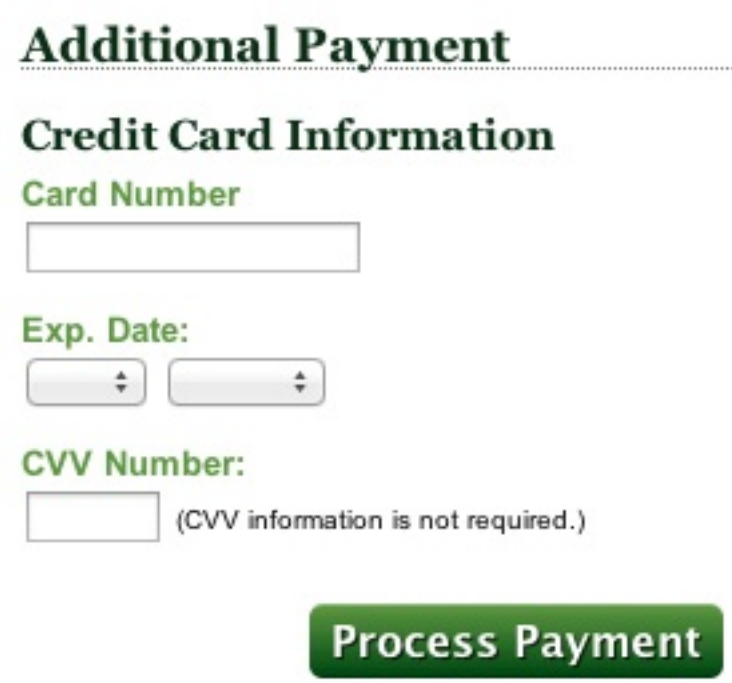
A similar page will display for a virtual transaction that is not successful for some reason. The blue and red text indicate that there has been some type of error or failure and the bold black text will display the specific error message to assist you in determining what the problem was. Many times, the text will offer suggestions as to what you should try instead. Often with a check failure in the Virtual Terminal, you will need to reload the interface page that you were using (rather than just hitting `Back` on your browser).

Figure 2.10. Partial Authorization: Receipt Page - Credit Card



A similar page will display when a transaction is partially authorized through a gateway set with a retail industry setting. By clicking on the `Complete This Order` button, this window will open and allow for you to enter a different card to capture the rest of the amount.

Figure 2.11. Partial Authorization: Additional Payment Entry - Credit Card



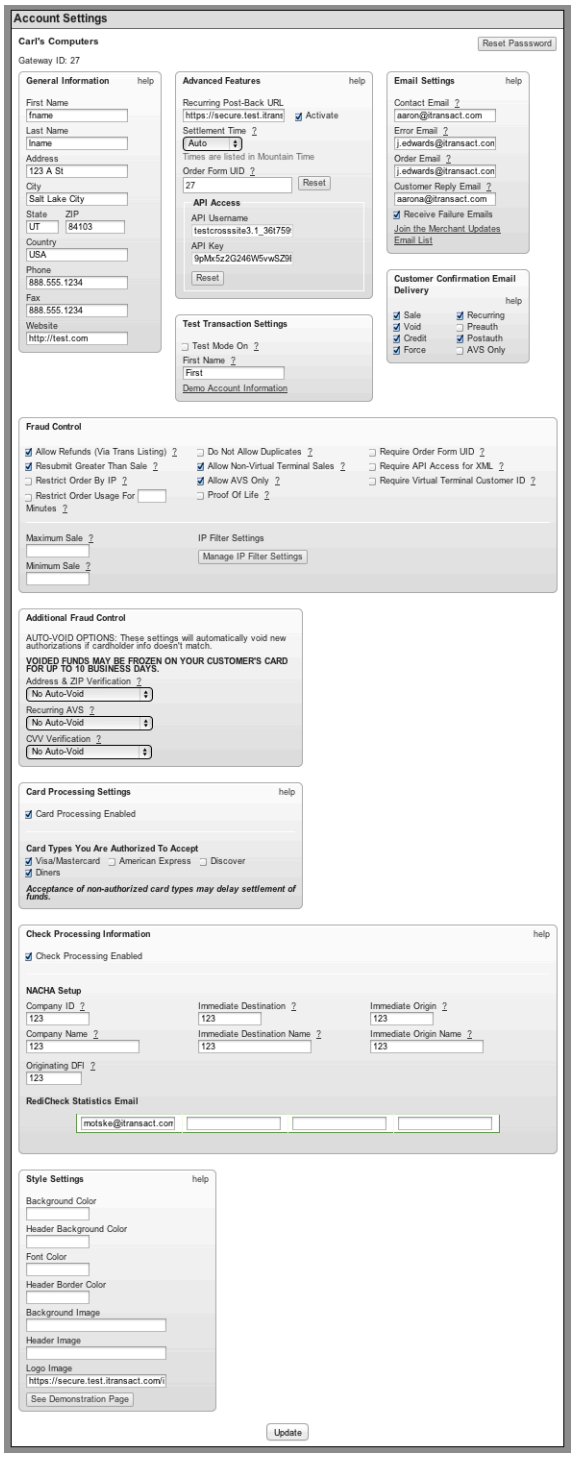
If no other payment option is available, you can run a void through the Transaction Options interface.

3. Account Settings

This area is used to view and/or update contact information, anti-fraud features, email settings, and general transaction functions. If you are in the Account Settings and you are wondering what a feature does, please click the question mark next to the item.

3.1. Overview

Figure 2.12. Account Settings Interface

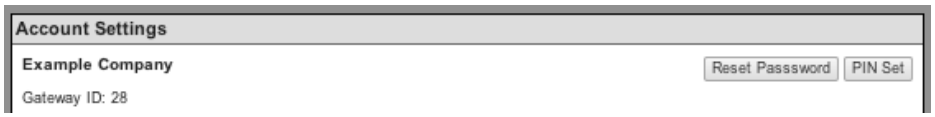


Merchants use the Account Settings interface to update contact information, anti-fraud features, email settings, and general transaction functions.

Please remember, any changes to data in this interface require the user to click the `Update` button at the bottom of the Account Settings interface.

3.2. Welcome Section

Figure 2.13. Account Settings - Welcome Section



This section lists the Business Name and the Gateway ID. Your account's gateway ID number will never change. To change the way your business name is listed here, please submit the request to our support team. This section also houses the following security code interfaces:

- **Password Change** - Use this to change your password. Click on the `Go` button to open the interface. Your new password must meet the following requirements:

  1. Minimum length: 8 characters
  2. Maximum length: 30 characters
  3. Must include at least two different character groups (lowercase letters, uppercase letters, digits, punctuation)
  4. Cannot include the same character repeated three times (like aaa or 555) or a sequence of three characters (like xyz or 789)
  5. Cannot include dictionary words, including common names

  To change the password, enter your then-current password into the `Current Password` field, enter your desired password in the other two fields, and then, click `Update`.

- **PIN Set/PIN Update** - Choose this to setup or change your PIN number for the Call-A-Charge [83] system. To setup a PIN code, click the `Go` button, enter your desired PIN code into the two fields, and click `Update`. To change the PIN code, click on the `Go` button to open the interface, enter your then-current PIN code into the `Current PIN`, enter your new desired PIN code in the other two fields. All PIN codes must be six numeric digits.

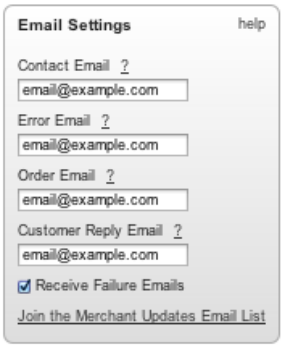
3.3. General Information Section

Figure 2.14. Account Settings - General Information Section

This section contains contact information for the merchant to be used by the gateway. Clicking `Help` will open a window that explains that the data in these fields should be edited if any of the information changes. If you modify any of this information, please remember to click `Update` at the bottom of the interface to save your changes.

3.4. Merchant Email Settings

Figure 2.15. Account Settings - Merchant Email Section



These settings determine what address is used to notify a merchant of sales, settlements, order form integration errors and gateway billings. These may all be different, or you may use the same address for each email type. The gateway supports delivery to only a single address. This means that only one address should be entered in each field. If you need more than one person to receive any of these notifications, please have your email provider or network administrator setup a multi-recipient alias address and use that address to populate the corresponding field. You may also access the *registration for the Merchant Updates opt-in email list*. Here is a more in-depth explanation of each of these settings:

- **Contact Email Address** - This address will be sent account activation information, any billings from iTransact, and settlements.

- **Error Email Address** - This address will be sent messages when there is an error in the order form scripting or gateway integration. Generally, emails are only sent to this address during the integration process. Error emails point out what fields are incorrect or missing. For ease of communication, consider having these sent directly to the person integrating or maintaining your web forms.

- **Order Email Address** - This address will receive email confirmations of all sales, voids, credits, forces, pre-authorizations, post-authorizations, recurring billings, and failure emails (if selected). There is no way to turn off these emails (except for the failures). The gateway is required to send these confirmations for a merchant's records. If a merchant decides not to do anything with these emails, they can setup a "garbage" email account and enter that address as the `Order Email Address`.

- **Customer Reply Email Address** - This address will be listed in the customer's copy of the email confirmation receipt as a contact address for the merchant.

- **Receive Failure Emails Checkbox** - Activate this setting to be sent an email anytime that a customer attempts to pay but is rejected for some reason. The email will explain the reason for the failure by listing the response from the credit card processing network.

- **Error Email Address** - This address will be sent messages when there is an error in the order form scripting or gateway integration. Generally, emails are only sent to this address during the integration process. Error emails point out what fields are incorrect or missing. For ease of communication, consider having these sent directly to the person integrating or maintaining your web forms.

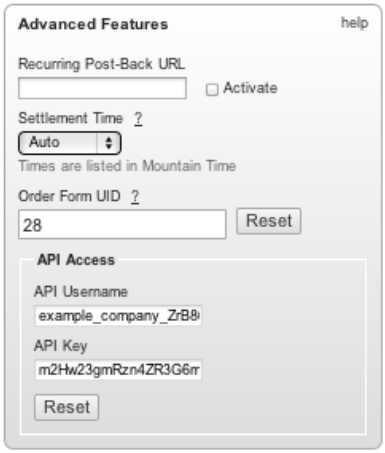
- **Customer Reply Email Address** - This address will be listed in the customer's copy of the email confirmation receipt as a contact address for the merchant.

- **The Merchant Updates Email List** - This link opens the interface that allows people to subscribe to the opt-in *Merchant Updates* email list and follow our Twitter feed (*iTransactStatus*). The mailing list is used to notify merchants of system maintenance, feature additions and other important information.

- **The Help Menu** - This link provides a list of the functions of each of the email settings.

3.5. Advanced Feature Settings

Figure 2.16. Account Settings - Advanced Feature Section



- **The Recurring Post-Back URL** - If using the recurring transaction features of the gateway, the merchant may specify a URL to receive transaction postback information generated at the time the transaction recurs. The `Activate` checkbox must be checked for this feature to function. For complete information concerning the Recurring Billing [146] system and the Recurring PostBack [60] feature, please review those sections of the documentation.

- **The Settlement Time Selector** - By default, this setting is `Auto`, meaning that the software will settle this account as the software cycles through all of the gateway accounts for settlement each night beginning at midnight. If you would like batch settlement to take place at a specific time each day, select the appropriate time. `Manual` should be selected by merchants who want to settle each of their own batches using the `Settle Now` tool the Control Panel.

- **The Order Form UID Value** - This is the preferred option to be used as the value for the `vendor_id` field in an html order form or shopping cart. You can change this value by clicking the `RESET` link.

  NOTE: *If your forms are using the UID instead of the Gateway ID and you reset this value, you MUST update the vendor_id value in your order form or shopping cart or your transactions will not process.*

If the XML connection is activated on a gateway account the following API Access fields will display:

- **Enable API Link** - This will display until API access that been enabled. It will be replaced with the credential fields once it has been selected.

- **API Username** - This is the username for people using the XML API Connection method [90]. You can change this value by clicking the `RESET` link.

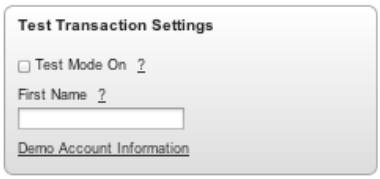
  NOTE: *You must update your scripts generating the XML requests if you reset the API Access information.*

- **API Key** - This is used to help generate the payload signature [91] for people using the XML API Connection method. You can change this value by clicking the `RESET` link.

  NOTE: *You must update your scripts generating the XML requests if you reset the API Access information.*

3.6. Test Transaction Settings

Figure 2.17. Account Settings - Test Transaction Section



Use these to test order forms and ordering system to make sure that web forms have been integrated with the gateway system correctly (without actually charging an account or generating a transaction record).

- **Test Mode Checkbox** - By default, this is unchecked. When checked, any transactions will run as test transactions and will not be processed or charged. Any transactions submitted as test transactions will not show up in the Transaction Listing [32] or Transaction Detail [40] interfaces.

  NOTE: *Do not activate this unless you intend to have all of your transactions be submitted as test transactions.*

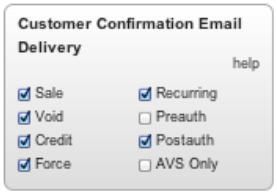
- **Test User First Name** - When the value in this field is passed as the customer's first name, the transaction will be processed as a test transaction. Do not use a real name, instead use something like "Test123". By default, this value is "First".

  NOTE: *This value is case-sensitive.*

- **Demo Account Information** - This link opens a window with account information to be used for form testing.

3.7. Customer Confirmation Email Settings

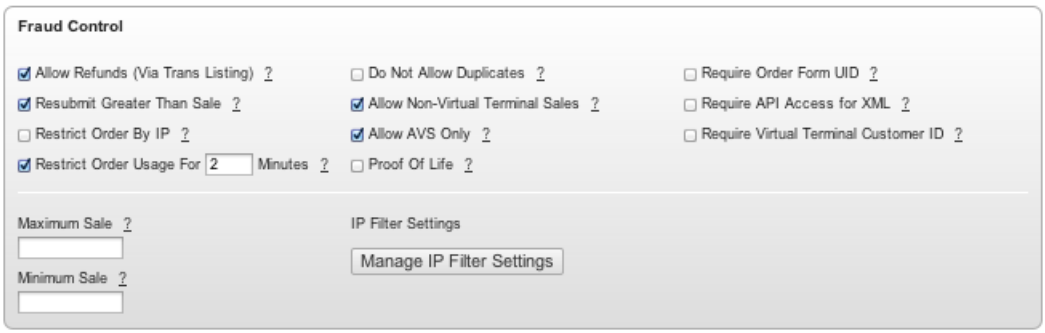
Figure 2.18. Account Settings - Customer Confirmation Email Section



Use this to select/deselect which emails [65] are sent to the email provided by the customer at the time of the transaction. Many merchants who use a shopping cart system prefer that the shopping cart sends a confirmation, rather than the gateway. In that case, they remove the check marks. All emails are set to be sent by default, except for the AVSOnly and PreAuth emails. The `Help` menu provides a description of each of the emails sent.

3.8. Fraud Control Settings

Figure 2.19. Account Settings - Fraud Control Section



These settings are provided as additional protection against potential fraud. Each of these features are optional. Use as many of these features as you choose. No feature can completely prevent all types of fraud, but these are some of the most powerful anti-fraud options available on any gateway software today.

- **Allow Refunds Via The Transaction Listing** - Enable this to refund past transactions through in the Transaction Listing or Transaction Detail Options [42] area without having to enter the customer's information.
- **Resubmit Greater Than Sale** - This setting allows you to generate a charge to a past customer [44] without having to re-enter the customer's information. This should be enabled if a you would ever possibly "re-bill" a past customer for an amount larger than the original sale. The feature does not effect recurring transactions, only resubmit transactions.
- **Restrict Order By IP** - Enable this to allow HTML-based [122] transactions only from specified IP addresses. This is activated and used by merchants who will be making all of their customers' order submissions from their own server directly to our system. If enabled, also enter acceptable IP addresses into the IP Filter Management window using the Transproc module for HTML transactions. It should not be activated if customers will be posting to the gateway system.
- **Restrict Order Usage** - This HTML fraud prevention module is specifically designed to reduce the number of “testers” hitting merchants with credit card numbers attempting to find valid cards. If a transaction is received and is NOT approved, a restriction will be automatically enabled. For the next *X* minutes no transactions will be allowed from the IP address of the original unapproved transaction. To activate this, click the checkbox and enter an amount of time in the minutes box.
- **Do Not Allow Duplicates** - This feature blocks duplicate transactions sent through the gateway. To be considered a duplicate transaction the following values must be identical to another successful transaction that has occurred in the last 24 hours. Currently, this service only works for non-XML based credit card transactions. The following fields are used to determine duplication:
  - Credit Card Number
  - Credit Card Expiration Month
  - Credit Card Expiration Year
  - Billing Street Address
  - Billing Zip/Postal Code
  - Order Total
- **Allow Non-Virtual Terminal Sales** - This must be selected if you will be accepting transactions via order form. The only instance in which a merchant would disable this feature would be if the merchant was only using the Virtual Terminal or XML [90] feature to manually enter transactions. If you are only accepting XML and Virtual Terminal transactions, please uncheck this box.
- **Allow AVS Only** - If this is selected (and your processor supports it) you can run zero amount authorizations to validate a credit card.
- **Proof of Life** - This HTML feature is commonly known as Captcha verification. When activated, a page is displayed to the user with a dynamically generated image containing random characters after order submission. The user must enter these characters correctly to complete the transaction submission. If a merchant is logged into an open session of the Control Panel, and submits a transaction through their website, it will not prompt the merchant for the entry. However, if the merchant does not have an open session, they will be prompted for the entry - like a customer.
- **Require Order Form UIDs** - This fraud prevention option gives merchants the ability to enforce the use of the Order Form UID [23] for the value of the vendor_id field on HTML based order forms. The value of the UID [23] is listed in the Account Settings. The fifteen to twenty character Order Form UID value is randomly generated and keeps fraudsters from guessing what ID you are using to process orders. This does not keep someone from looking at your order form and obtaining this value, but we have found that the majority

of fraud is done using automated tools. This setting keeps fraudsters from stumbling upon your account. You may reset the value as often as you like.

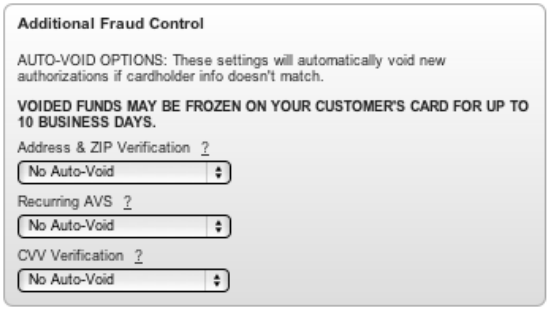
- **Require API Access for XML** - If this is enabled, the gateway will only recognize XML submissions that are using the API credentials.
- **Require Virtual Terminal Customer ID** - This Virtual Terminal module allows you to enforce the Customer ID [41] value for all Virtual Terminal transactions if you wish to do so.
- **Maximum Sale** - Use this to place a cap on the highest amount that you will allow to process through your gateway. Often, a merchant will set this amount to coincide with the “maximum volume ticket limit” imposed on them by their credit card merchant processing bank. A customer who attempts to place a transaction which exceeds that amount will be shown an error which indicates that they've entered an invalid amount.
- **Minimum Sale** - This allows you to place an "at least" amount. A customer who attempts to place a transaction which does not meet at least that amount, will be shown an error which indicates that they've entered an invalid amount.
- **IP Filter Settings Interface** - Access the IP Filter Management window by clicking the `Go` button.

Figure 2.20. IP Filter Settings Interface

The filter allows a HTML [122] or XML [171] connection method users to limit transaction submissions to the IP address/range of a specific server. This is required for use of the deprecated XML connection method and optional for HTML users. If used with HTML, be sure to activate the Restrict Order By IP feature. The *Transproc Module* is used for filtering HTML based transactions [122]. The *XMLTrans* module is used for filtering the deprecated XML method [171]. The status must be set to `Active`. To delete an IP address, click the `Go` button in the `Delete` column.

3.9. Auto-Void Settings

Figure 2.21. Account Settings - Check Processing Section



Auto-Void Options - These features will automatically void credit card transactions that would otherwise be charged successfully based on your own specified risk tolerance.

- **Address and ZIP Verification (AVS) Auto-Void** - All of the major credit card processors will approve transactions that do not have matching address and/or zip codes. The fact that processors do not reject non-matching AVS transactions is a great concern of ours. Our system can void transactions with non-matching AVS data, even though they had been approved through the processor. Please understand that the gateway does not provide the AVS responses [32]. Those responses are generated by the credit card issuing bank and reported by the credit card processor based on information located in the bank's AVS database (which may or may not match the bank's statement database). However, the gateway system will perform the auto-void according to the requirements set by the merchant. These settings may be modified in the Account Settings as needed. Some foreign credit card issuers do not support AVS, so use of the AVS auto-void may be impractical if you are accepting international payments. In each case, an auto-void will attempt to reverse the approval and release the “frozen” funds back to the cardholders available credit line. For an unsuccessful reversal attempt a(n) void/auto-void of an authorized transaction cancels the charge, but does not cancel an authorization. An authorization freezes funds in an account, so that a completed charge can withdraw those frozen funds. A voided authorization may "freeze" the funds in the customer's account for up to 10 days. The following levels are available:

  - No Auto-Void - This will allow any approved transaction to process regardless of the address verification response.

  - Void Unless ZIP Matches - This will void any approved transaction for which the processor indicates that the ZIP Code entered does not match the ZIP Code listed in the bank's AVS database (even if the street address matches).

  - Void Unless Addr Matches - This will void any approved transaction for which the processor indicates that the street address entered does not match the street address listed in the bank's AVS database (even if the ZIP Code matches).

  - Void Unless Both Match-This setting requires that both the address and the ZIP Code match exactly what the issuing bank's AVS database has on file for the customer. If either the address or ZIP Code, or both, come back as a non-match, that approved transaction will be voided.

- **Recurring AVS Auto-Void** - This feature provides auto-voiding of recurring transactions based on the address and ZIP Code verifications [32] returned by the processing network. Please understand that the gateway does not provide the AVS responses. Those responses are generated by the credit card issuing bank and reported by the credit card processor based on information located in the bank's AVS database (which may or may not match the bank's statement database). However, the gateway system will perform the auto-void according to the requirements set by the merchant. These settings may be modified in the Account Settings as needed. Some foreign credit card issuers do not support AVS, so use of the AVS auto-void may be impractical if you are accepting international payments. Keep in mind, a(n) void/auto-void of an authorized transaction cancels the charge, but does not cancel an authorization. An authorization freezes funds in an account, so that a completed charge can withdraw those frozen funds. A voided authorization may "freeze" the funds in the customer's account for up to 10 days. The following levels are available:

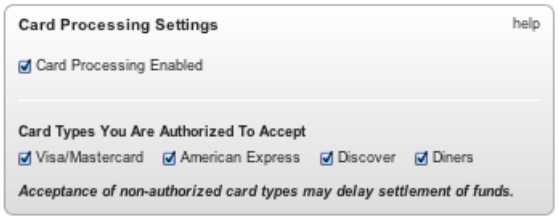
- No Auto-Void - This will allow any approved transaction to process regardless of the address verification response.
- Void Unless ZIP Matches - This will void any approved transaction for which the processor indicates that the ZIP Code entered does not match the ZIP Code listed in the bank's AVS database (even if the street address matches).
- Void Unless Addr Matches - This will void any approved transaction for which the processor indicates that the street address entered does not match the street address listed in the bank's AVS database (even if the ZIP Code matches).
- Void Unless Both Match-This setting requires that both the address and the ZIP Code match exactly what the issuing bank's AVS database has on file for the customer. If either the address or ZIP Code, or both, come back as a non-match, that approved transaction will be voided.

- **CVV Verification Auto-Void** - The CVV code is a security feature for “card not present” transactions (e.g., Internet transactions), and now appears on most (but not all) major credit and debit cards. This feature is a three or four digit code which provides a cryptographic check of the information embossed on the card. Therefore, the CVV code is not part of the card number itself. This setting automatically voids approved if the processing network indicates that the CVV [35] entered does not match the CVV database at the customer's credit card issuing bank. Most issuing banks do not require a CVV number to be entered for a transaction to process. A small group of banks do require correct CVV entry for Internet based transactions. The gateway system will perform the auto-void according to the requirements that you deem necessary. Modify these settings as needed. Keep in mind, a(n) void/auto-void of an authorized transaction cancels the charge, but does not cancel an authorization. An authorization freezes funds in an account, so that a completed charge can withdraw those frozen funds. A voided authorization may "freeze" the funds in the customer's account for up to 10 days. The following levels are available:
  - No Auto-Void - This will allow any approved transaction to process regardless of the address verification response.
  - Void Unless CVV Matches - This setting will void any authorized transaction which is returned with a non-matching or empty CVV response.
  - Void If CVV Not Entered - With this setting a customer's transaction will be voided if the bank indicates that a CVV code should exist on the card, but was not entered.

  NOTE: *The iTransact gateway does not perform CVV auto-voids for American Express transactions.*

3.10. Card Processing Settings

Figure 2.22. Account Settings - Card Processing Section

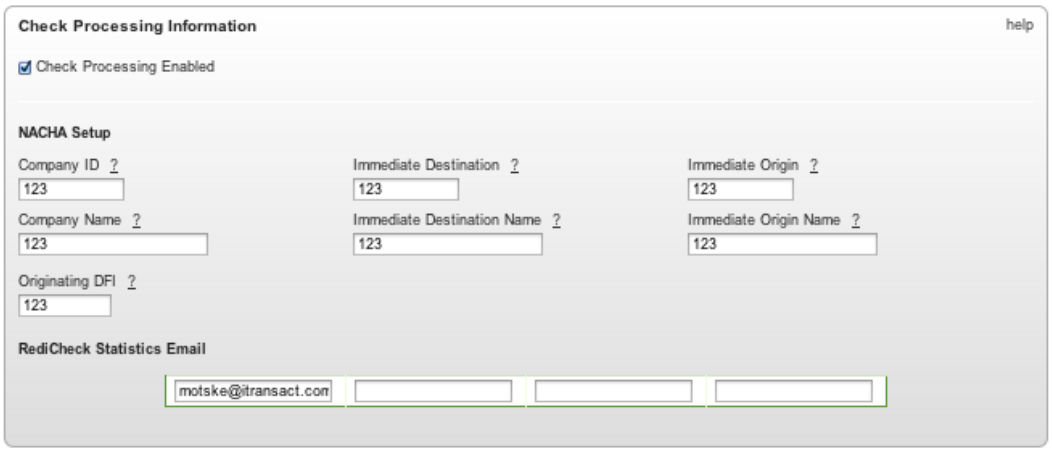


This area will only display if the Card Setup [64] has been completed for your account. If you would like the the card acceptance script to display on a Split Form [129], click the `Card Processing Enabled` check-box. By default, all US based credit card merchant accounts are established to process transactions on Visa and MasterCards. If a merchant has also established ‘appendage’ accounts (i.e. AMEX, Discover, and Diners), a merchant must provide those merchant IDs to their Visa/MC merchant provider. Once that's complete, a merchant can update the Card Type settings by clicking on the appropriate card types. A merchant can disable a card type by unchecking the card type in the same area. The selected card types will display on the secure half of the form (if a merchant uses that method).

NOTE: *Do not enable any card type that you have not established a corresponding merchant account for. Accepting a transaction from a card type you are not authorized to accept may delay the settlement of the rest of your transactions.*

3.11. Check Processing Settings

Figure 2.23. Account Settings - Check Processing Section



This area will only display if the check setup has been completed for your gateway. If you would like the check acceptance script to display on a Split Form [129], click the `Check Processing Enabled` checkbox. For merchants utilizing the RediCheck system for accepting check payments, up to four email addresses can receive the daily Check Statistics [80] emails. Please enter the desired recipients in the interface fields in the Account Settings of the Control Panel.

If you are authorized to use the NACHA processing system, the NACHA Setup fields will display in this section. To be authorized to run NACHA transactions, you need to first establish a NACHA processing account at your deposit bank. Many banks refer to these accounts as “Treasury Service” accounts. When the account is established at your bank, they will need to provide you with a list of seven identifiers that are specific to your account and the bank. Enter the values provided by your NACHA processor in the Check Processing settings. The required information is:

- **Company ID** - Your 10-digit company number assigned by your NACHA processor
- **Company Name** - Your company's name as listed with your NACHA processor
- **Immediate Destination** - Your bank's ABA transit routing number
- **Immediate Destination Name** - Your bank's name
- **Immediate Origin** - Your 10-digit origination number (may be same number as Company ID)
- **Immediate Origin Name** - Your company's origination name (may be same as Company Name)
- **Originating DFI** - Your bank's origination number

NOTE: *Each bank may use slightly different information for the above fields. In fact, they may call the fields by different names, but every NACHA file requires these pieces of data.*

3.12. Style Settings

Figure 2.24. Account Settings - Style Section

The section is used to make the secure page of the Split Form [129] or BuyNow [135] format to appear the way that you would like it to. This can be helpful in making this type of transaction appear more seamless. Use of any of the supported HTML body tags [133] will supersede any setting here. These are the options that can be set:

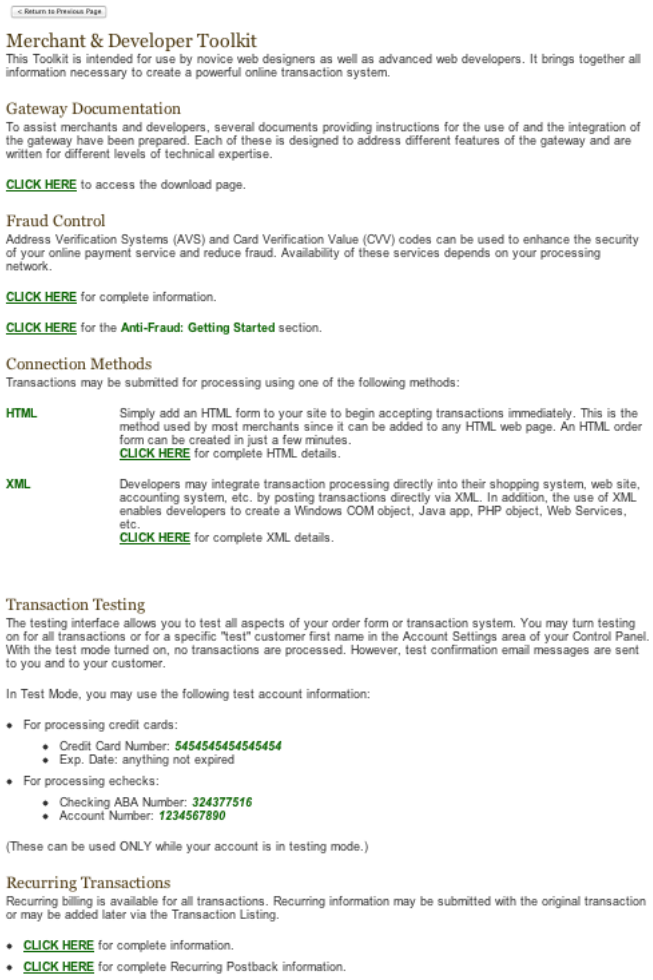
- **Background Color** - This can be a six digit hexadecimal value.
- **Font Color** - This can be a six digit hexadecimal value.
- **Header Background Color** - This can be a six digit hexadecimal value.
- **Background Image** - This needs to be the absolute URL of an image. You can upload this image to iTransact's secure server by submitting a ticket at http://support.itransact.com.
- **Header Border Color** - This can be a six digit hexadecimal value.
- **Header Image** - This needs to be the absolute URL of an image. You can upload this image to iTransact's secure server by submitting a ticket at http://support.itransact.com.
- **See Demonstration Page Link** - This will show an example of what the layout of the form will look like. The example page will be populated with Latin gibberish text, but will give you an idea of how the color scheme will work with your images. This will only display settings that have been entered and submitted by the pressing of the `Update` button at the bottom of the interface.

3.13. Updating Your Changes

The `Update` button must be clicked anytime you make any changes in the Account Settings interface. Please remember to click `Update` after any modification. If you do not click the `Update` button, your changes will not be saved.

4. Merchant Toolkit

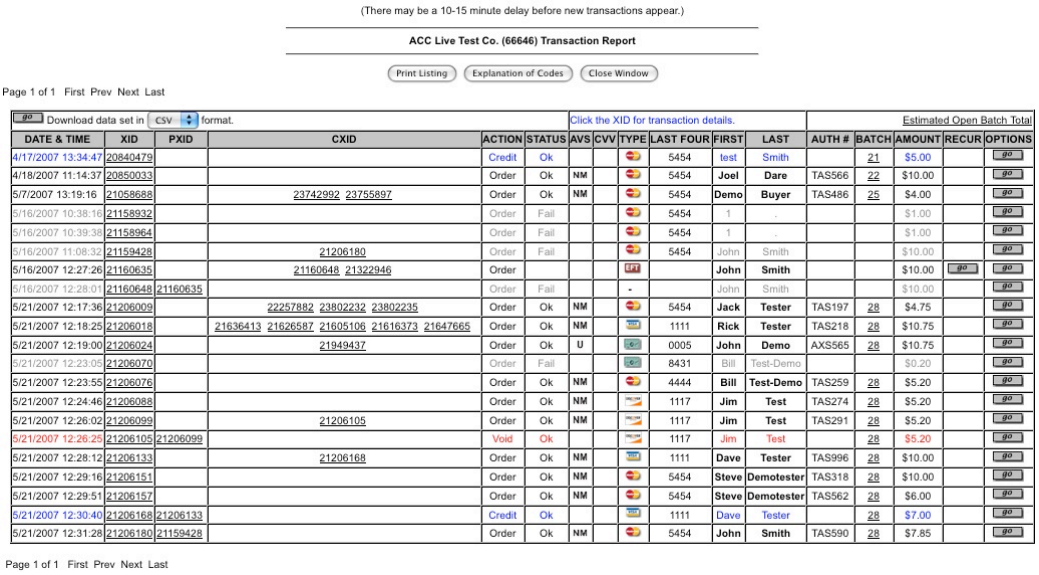
Figure 2.25. Merchant Toolkit



The *Merchant Toolkit* is a repository of all of the documentation, integration, and user instructions for the gateway. In-depth explanations of the anti-fraud tools are available here. It includes examples and links to libraries that can be used for linking your gateway to your web site. Testing information is available for credit card and check transactions that can be used to test forms.

5. Transaction Listing

Figure 2.26. Transaction Listing



(There may be a 10-15 minute delay before new transactions appear.)

ACC Live Test Co. (66646) Transaction Report

Print Listing | Explanation of Codes | Close Window

Page 1 of 1 First Prev Next Last

go Download data set in CSV format. Click the XID for transaction details. Estimated Open Batch Total

DATE & TIME	XID	PXID	CXID	ACTION	STATUS	AVS	CVV	TYPE	LAST FOUR	FIRST	LAST	AUTH #	BATCH	AMOUNT	RECUR	OPTIONS
4/17/2007 13:34:47	20840479			Credit	Ok				5454	test	Smith		21	$5.00		go
4/18/2007 11:14:37	20850033			Order	Ok	NM			5454	Joel	Dare	TAS566	22	$10.00		go
5/7/2007 13:19:16	21058688		23742992 23755897	Order	Ok	NM			5454	Demo	Buyer	TAS486	25	$4.00		go
5/16/2007 10:38:16	21158932			Order	Fail				5454	1	-			$1.00		go
5/16/2007 10:39:38	21158964			Order	Fail				5454	1	-			$1.00		go
5/16/2007 11:08:32	21159428		21206180	Order	Fail				5454	John	Smith			$10.00		go
5/16/2007 12:27:26	21160635		21160648 21322946	Order						John	Smith			$10.00	go	go
5/16/2007 12:28:01	21160648	21160635		Order	Fail				-	John	Smith			$10.00		go
5/21/2007 12:17:36	21206009		22257882 23802232 23802235	Order	Ok	NM			5454	Jack	Tester	TAS197	28	$4.75		go
5/21/2007 12:18:25	21206018		21636413 21626587 21605106 21616373 21647665	Order	Ok	NM			1111	Rick	Tester	TAS218	28	$10.75		go
5/21/2007 12:19:00	21206024		21949437	Order	Ok	U			0005	John	Demo	AXS565	28	$10.75		go
5/21/2007 12:23:05	21206070			Order	Fail				8431	Bill	Test-Demo			$0.20		go
5/21/2007 12:23:55	21206076			Order	Ok	NM			4444	Bill	Test-Demo	TAS259	28	$5.20		go
5/21/2007 12:24:46	21206088			Order	Ok	NM			1117	Jim	Test	TAS274	28	$5.20		go
5/21/2007 12:26:02	21206099		21206105	Order	Ok	NM			1117	Jim	Test	TAS291	28	$5.20		go
5/21/2007 12:26:25	21206105	21206099		Void	Ok				1117	Jim	Test		28	$5.20		go
5/21/2007 12:28:12	21206133		21206168	Order	Ok	NM			1111	Dave	Tester	TAS996	28	$10.00		go
5/21/2007 12:29:16	21206151			Order	Ok	NM			5454	Steve	Demotester	TAS318	28	$10.00		go
5/21/2007 12:29:51	21206157			Order	Ok	NM			5454	Steve	Demotester	TAS562	28	$6.00		go
5/21/2007 12:30:40	21206168	21206133		Credit	Ok				1111	Dave	Tester		28	$7.00		go
5/21/2007 12:31:28	21206180	21159428		Order	Ok	NM			5454	John	Smith	TAS590	28	$7.85		go

Page 1 of 1 First Prev Next Last

The Transaction Listing provides easy access to information about transactions. You can use this tool to look at transactions that occurred during a date range, or to further drill down using specific search criteria.

5.1. Standard Listing

Select a date range and click Search to open the standard display of the Transaction Listing. Across the top of the listing, your company's name and gateway ID is displayed. Each of the items across the top have important features:

- **Page Links** - These indicate how many pages of transactions a date range includes. Each listing page lists up to 100 transactions. The pages may be moved through by clicking on the `First`, `Prev`, `Next`, and `Last` buttons at the top or the bottom of the page.
- **Print Listing Button** - This instructs a computer to print the page being viewed.
- **Explanation of Codes Button** - This opens a window explaining the information displayed in the Status, AVS, and CVV columns.
- **Close Window Button** - This closes the Transaction Listing window.
- **Download Data Options** - By selecting `CSV`, `QIF`, or `XML` in the drop down menu and clicking the `Go` button, a file will download that includes the details of all of transactions in the opened date range.
- **Estimated Open Batch Total Link** - This link opens a window that displays the then-current batch total based on information in the gateway. This may or may not match with the batch amount on the processor's system. If you change to a new credit card merchant account this will display a historical record of all card transactions processed through the previous account until a batch settles through the new account.
- **Date and Time** - The value in this column is the date and time of the original transaction attempt.
- **XID** - This value is the numeric identifier assigned automatically to a transaction when it is submitted through the gateway system. By clicking on the XID number, the Transaction Detail [40] for that transaction will open. For an explanation of the Transaction Detail window, please see the section detailing this feature [40].
- **PXID** - This value is the parent transaction ID, or the XID, of the transaction which is the originating transaction of a recurring payment, a void, a credit/refund, a postauth, or a resubmitted transaction. This field will be blank unless the transaction has an originating transaction. When a merchant clicks on the PXID number, the Transaction Detail [40] for that transaction will open. For an explanation of the Transaction Detail window, please see the section detailing this feature [40].

- **CXID** - This value is the child transaction ID, or the XID, of the transaction which has used this transaction as the originating transaction for a recurring payment, a void, a credit/refund, a postauth, or a resubmitted transaction. This field will be blank if the transaction has no child transactions. When a merchant clicks on the CXID number, the Transaction Detail [40] for that transaction will open. For an explanation of the Transaction Detail window, please see the section detailing this feature [40].
- **Action** - This field identifies what type of transaction was submitted.
- **Status** - The field identifies whether a transaction was successful or not. Potential values for credit card transactions are listed below:
  - Ok - Valid Transaction
  - Incomplete - Generally indicates transaction is currently in process
  - Error - Error receiving response from network. May be caused by user error or an unknown response during transaction processing.
  - Fail - Could be any of the following reasons:
    - Declined
    - Card Expired
    - Invalid Card Number
    - Lost or Stolen Card
    - Card Type Not Supported
    - Service Not Allowed
    - User Input Data Error
    - No Answer From Processing Network
    - No Response From Processing Network

  Review the Error Message [41] field in the Transaction Detail for the specific reason for failure. See the section detailing this [178] to review the possible response values.

  NOTE: *EFT transactions do not display a value for the Status field.*
- **AVS** - This column is blank for check or EFT transactions. The value is the response received from the credit card processing network to indicate whether the address and ZIP Code matched the account's address on file in the bank's AVS database or not. Different processors have different values. To keep the listing simple, the listing shows 6 easy to understand categories (A&Z - address and ZIP match, ZIP - address does not match but ZIP does, Addr - address matches but ZIP does not, GNP - Global Non-Participator/foreign card can not be verified, U - unsupported AVS, and N - neither address nor ZIP matches). Those simple categories include several types of responses generated from the processors. Here are the possible responses from the different processors:

Table 2.1. AVS Responses: Elavon/Nova - All card types

A	Address match. ZIP does not match.
D	Address and postal match.
E	AVS error.
G	Global non-participant. AVS not available for international customers.
I	International address not verified.
N	No match. Neither address nor ZIP match.

O	No response to AVS request.
S	AVS service not supported for this transaction.
U	AVS unavailable.
Y	Address and 5-digit ZIP Code match.
Z	ZIP Code match. Address does not match.

Table 2.2. AVS Responses: TSYS/Visanet/VITAL - All card types

A	Street address matches. ZIP Code does not.
E	AVS error. No service.
N	No match. Neither address nor ZIP match.
R	Retry. AVS not available.
S	No AVS service.
U	Unavailable.
Y	Street address and 5-digit ZIP Code match.
Z	5-digit ZIP Code matches. Street address does not.

Table 2.3. AVS Responses: Paymentech - Visa

A	Address matches, ZIP Code does not.
E	Transaction is ineligible for address verification.
G	Global non-participant. Foreign card. Cannot verify address.
N	Neither the ZIP nor the address matches.
R	Issuer's authorization system is unavailable. Try again later.
S	AVS not supported at this time.
U	Unable to perform address verification because either address information is unavailable or unsupported by the Issuer.
Y	Address & 5-digit or 9-digit ZIP match.
Z	Either 5-digit or 9-digit ZIP matches. Address does not.

Table 2.4. AVS Responses: Paymentech - Discover

A	ZIP and address both match.
G	Global non-participant. Foreign card. Cannot verify address.
N	Neither the ZIP nor the address matches.
U	Unable to verify address.
W	Cardholder record.
Y	Address only matches.
Z	Zip only matches.

Table 2.5. AVS Responses: Paymentech - American Express

A	Address matches, ZIP Code does not.
G	Global non-participant. Foreign card. Cannot verify address.
N	Neither the ZIP nor the address matches.
R	Issuer's authorization system is unavailable. Try again later.

S	AVS not supported at this time.
U	The necessary information is not available. Account number is neither U.S. nor Canadian.
Y	Yes, address and ZIP Code are both correct.
Z	Zip Code only is correct.

Table 2.6. AVS Responses: Paymentech - MasterCard

A	Address matches, ZIP Code does not.
G	Global non-participant. Foreign card. Cannot verify address.
N	Neither the ZIP nor the address matches.
R	Retry. System unable to process.
S	AVS not supported at this time.
U	No data from Issuer/BankNet switch.
W	9-digit ZIP Code matches, but address does not.
X	Exact, all digits match, 9-digit ZIP Code.
Y	Exact, all digits match, 5-digit ZIP Code.
Z	5-digit ZIP Code matches, but address does not.

Table 2.7. AVS Responses: First Data - Visa/MasterCard/American Express

A	Address Match Only.
B	Address match for non-US transaction. Postal code not verified.
C	Address & postal code not verified for non-US transaction.
D	Address & postal code match for non-US transaction.
E	Address Verification Not Allowed for Card Type.
G	Global Non-AVS participant. Non-U.S. Card issuing bank does not support AVS.
I	Address information not verified for non-US transaction.
M	Address and postal code match for non-US transaction.
N	Neither the Address nor ZIP Match.
P	Postal code match for non-US transaction. Address not verified.
R	Retry/System unavailable.
S	Service not supported.
U	Address information not verified for domestic transaction.
W	9-digit ZIP match only.
X	Address & 9-digit ZIP Match.
Y	Address & 5-Digit ZIP Match.
Z	5-Digit ZIP Match Only.

Table 2.8. AVS Responses: First Data - Discover

A	Address & 5-Digit ZIP Match.
B	Address match for non-US transaction. Postal code not verified.
C	Address & postal code not verified for non-US transaction.
D	Address & postal code match for non-US transaction.
E	Address Verification Not Allowed for Card Type.

G	Global Non-AVS participant. Non-U.S. Card issuing bank does not support AVS.
I	Address information not verified for non-US transaction.
M	Address and postal code match for non-US transaction.
N	Neither the Address nor ZIP Match.
P	Postal code match for non-US transaction. Address not verified.
R	Retry/System unavailable.
S	Service not supported.
U	Address information not verified for domestic transaction.
W	Card number not on File.
Y	Address match only.
Z	5-Digit ZIP Match Only.

Table 2.9. AVS Responses: NDC Global - All card types

A	Address Match Only.
B	Incompatible formats (postal code). Street address match. Postal code not verified.
C	Incompatible formats (all information). Street address & postal code not verified.
D	Street address & postal codes match.
E	Edit error. For example, AVS not allowed for this transaction.
G	Global Non-AVS participant. Non-U.S. Card issuing bank does not support AVS.
I	International Transaction. Address information not verified.
M	Match. Street address and postal codes match.
N	No. Address and ZIP Code do not match.
P	Postal codes match. Street address not verified.
R	Retry. System unavailable or timed-out.
S	Service not supported. Issuer does not support AVS.
U	Unavailable. Address information not verified for domestic transactions.
W	Whole ZIP. 9-digit ZIP match only.
X	Exact. Address & 9-digit ZIP Match.
Y	Yes. Address & 5-Digit ZIP Match.
Z	ZIP. Five digit ZIP Code matches. Address does not match.

- **CVV Response** - This field is blank for check or EFT transactions. The value is the response received from the processing network to indicate whether the CVV/CID/CVV2 code matched or not. These are the possible responses:

Table 2.10. CVV Responses

M	Match
N	No Match
P	Not Entered/Processed
S	CVV should be present, but was not entered/processed.
X	No response from card issuer.
U	Unavailable. Card issuer not certified for CVV.

- **Type** - This displays the payment instrument/card type used for the transaction.

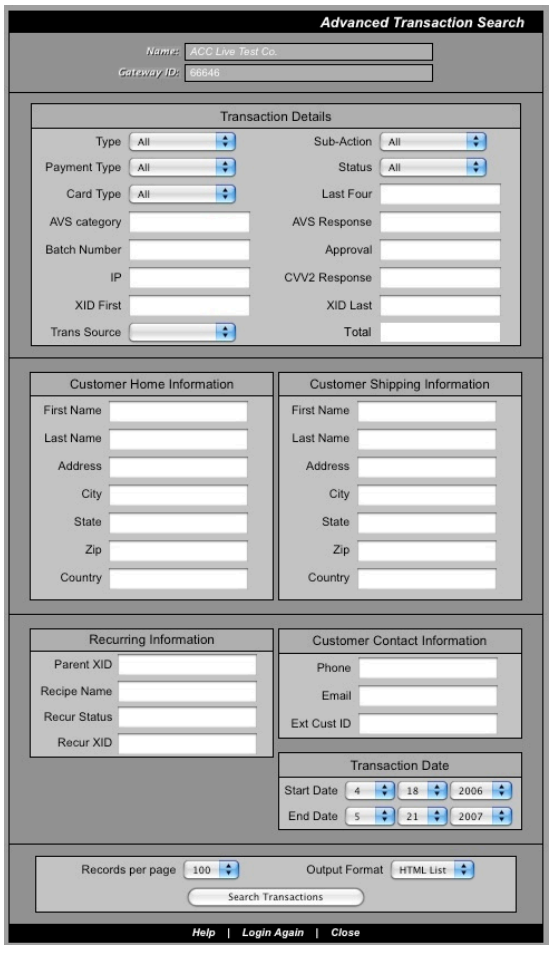
- **First Name** - This value is the customer's first name as submitted with the transaction information.
- **Last Name** - This value is the customer's last name as submitted with the transaction information.
- **Auth #** - This is the authorization code issued by the credit card issuing bank for approved transactions.
- **Batch** - This number indicates the batch that the transaction is a part of. This is not listed for EFT transactions.
- **Amount** - This value is the total amount that was billed to a customer's account. If the transaction was a partial authorized retail swiped/keyed transaction, the system will display transaction amounts in this format:

  $XXX.xx/XXX.xx

  The first of these numbers is the dollar amount actually authorized; the second number is the total dollar amount attempted.
- **Recur** - The `Go` button in this column opens the Recurring Detail [44] window where you can modify the recurring commands for a transaction.
- **Options** - The *Go* button in this column opens the Transaction Options [42] window where a merchant can run refunds, voids, forces, resubmits, postauths, and resend emails.

5.2. Advanced Transaction Search

Figure 2.27. Advanced Transaction Search Tool



To use this tool, click the `Advanced` checkbox, set the date range, and click the `Go` button. As many (or as few) of the search criteria can be used for a single request. The more criteria entered, the more specific the search will be.

When doing any type of search, be sure to enter a valid date range in the Transaction Date Information [39] section. For a search which yields a wider result, use fewer search criteria. Each section of the interface has different types of search criteria. You can use multiple items in any of the sections as needed.

5.2.1. Transaction Details

The items in this section allow you to search specific to the account and the processor responses.

- **Type** - Use this drop down to view a certain kind of transaction (i.e. `Order`, `Credit`, `Void`, `Postauth`). This defaults to `All` and allows for a search for any transaction type.
- **Payment Type** - This identifies whether a merchant is searching for a credit card payment, check payment, or EFT payment. This defaults to `All` and will search for any payment type.
- **Card Type** - This is used when a merchant needs to search for transactions run on a specific card type (i.e. `American Express`, `Discover`, `Diners Club/Carte Blanche`, `MasterCard`, `Visa`). This defaults to `All` and allows for a search for any card transactions.
- **AVS Category** - This allows for a search of transactions with a specific AVS category.
- **Batch Numbers** - Use this feature to pull open a list of all transactions in a specific batch.
- **IP** - This is used to search for transactions generated from a specific IP address.
- **Trans Source** - Use this to search for transactions generated from a specific source (i.e. `Auto-Voided`, `Web Form`, `Phone System`, `Recurring`, `Virtual Terminal`, *XML*).
- **Sub-Action** - This allows for a search of different types of order transactions (i.e. `Sale`, `Force`, `PreAuth`). This defaults to `All` and will search for any type of order transaction.
- **Status** - This allows for a search of transactions using the status as the criteria (i.e. `OK`, `Failure`, `Error`, `AVS Failure`, `Unknown Error`, `Incomplete`). This defaults to `All` and will search for a transaction with any status.
- **Last Four** - This is used to search for transactions which were generated from an account ending in a specific last four digits of the account.
- **AVS Response** - This allows for a search of transactions with a specific AVS response.
- **Approval** - This is used to search for a transaction with a specific authorization code.
- **CVV2 Response** - This allows for a search of transactions with a specific CVV response.
- **XID First** - This is the first entry if doing a search of all transactions in an XID range.
- **XID Last** - This is the ending entry if doing a search of all transactions in an XID range.
- **Total** - This is total amount of a transaction.

5.2.2. Customer Home Information

The items in this section enable you to search customer information that was submitted at the time of the transaction. This is not necessarily the cardholder information.

- **First Name** - This can be used to search for all transactions generated using a specific customer first name.
- **Last Name** - This can be used to search for all transactions generated using a specific customer last name.
- **Address** - This allows for a search for transactions using a specific street address.
- **City** - This will search for transactions submitted from a specific shipping city.

- **State** - This allows for transactions to be searched for using the shipping state as the criteria.
- **Zip** - This searches transactions generated using a specific shipping ZIP Code.
- **Country** - This allows a merchant to search for all transactions submitted using a specific shipping country.

5.2.3. Recurring Information

- **Parent XID** - This is used to search for any transactions that were generated as child transactions from the same parent transaction.
- **Recipe Name** - This allows for a merchant to find all transactions associated with a specific recurring recipe [61].
- **Recur Status** - This searches for recurring transactions based on the status of the transaction (i.e. `Begun`, `Error`, `Complete`).
- **Recur XID** - This searches by the Recurring XID.

5.2.4. Customer Contact Information

- **Phone** - This allows a search of transactions by phone number.
- **Email** - A merchant can search for all transactions submitted with a specific email address.
- **Ext Cust ID** - This can be used to search for a specific transaction that was submitted with the cust_id field.

5.2.5. Transaction Date Information

This section is the cornerstone of the entire search mechanism. When doing any type of search, enter a valid date range. The date range only allows a merchant to view a maximum of two calendar years in one search.

5.2.6. Output Information

Use this to specify how much information and in what format the search data is displayed.

- **Records Per Page** - This is used to display the number of transactions in the search display. The default value is `100` transactions, but alternate values are `25`, `50`, `250`, `500`.
- **Output Format** - This is used to create different types of files for the data to display. Here are the options:
  - HTML List - This is the default display. It will pull open an HTML page that displays the information in a version much like the Standard Transaction Listing.
  - XML List - This will download the data in an XML file.
  - CSV List - This will download the data in a CSV (comma separated value) file. Microsoft Excel or similar applications can be used to view the CSV file.

5.3. Retrieval of Saved Meta-Data

The Save Meta-Data function enables you to save additional transaction data on the gateway. This feature is typically used to save fields that are not normally saved in the gateway's database (such as the fields requested using the Passback [143] function). For HTML transactions, use the save [126] command to tell the gateway which additional fields you would like recorded. For XML transactions, use the VendorData element set. The recorded data can be used to search for transactions using the advanced transaction search and is displayed in the transaction details page. This data is retrieved in the XML download report. While in test mode [24], no meta-data is saved. Data is only saved when a transaction processes live.

6. Transaction Detail Window

Figure 2.28. Transaction Detail Window

The data for a specific transaction is accessed by entering the XID number into the Transaction Detail interface and pressing the `Go` button or by clicking on the XID in a Transaction Listing.

6.1. Transaction Summary

This section includes the same summarized information displayed about a transaction in the Transaction Listing [32].

6.2. Transaction Data

This area provides information about how a transaction was submitted and the status of the transaction.

- **Order Total** - The full amount that was attempted.
- **Authorized Amount** (Not Pictured) - This displays in the case of a partially authorized retail swiped/keyed transaction. This value is the amount that was actually charged to a customer.
- **Balance** - The available credit line amount of the gift card/debit/prepaid if passed back from the processor.
- **Entry Method** - This indicates whether a transaction was keyed or swiped.
- **Credit Card** - The last four digits of the credit card number. This is not listed for check or EFT transactions.
- **AVS Response** - This field is blank for check or EFT transactions. The value is the response received from the processing network to indicate whether the address and ZIP Code matched the credit card's account address on file in the bank's AVS database or not. Each processor has different AVS response values [33].
- **CVV Response** - This field is blank for check or EFT transactions. The value indicates whether the CVV/CID/CVV2 code matched or not. There are six CVV response values [36].
- **Batch** - This number indicates the batch that the transaction is a part of. This is not listed for check or EFT transactions.

- **Transaction Source** - This indicates how a transaction was submitted. The possible values are:
  - Cust - This indicates that a transaction was submitted via an HTML form post generated from an Internet order form.
  - HTML - This indicates that a transaction was submitted via a password with credentialed HTML form post generated from an Internet order form.
  - Phone - This indicates that a transaction was submitted via the Call-A-Charge [83] transaction by telephone service.
  - Session - This indicates that a transaction was submitted via any of the Control Panel interfaces.
  - Recur - This indicates that a transaction was generated through the automated recurring billing [146] system.
  - XML - This indicates that a transaction was submitted via an XML [90] request.
  - Upload - This indicates that a transaction was submitted via a file upload through the Blue Control Panel.
  - AV - This indicates that a transaction was a void generated by an auto-void setting [27].
- **Account Type** (Not Pictured) - This value is for EFT/Check transactions and identify the type of account the payment was drafted from.
- **Account Source** (Not Pictured) - This value is for NACHA transactions and identify whether the payment was drafted from a checking or savings account.
- **IP Address** - This is the IP address from which a transaction was submitted.
- **Error Message** - When a transaction fails, the response from the processor is listed in this field. This field is blank on a successful transaction.

6.3. Customer Information

The Customer Information section displays the information that was entered concerning the cardholder at the time of the transaction. This is the definition of each of those fields:

- **Name** - This displays the first and last name entered for the cardholder.
- **Address** - This displays the street address, city, state, ZIP Code, and country entered at the time the transaction was submitted. This information was submitted as the data to the AVS (address verification system) at the card processor.
- **Telephone** - This value is the phone number entered at the time of the transaction.
- **Email** - This is the email address which was submitted at the time of the transaction, and the address to which customer confirmation emails will be sent to (if the merchant has that feature enabled).
- **Cust ID** - The custom ID value that can be assigned by a merchant.

6.4. Shipping Address Information

This area will be blank unless the optional shipping information is submitted with the transaction attempt.

6.5. Recurring Information

When a transaction is set to be a recurring transaction, this area will display. To access the Recurring Details [44] tool, please click the Go button [37] to launch the window.

- **Start Date** - The day of when a transaction was set as a recurring transaction. This date may or may not coincide with the date of the original transaction.
- **Recipe Name** - This value is the name of the recurring recipe that is set to the transaction at that time.
- **Remaining Reps** - The number of repetitions left that the transaction will continue to rebill until it cycles to zero or is set to zero [150].
- **Recurring Total** - The amount that will be charged for future transactions.
- **On Hold** - This will display a red `Yes` or a green `No` indicating whether the transaction is on hold to prevent future billings or not. By clicking on the value, you can toggle between on hold (`Yes`) or no on hold (`No`).
- **Recurring History** - Clicking this `VIEW` button will pull open the history [152] of the recurring attempts on the specific transaction.

6.6. Order Items

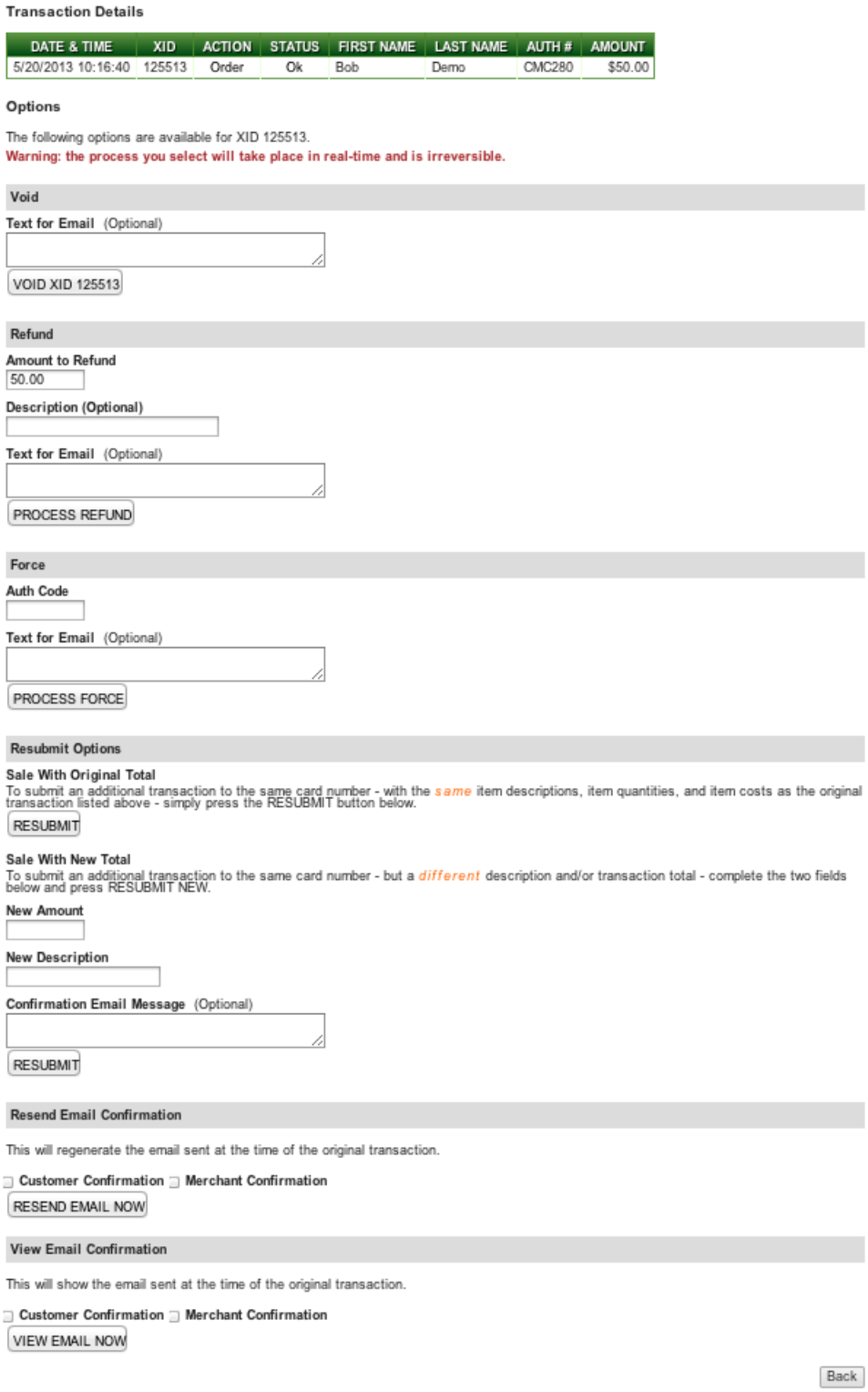
For each item submitted to the gateway system as a part of a total transaction, a separate Item Explanation will be displayed.

- **Cost** - This value is the price of each quantity of an item. This value is multiplied by the Qty value to populate the Tot field.
- **Desc** - This value is the description of the item submitted with the transaction.
- **Qty** - This value is the number of this item purchased. This value is multiplied by the Cost value to populate the Tot field.
- **Tot** - This value is the item total. The value is reached by multiplying the Cost by the Qty.

6.7. Transaction Options

Click the `Go` button for `Options` in Transaction Information section to launch this interface.

Figure 2.29. Transaction Options Interface



The features displayed above are the options available for a successful credit card sale transaction. In addition to those displayed in the illustration, there are several others, including items specific to check transactions. Here is what these features do:

- **Void** - This feature can be used to cancel sale, postauth, resubmit, or refund/credit transactions. Voids may only be issued prior to the batch settlement for the open batch that the original transaction is processed in. Batch settlement cycles daily. The `Text for Email` field can be used to add a message of up to 255 characters which will display on the emailed transaction confirmation. For all processors , a void will attempt to generate an

actual reversal of the authorization to release the “frozen" funds back to the cardholder’s account. If the reversal is unsuccessful, a void will be generated which will prevent the authorization from settling and allowing the authorization to “fall off” (which may take up to 10 days).

- **Refund** - This interface is used to generate a refund to a customer's account. By default, the amount in the interface will be the amount of the original transaction. You can modify the amount as needed. The `Text for Email` field can be used to add a message of up to 255 characters which will display on the emailed transaction confirmation. Additionally, a separate description value may be entered.

- **Force** - This feature should only be used if you are instructed to by the credit card processing network. When instructed to process a forced transaction, enter the approval code in the `Auth Code` field. The Text for Email field can be used to add a message of up to 255 characters which will display on the emailed transaction confirmation.

- **PostAuth** - Although not displayed in the illustration, this feature is used to complete a sale on a pre-authorized transaction.

- **Mark as Returned** - Although not displayed in the image, this feature is used to notate bounced or returned RediCheck or NACHA transactions. Marking a transaction as returned will not initiate any action with your bank. It will however set a recurring transaction to an on hold/held for failure state if your recipes have the *Hold On Failure* feature enabled. These return transactions will also show up in your reports so that you have a more accurate accounting of your check activity. This feature is not available for EFT transactions.

- **Resubmit Options** - There are two options to be used in different situations.

  - Resubmit Original Transaction - This will generate a new transaction using all of the information, including amount and description, processed during the original transaction.

  - Resubmit New Transaction - This will generate a new transaction using all of the billing information from the original transaction, but will process using the newly entered amount and description.

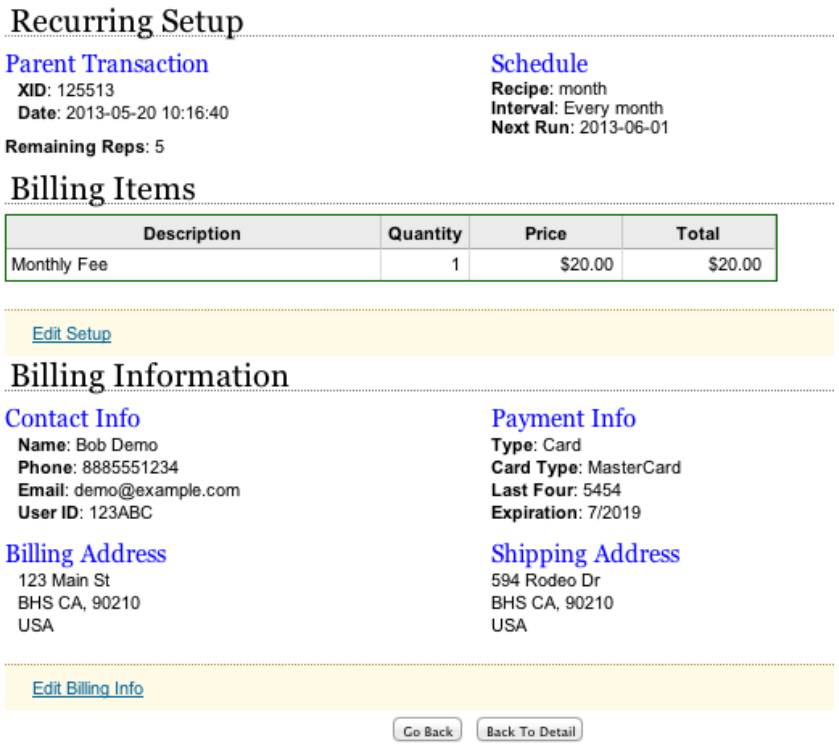
- **Resend Email Confirmation** - This reproduces a copy of the email(s) sent at the time of the transaction. This can send either the Customer Confirmation [66], the Merchant Confirmation [66], or both.

- **View Email Confirmation** - This displays reproductions of the emails sent at the time of the transaction.

6.8. Recurring Details

Clicking the `Go` button allows you to activate and edit recurring information for transactions.

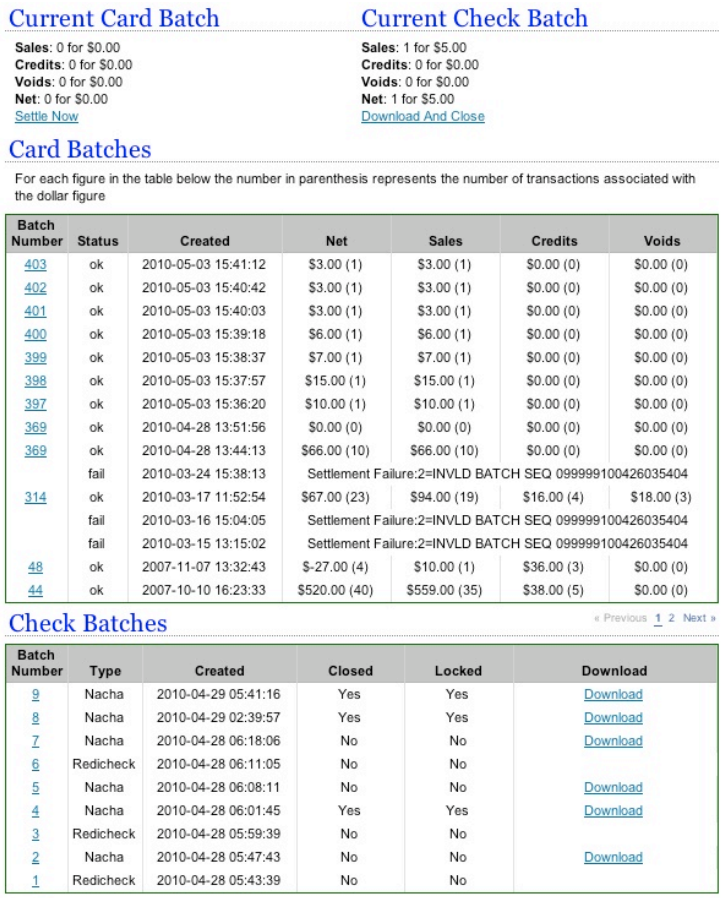
Figure 2.30. Recurring Details Window



Please read the in-depth instructions for the entire recurring system [60] in this section of the document.

7. Batches Window

Figure 2.31. Batch Tool



Current Card Batch

Sales: 0 for $0.00
Credits: 0 for $0.00
Voids: 0 for $0.00
Net: 0 for $0.00
Settle Now

Current Check Batch

Sales: 1 for $5.00
Credits: 0 for $0.00
Voids: 0 for $0.00
Net: 1 for $5.00
Download And Close

Card Batches

For each figure in the table below the number in parenthesis represents the number of transactions associated with the dollar figure

Batch Number	Status	Created	Net	Sales	Credits	Voids
403	ok	2010-05-03 15:41:12	$3.00 (1)	$3.00 (1)	$0.00 (0)	$0.00 (0)
402	ok	2010-05-03 15:40:42	$3.00 (1)	$3.00 (1)	$0.00 (0)	$0.00 (0)
401	ok	2010-05-03 15:40:03	$3.00 (1)	$3.00 (1)	$0.00 (0)	$0.00 (0)
400	ok	2010-05-03 15:39:18	$6.00 (1)	$6.00 (1)	$0.00 (0)	$0.00 (0)
399	ok	2010-05-03 15:38:37	$7.00 (1)	$7.00 (1)	$0.00 (0)	$0.00 (0)
398	ok	2010-05-03 15:37:57	$15.00 (1)	$15.00 (1)	$0.00 (0)	$0.00 (0)
397	ok	2010-05-03 15:36:20	$10.00 (1)	$10.00 (1)	$0.00 (0)	$0.00 (0)
369	ok	2010-04-28 13:51:56	$0.00 (0)	$0.00 (0)	$0.00 (0)	$0.00 (0)
369	ok	2010-04-28 13:44:13	$66.00 (10)	$66.00 (10)	$0.00 (0)	$0.00 (0)
	fail	2010-03-24 15:38:13	Settlement Failure:2=INVLD BATCH SEQ 099999100426035404			
314	ok	2010-03-17 11:52:54	$67.00 (23)	$94.00 (19)	$16.00 (4)	$18.00 (3)
	fail	2010-03-16 15:04:05	Settlement Failure:2=INVLD BATCH SEQ 099999100426035404			
	fail	2010-03-15 13:15:02	Settlement Failure:2=INVLD BATCH SEQ 099999100426035404			
48	ok	2007-11-07 13:32:43	$-27.00 (4)	$10.00 (1)	$36.00 (3)	$0.00 (0)
44	ok	2007-10-10 16:23:33	$520.00 (40)	$559.00 (35)	$38.00 (5)	$0.00 (0)

Check Batches

« Previous 1 2 Next »

Batch Number	Type	Created	Closed	Locked	Download
9	Nacha	2010-04-29 05:41:16	Yes	Yes	Download
8	Nacha	2010-04-29 02:39:57	Yes	Yes	Download
7	Nacha	2010-04-28 06:18:06	No	No	Download
6	Redicheck	2010-04-28 06:11:05	No	No	
5	Nacha	2010-04-28 06:08:11	No	No	Download
4	Nacha	2010-04-28 06:01:45	Yes	Yes	Download
3	Redicheck	2010-04-28 05:59:39	No	No	
2	Nacha	2010-04-28 05:47:43	No	No	Download
1	Redicheck	2010-04-28 05:43:39	No	No	

Use this tool to view your batch settlement history and the data from the current open batch, download NACHA files, and generate manual settlements. EFT data does not list in this interface. This only displays information for Credit Card, NACHA, and RediCheck transactions.

For credit cards, funds do not deposit to you until after a settlement takes place. By default, your gateway account will automatically settle each day and this interface shows the history of those settlement attempts. And when needed, you can run manual settlements very simply. You can click on the batch number to display a listing of card transactions that make up that batch.

For NACHA transactions, you can download the files of transaction information that have been formatted using your bank's NACHA identifiers. NACHA transactions will not deposit to you until your have taken the downloaded file and provide it to your bank following their file submission procedures. You can click on the batch number to display a listing of transactions that make up that batch.

For RediCheck transactions, this interface lists all groups of checks that have processed. You can click on the batch number to display a listing of transactions that make up that batch.

7.1. Card Batch Settle Now Tool

This allows a merchant to run settlement at any time without the assistance of iTransact's service technicians. When you click the `Settle Now` link, the account will begin the settlement request for the account. Batch settlements may take a few minutes depending on the number of transactions in the open batch and network traffic. When the batch has settled it will either display a successful response or a failed response. A failed response will indicate the cause of the error.

Figure 2.32. Successful Response Example

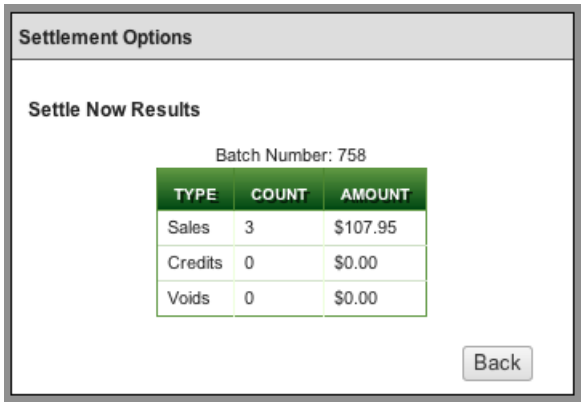
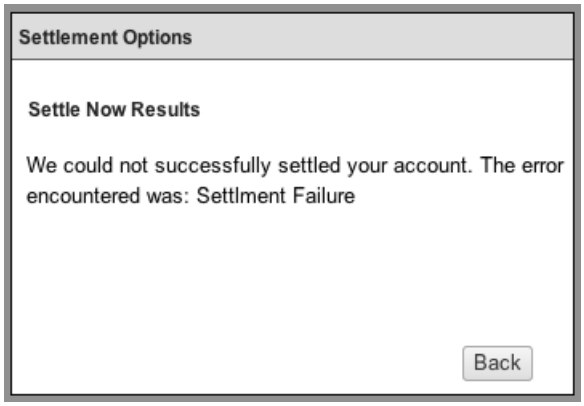


Figure 2.33. Failed Response Example



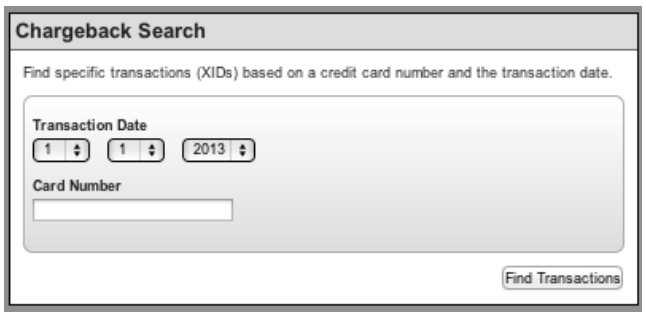
This tool can be used by merchants who have any settlement setting (`Auto`, `Manual`, or Specified time) in the Account Settings. Remember, if an account is set to `Manual` settlement, it will not batch until you use the `Settle Now` tool.

7.2. NACHA Download Tool

If you are authorized by your bank and iTransact to use the NACHA payment system, your NACHA formatted files will be available to download in this interface. The interface creates a plain text (.txt) file with your NACHA data. When you download your NACHA file, it will lock the batch. Until the batch is locked, you can void check transactions using the Transaction Options [42] tool.

8. Chargeback Interface

Figure 2.34. Chargeback Interface

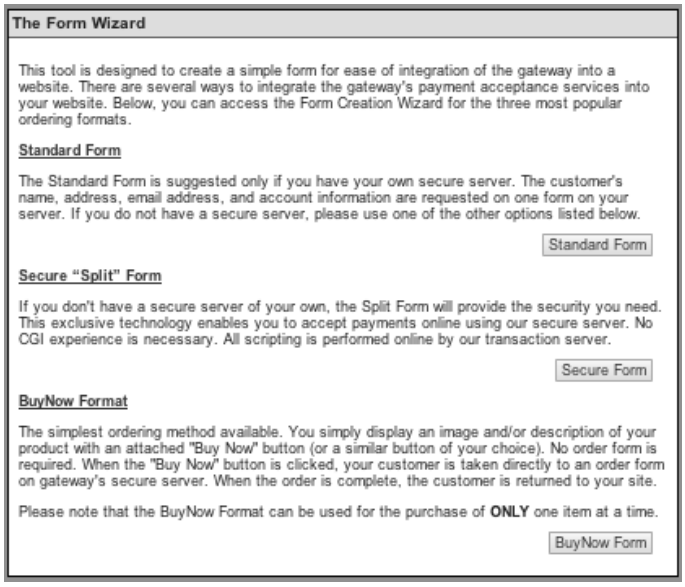


Accepting credit cards is one of the simplest methods for taking payments from customers. Occasionally, a credit card charge is disputed. The dispute and research process is called a chargeback. When a charge is disputed, a letter is sent to the merchant by the card processing bank which lists only the card number and the date of the transaction. If this happens to you, you can use this interface to find the XID for the transaction so that you can pull the contact information for the customer. This will be helpful to allow you to answer the dispute. Set the approximate date and enter the card number and matching XID(s) will list. Click on the XID to get to the details of the transaction.

Please deal with all chargebacks quickly.

9. Form Wizard

Figure 2.35. Form Wizard



The Form Wizard allows you to create forms quickly and easily for use with the Standard Form [123], Split Form [129], and BuyNow [135] formats. This simple interface walks you through several questions and then builds an HTML form. View and copy the source of that page and then upload directly into your web server. This tool can also be used to generate a simple HTML form skeleton which can be modified by the merchant to add advanced features, formatting, and/or functionality.

9.1. Generating Standard Forms

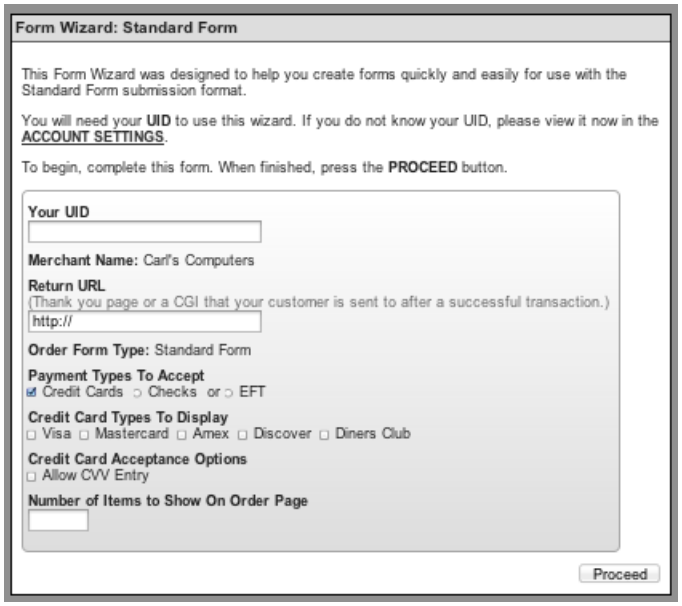
Standard Forms should only be used by merchants who have their own secure servers and meet CISP/PCI-DSS standards. This format collects all order and payment information on the merchant’s own secure server and then posts that data to the gateway's secure transaction server. Building a simple Standard Form using the Form Wizard is an easy process. By filling out the fields in the wizard, a form will be generated that is integrated with the gateway. The form can be additionally modified to look and feel like the rest of your site. Follow these instructions to generate a form:

1. Log into the Control Panel.

2. Access the Form Wizard.

3. Chose the Standard Form option. This generation tool window will open.

Figure 2.36. Form Wizard - Standard Form



4. Enter the UID [23] listed in your [23]Account Settings [23].

5. Enter the correct address for the Return URL [123]. This will be the address of the page or script that a customer will be directed to after a successful transaction. This becomes the value of the ret_addr [123] field of your form.

6. Select the appropriate check marks for the payment types to display on the payment page. Only select a payment type if you are approved to accept that method.

7. Select the check box for `Allow CVV Entry` if you'd like to ask your customer's for the cvv value. This is not required.

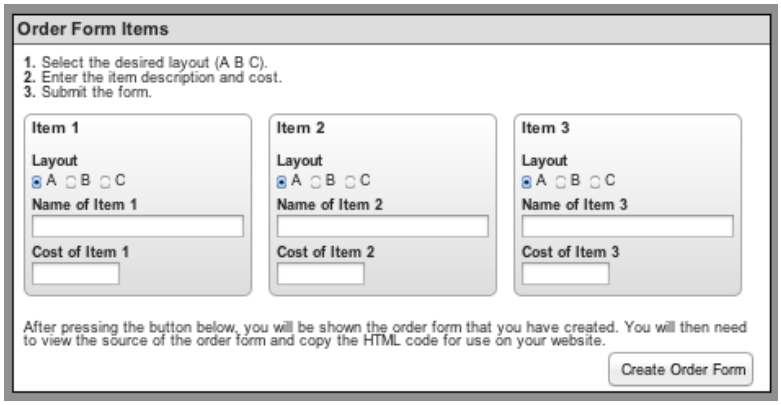
8. Enter the number of items that will display on the order form page and click `Proceed`. (For this example, the value for this field is "3")

9. A dialogue box will open displaying instructions for the rest of the Form Wizard.

Figure 2.37. Form Wizard Instructions

10. Immediately, a second window, the Order Form Items page, will open.

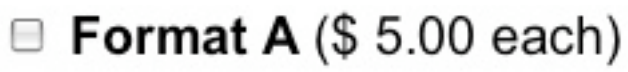
Figure 2.38. Wizard Order Form Items



11.The `A B C` layout options determine the way an item will display on an order form page. Choose the format the best fits your needs. Use the same format for each item, or any combination of formats on any form.

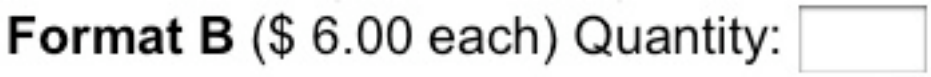
- **Format A** - This style allows for the purchase of a single quantity of one item, by clicking a checkbox and completing the billing information.

Figure 2.39. Standard - Item Format A



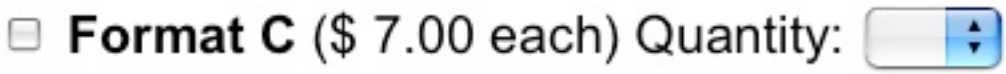
- **Format B** - This style allows for the purchase of multiple quantities of an item by entering the quantity value into a text box.

Figure 2.40. Standard - Item Format B



- **Format C** - This style allows a customer to select the item by clicking the checkbox and choosing the quantity from a drop-down menu.

Figure 2.41. Standard - Item Format C



12.After selecting the display format, enter the name of the item in the appropriate item fields (in the examples above, the item names are "Format A", "Format B", and "Format C").

13.Enter the cost of each item (entered without "$") and click the `Create Order Form` button.

14.The order form will then display.

Figure 2.42. Standard Form - Example

15.Right-click with the mouse on the form page and view the source of the page. Copy and save that HTML code, edit/modify it as needed, and upload the page into your website's secure server.

9.2. Generating Split Forms

The Split Form is used by merchants who do not have their own secure servers. This enables a customer to enter non-secure information on your server and enter the secure billing information on the gateway's secure server. Building a simple Split Form using the Form Wizard is an easy process. By filling out the fields in the wizard, it will generate a form that is ready to go or a form that can be additionally modified to look and feel like the rest of your site. Follow these instructions to generate a form:

1. Log into the Control Panel.

2. Access the Form Wizard.

3. Chose the secure Split Form option. This generation tool window will open.

Figure 2.43. Form Wizard - Split Form

4. Enter the UID [23] listed in your Account Settings. [23]

5. Enter the correct address for the Return URL [123] .This will be the address of the page or script that a customer will be directed to after a successful transaction. This becomes the value of the ret_addr [123] field of your form.

6. Select the appropriate check marks for optional credit card acceptance features. None of these features are required.

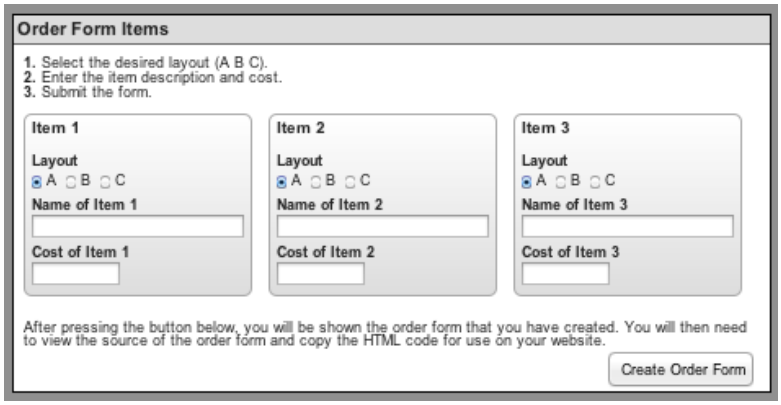
7. Enter the number of items that will display on the order form page and click `Proceed`. (For this illustrated example, the value for this field is "3")

8. A dialogue box will open displaying instructions for the rest of the Form Wizard.

Figure 2.44. Form Wizard Instructions

9. Immediately, a second window, the Order Form Items page, will open.

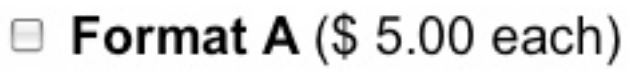
Figure 2.45. Wizard Order Form Items



10.The `A B C` layout options determine the way an item will display on an order form page. Choose the format the best fits your needs. Use the same format for each item, or any combination of formats on any form.

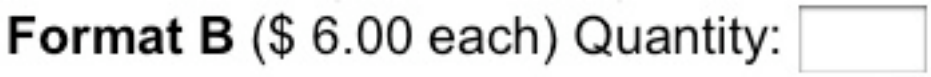
- **Format A** - This style allows for the purchase of a single quantity of one item, by clicking a checkbox and completing the billing information.

Figure 2.46. Split - Item Format A



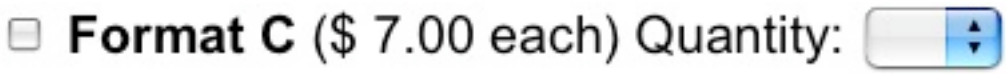
- **Format B** - This style allows for the purchase of multiple quantities of an item by entering the quantity value into a text box.

Figure 2.47. Split - Item Format B



- **Format C** - This style allows a customer to select the item by clicking the checkbox and choosing the quantity from a drop-down menu.

Figure 2.48. Split - Item Format C

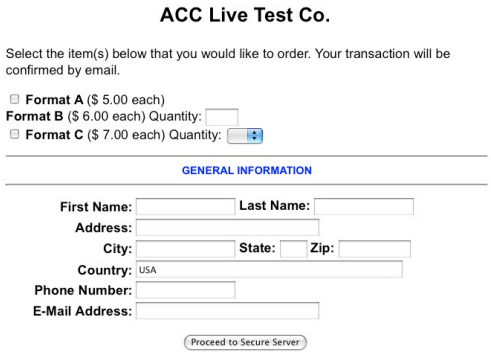


11.After selecting the display format, enter the name of the item in the appropriate item fields (in the examples above, the item names are "Format A", "Format B", and "Format C").

12.Enter the cost of each item (entered without "$") and click the `Create Order Form` button.

13.The order form will then be created and displayed. This form will be the half of the form that resides on the your web server. It does not display the billing information request fields on the first half of the Split form. The secure half of the order form (where payment information is entered) resides on the gateway's secure server. The HTML coding that is included in the order form page will generate merchant specific information on the secure half of the Split form.

Figure 2.49. Split Form - Example



14. Right-click with the mouse on the form page and view the source of the page. Copy and save that HTML code, edit/modify it as needed, and upload the page into your website's server.

NOTE: *The secure half of the Split Form has optional form tag options and style settings that will determine how the secure page display.*

9.3. Generating BuyNow Buttons

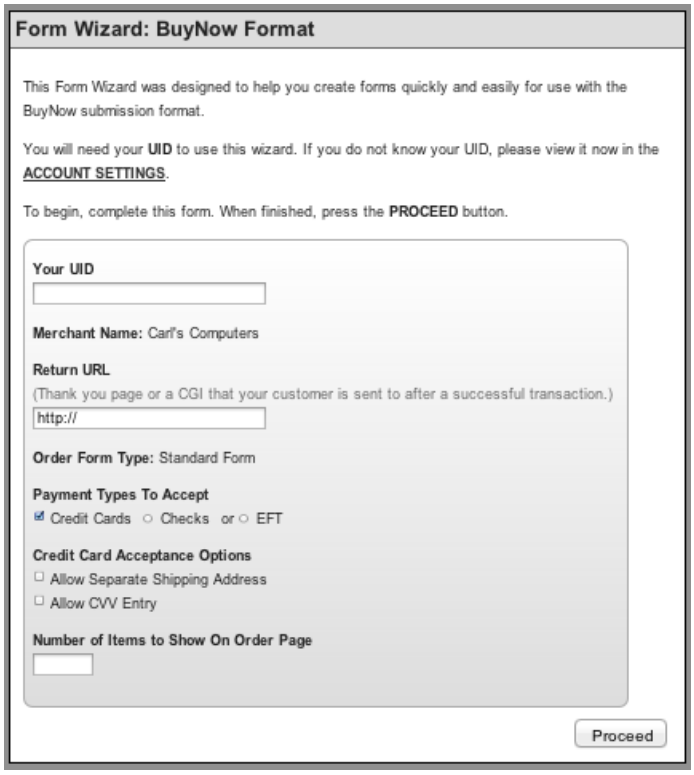
BuyNow buttons are used by merchants who do not have their own secure servers. Your customers choose an item on your web site and enter their secure billing information on your dynamic page on gateway's secure server. Building a BuyNow button using the Form Wizard is an easy process. By filling out the fields in the wizard, it will generate a button, or an entire form, that can be added to your site. Follow these instructions to generate a form:

1. Log into the Control Panel.

2. Access the Form Wizard.

3. Chose the BuyNow Format option. A new window will open.

Figure 2.50. Form Wizard - BuyNow Form

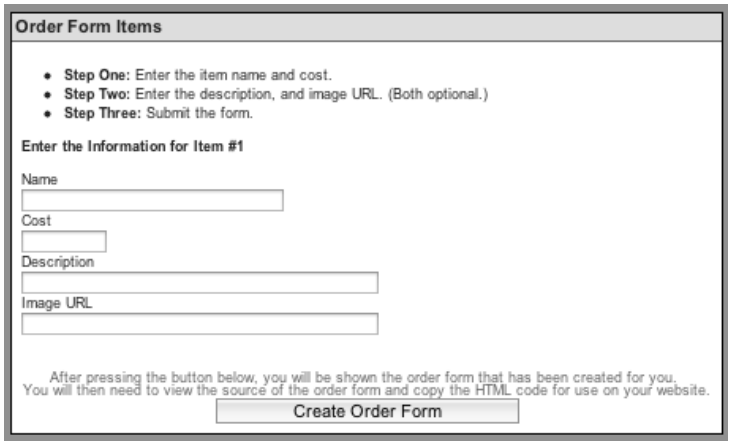


4. Enter the UID [23] listed in your [23]Account Settings [23].
5. Enter the correct address for the Return URL [123]. This will be the address of the page or script that a customer will be directed to after a successful transaction. This becomes the value of the ret_addr [123] field of your form.
6. Select the appropriate check marks for the payment types to display on the payment page. Only select a payment type if you are approved to accept that method.
7. Select the appropriate check marks for optional credit card acceptance features. None of these features are required.
8. Enter the number of items that will need separate buttons and click `Proceed`. (For this illustrated example, the value for this field is "1")
9. A dialogue box will open displaying instructions for the rest of the Form Wizard.

Figure 2.51. Form Wizard Instructions

10. Immediately, a second window, the Order Form Items page, will open.

Figure 2.52. Wizard Order Form Items



11.In the `Name` field, enter the item. (In this example, that value is "ITEM 1")

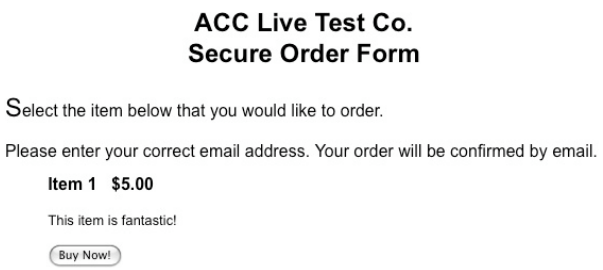
12.In the `Cost` field, enter the cost of the item without "$" sign. (In this example, the cost is five dollars entered as "5.00")

13.In the `Description` field, enter a message about the item. (In this example, the description is "This item is fantastic!").

14.In the optional `Image URL` field, include an absolute address of an image file that you would like to display next to the button (i.e. http://www.domain.com/imagefile.jpg) and then click the `Create Order Form` button.

15.The BuyNow button form page will then display.

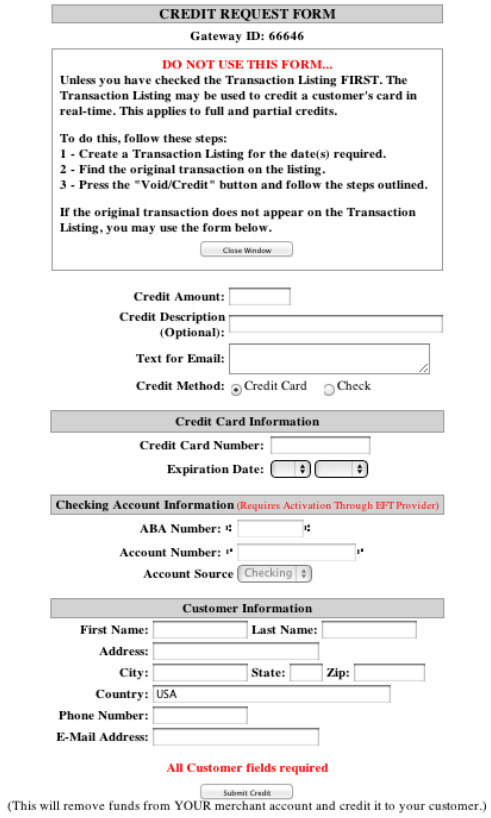
Figure 2.53. BuyNow Button - Example



16.Right-click with the mouse on the form page and view the source of the page. Copy and save that source, edit/modify it as needed, and upload the page into your website's server in it's entirety, or cut the button commands and paste it into a separate page.

10. Post a Credit

Figure 2.54. Post a Credit Interface



In most cases, this interface is not available in the Control Panel. Contact the *support team* if you need to use this feature. When accessible, this interface is used to generate a refund or payment to a cardholder's account that was not originally charged through the gateway account (with the prior permission from the processor). This will remove monies from your account and deposit to the account entered in this interface. This feature should only be used in that case. If a transaction has processed through the gateway, please use the Void or Credit/Refund functions in the Options [42] interface of the Transaction Listing. Depending on the transaction types a merchant is authorized to accept, this interface will display differently.

10.1. Instructions

Entering a credit transaction requires entering all of the correct information in the Post a Credit interface.

- **Credit Amount** - This amount that will be withdrawn from the merchant's account and will deposit into the customer's account. Enter the amount in the XXXX.xx format (no dollar signs or commas).
- **Credit Description** - This is an optional field which can be used to describe a refund.
- **Text for Email** - The value of this field will be included in an email to the customer. This message can be up to 255 characters.
- **Credit Method Button** - Use this to select whether the credit will be issued to a customer's credit card or checking account.
- **Credit Card Information Section** - This section is to be used if you need to generate a payment to the customer's credit card account. This can only be used if you are setup to process credit card transactions.
  - Credit Card Number - The value of this field is the credit card account which will receive the credit payment. This should be entered with no spaces or dashes.

  - Expiration Date - These drop down menus should be set to the expiration date (month and year) which is listed on the cardholder's account.

- **Checking Account Information Section** - Complete this section if you need to generate a payment to the customer's checking account. This can only be used if a merchant is setup to process EFT transactions.
  - ABA Number - This is the nine digit ABA Routing number for a customer's bank. These are generally the first nine numbers listed in the line of numbers across the bottom of a check.
  - Account Number - This is the customer's checking account number as it appears on a check.
- **Customer Information Section** - This section is always required for any type of credit transaction processed through this interface.
  - First Name - This should be the account holder's first name.
  - Last Name - This should be the account holder's last name.
  - Address - This should be the account holder's street address as listed with the account issuer.
  - City - This should be the account holder's city as listed with the account issuer.
  - State - This should be the state abbreviation of the account holder as listed with the account issuer.
  - ZIP - This should be the account holder's postal code as listed with the account issuer.
  - Country - This should be the account holder's country as listed with the account issuer.
  - Phone Number - This should be a contact phone number for the customer.
  - Email Address - This should be the customer's email address. The transaction confirmation email will be sent to this address.

Once the appropriate fields are completed, the merchant needs to click the `Submit Credit` button, and the credit transaction will be submitted. Emails will be generated to the merchant and the customer (if you have the Credit email delivery selected), and the transaction information will listed in the Transaction Listing [32].

11. Auction Payments

Figure 2.55. Auction Payment Interface

For merchants who sell items at online auctions sites, one of the biggest frustrations is receiving payment from the winning bidder in a prompt manner. Our gateway provides a simple solution for the problem. With very little effort, a merchant can obtain instant check or credit card payments from bid winners using the Auction Payments secure transaction services - even if they don't have their own website. By filling out this simple form, the buyer receives an e-Invoice email [79] with a link to the merchant's secure Auction Payments page (which is hosted on the gateway's secure server) where payment can be made.

11.1. Submission Form Details

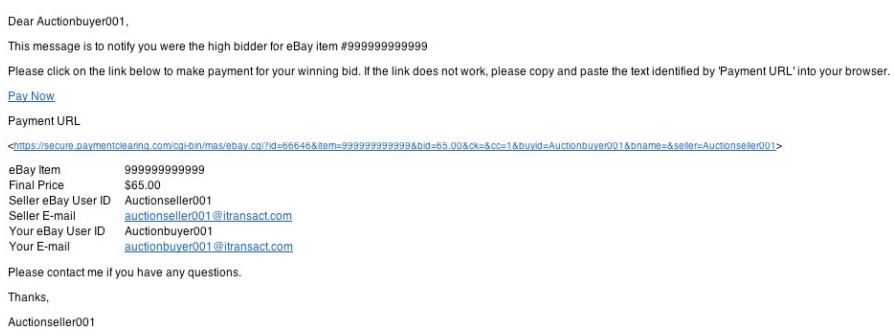
To use the *Auction Payments* tool, please complete the request form and an email with a secure payment link will be sent. There are the definitions of these fields:

- **Your Gateway UID** - Enter the UID [23] listed in your [23]Account Settings [23].
- **eBay Seller ID** - The value for this field should be your eBay registered seller id.
- **Seller E-Mail Address** - This should be the email address to which a buyer can contact you.
- **Buyer ID** - The value for this field should be the winning bidder's registered eBay buyer id.
- **Buyer E-Mail Address** - This address will receive an email with the link to your secure Auction Payments page (Which is hosted on the gateway's secure server) where payment can be made.
- **eBay Item #** - This should be the item ID assigned by eBay at the time of the posting of the auction.
- **Bid Amount** - This value should be the amount of the winning bid (entered without '$ or commas - should be XXXX.xx).
- **Type of Payments to Accept** - These checkboxes will indicate to the gateway which payment fields will display when a customer access the Auction Payments page.
- **Submit AuctionPay Request Button** - This button needs to be click once the fields are filled out correctly.
- **Close Window Button** - This is to be used to cancel an *Auction PayMe* request prior to clicking the `Submit AuctionPay Request` button.

11.2. Auction Payments e-Invoice

When you complete the submission form and an Auction Payment request is submitted, an e-Invoice email is sent to the winning bidder with the link that they can follow to complete payment.

Figure 2.56. Auction Payments e-Invoice



Dear Auctionbuyer001,

This message is to notify you were the high bidder for eBay item #999999999999

Please click on the link below to make payment for your winning bid. If the link does not work, please copy and paste the text identified by 'Payment URL' into your browser.

Pay Now

Payment URL

<https://secure.paymentclearing.com/cgi-bin/mas/ebay.cgi?id=66646&item=999999999999&bid=65.00&ck=&cc=1&buyid=Auctionbuyer001&bname=&seller=Auctionseller001>

eBay Item	999999999999
Final Price	$65.00
Seller eBay User ID	Auctionseller001
Seller E-mail	auctionseller001@itransact.com
Your eBay User ID	Auctionbuyer001
Your E-mail	auctionbuyer001@itransact.com

Please contact me if you have any questions.

Thanks,

Auctionseller001

11.3. Auction Payments Form

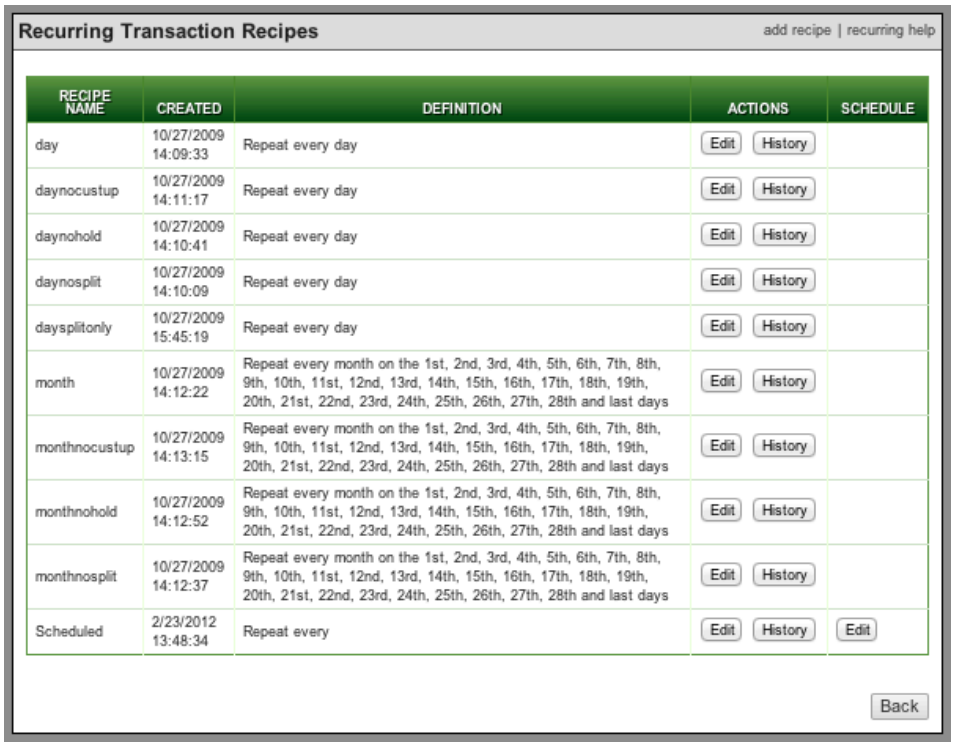
Figure 2.57. Auction Payment Page

The winning bidder will need to fill out this form and enter their account information to complete their payment. The bidder will be shown a success or failure page. If the transaction is approved, you and the bidder will receive an email confirmation.

NOTE: *The gateway does not track unpaid e-Invoices. If you need to send an e-Invoice a second time, you will have to fill out the submission form a second time.*

12. Recurring Transactions

Figure 2.58. Recurring Recipe Window



The iTransact gateway offers one of the most robust recurring billing features available on the Internet. This is an ideal tool for merchants who bills for subscription, payment plans, or according to a set schedule. This recurring

billing tool automates ongoing billings in a simple manner which gives you as much control as if you were manually entering each transaction - without the hassle of manually entering the transactions. This is available for credit card and check transactions. This window displays your recipes. The window can be accessed from the `Recurring Transactions` link in the Control Panel or from the `List Recipes` link in the Recipe Builder [61].

12.1. Understanding the System

There are some basic ideas that you need to understand when using our recurring billing system. Please keep these items in mind:

- A Recurring Recipe is the schedule which contains the instructions as to when a recurring transaction is billed. The Recurring Repetitions/Remaining Repetitions is the number of times that a transaction follows the recipe. Once a transaction is set as a recurring transaction, it will continue to follow the recipe until the number of repetitions cycles down to or is manually set to zero. The transaction on-hold setting will prevent future billings without changing the number of Recurring Repetitions.

- Separate recipes do not need to be built for transactions following the same schedule, even if the transactions are initiated at different times, have different amounts, and different necessary repetitions. There is no limit to the number of transactions that can use the same recipe.

- There is no limit to the number of recipes that you can build.

- The recurring cycle begins each night at 12 Midnight, Mountain time. Any necessary modifications to a recurring transaction, or its recipe, must be completed prior to 11:59 PM for it to be reflected as a part of the next day's recurring cycle. For instance, if it is January 31st and a recurring transaction is scheduled to process on February 1st, that transaction can be modified up to 11:59 PM on January 31st. In further explanation, if a merchant needed to set the remaining repetitions to zero to prevent future transactions, but missed the 11:59 PM deadline, the merchant would have access to change the number of remaining repetitions to zero. However, since the cycle had already begun, the transaction would still be billed. Future transactions would be prevented, but a refund or void would now be necessary because the transaction which the merchant had intended to stop was billed. The remaining repetitions in such a case would display as “-1”.

- When a transaction is initially submitted for processing, recurring details may be passed as part of the form that will automatically create future recurring charges, based on the details that you provide. In addition, you may also modify previously submitted transactions and mark them as recurring. This is done via the Transaction Details window.

- If you do not need to initiate a credit card charge at the time of entry (and your processor supports this type of transaction) use a zero amount AVSOnly transaction type. If your processor does not support AVSOnly transactions, you can use a pre-auth for a minimal amount.

- The calculations used to determine when a transaction will recur are based on the initial transaction date and when the transaction was set to recur. Please contact the support team for clarification.

- The largest allowed value for the Recurring Repetitions is “99999”. There is no "unlimited" setting, but using the "99999" value will allow for many years of recurring attempts.

- Recurring transaction details can be updated by you or by the card holders.

12.2. How to Add a Recipe

The schedule that a recurring transaction follows in the iTransact gateway is called a Recurring Recipe. Just like a baking recipe, there are specific items required in the recipe to give you the outcome that you desire.

To access the recipe builder tool, please click the `Add Recipe` link in the Recurring Transactions window. Complete the interface that opens and click the `Create Recipe` button and the recipe will be added to your list of available recipes.

Figure 2.59. Recurring Recipe Builder

Here are definitions of the items in this interface:

- **Recipe Name** - When naming a recipe, please remember that it will be case-sensitive (must be lowercase) and can be only one word. Any alpha-numeric characters can be used. You should make it easy to remember. For instance, you may want to name a recipe “1stofmonth” if it's designed to bill on the first day of the month.

- **Recipe Types** - Select a type of recipe. They each function differently.

  - Scheduled Recipes - Don’t be confused by this name. All recipe types allow you to set a schedule. Using a Scheduled Recipe allows you to run all transactions linked to a recipe at a date that can be controlled and scheduled manually using the scheduling tool. The scheduling tool may be accessed from your Recipe List after a Scheduled Recipe is built. The scheduling tool is only available for scheduled recipes. The Delay Period can be used to prevent a transaction from recurring too soon after the initial transaction is processed. The Delay Period *is* the number of days after the original transaction before it is eligible for a scheduled recurrence. To build a Scheduled Recipe, choose a recipe name, click the radio button to the left of `Scheduled`, enter a numeric value for the number of days in the `Delay Period`, add any additional features, and click the `Create Recipe` button.

  - Day Recipes - This type of recipe allows you to bill transactions applied to a recipe to bill every *X* number of days after the initial billing (and from billing to billing). To build a Day Recipe, choose a recipe name, click the radio button to the left of `Day`, enter the value for the number of days between recurring attempts, add any additional features, and click the `Create Recipe` button.

- Week Recipes - Building a Week Recipe allows you bill a transaction on specific days of the week - even multiple days during the same week. You can select `1`, `2`, or `3` weeks between billings and can check any day or (days) for the billings to take place. To build a Week Recipe, choose a recipe name, click the radio button to the left of `Week`, select the value for the number of weeks between recurrings, select the day (or days) of the week on which the billings will take place, add any additional features, and click the `Create Recipe` button.

- Month Recipes - The type of recipe allows you to bill transactions every X number of months on the “Nth day” (or days) of that month. Since some months have only 28, 30, or 31 days in the month, days 29-31 are covered under the `Last Day` selection. This type of recipe assumes that the recurring will begin in the calendar month after the initial transaction is processed. This means if, for instance, a transaction is billed on January 5th, and the recipe instructions are built to bill every 1 month on the 15th day of the month, the transaction would experience its first recurring billing on February 15th (not on January 15th). To build a Month Recipe, choose a recipe name, click the radio button to the left of `Month`, select the value for the number of months between recurrings, select the day (or days) of the month on which the billings will take place, add any additional features, and click the `Create Recipe` button.

- **Split Billing** - Don't be confused by the name of this function. This feature can be used with any of the type of recipe and any of our connection types. This feature allows the recurring transactions to be billed a different amount than the initial transaction. The amount can be changed automatically when setting up a form based recurring transaction using a recipe built with the Split Amounts function, or the amount can be changed manually in the Edit Recurring Items [150]. If there is ever the potential that the amount of a billing may increase, it is wise to set all recipes to allow Split Billing.

- **Hold On Failure** - This feature is available for credit card transactions only. If this is enabled on a recipe and a recurring attempt fails, that transaction will be put “on hold” automatically to prevent future billing attempts so that account information can be updated (either by the merchant or the customer). If this is enabled, the Recurring Postback [153] will include the “on hold” parameter to notify if this has been triggered due to a failure.

- **Allow Customer Update** - This feature is available for credit card transactions only. If enabled for a recipe, confirmation emails for successful [76] and failed [78] recurring transactions will include a link to a secure billing update page [87]. A cardholder will be able to follow the simple instructions to change their credit card billing information through a secure billing interface. If a recurring transaction failure triggered the necessary update, a new transaction will be attempted at the time of the update to bring the account payments up to date for the failed billing. If enabled, the Recurring Postback [153] will include the *billing_update_token* value.

- **Email Text** - This allows a merchant to pass a generic message in the text of each of the confirmation emails sent out when a transaction using the recipe recurs.

12.3. Recurring Help

This link opens the Recurring Help page. That page offers a quick reference guide when creating new recipes or setting transactions as recurring.

12.4. Further Details

The instructions for implementation and integration of automated recurring is available here [146].

13. Card Setup

Figure 2.60. Card Setup Interface

CARD SETUP
Select your Credit Card Processing Network from the list below.

PROCESSING NETWORK NAME	REQUIRED INFORMATION
FIRST DATA - CARDNET (CES/First Data/FDMS)	Cardnet Merchant Number, Datawire ID* (20 Digits), Cardnet Termid (Terminal ID)
FIRST DATA - NABANCO (FDMS SOUTH, First Data - Canada)	Merchant Number (11 digits), Datawire ID* (20 Digits), Terminal ID (Optional), "SE" Numbers (If accepting Amex, Discover, Diner's)
FIRST DATA - OMAHA (FDR7/ETC7/FDMS Type 7)	Merchant Number (12, 15 or 16 digits), Datawire ID* (20 Digits), Device ID (Optional)
PAYMENTECH (Gensar)	Client Number (ISO # or Bank #), Merchant Number, Terminal Number **(Must be terminal-based.)**
NDC - Atlanta East Platform (GLOBALPAY/GPS/CNET)	Bank ID, Terminal ID, Settlement Type **(Must be terminal-based.)**
Elavon (formerly NOVA)	BIN Number, Terminal ID **Elavon Setup Information:** Settlement Type: MANUAL (not Auto) VAR: PaymentClearing
TSYS (Visanet/VITAL)	Merchant Name, City, State, Zip, Customer Service Phone, Time Zone, Merchant Number (12 digits), Bank ID Number / BIN (6 digits), Terminal Number (4 digits), Store Number (4 digits), Agent Number (6 digits), Chain Number (6 digits), Merchant Category / SIC Code (4 digits)
Exchange (HEARTLAND/EXCHANGE)	Merchant Name, City, State, Zip, Customer Service Phone, Time Zone, Merchant Number (12 digits), Bank ID Number / BIN (6 digits), Terminal Number (4 digits), Store Number (4 digits), Agent Number (6 digits), Chain Number (6 digits), Merchant Category / SIC Code (4 digits)

**** Datawire ID is optional. However, it is highly suggested since this will allow your transactions to be processed much faster. You can obtain a Datawire ID, generally free of charge, from your processor.***

Close Window

Until this interface has been completed, you will not be able to accept credit cards. In most cases, your sales agent will have this completed for you. The interface can only be used once. If you need to modify the merchant account information anytime after the submission through this interface, the information can be submitted through the *support system*. Each processing network listed in the interface requires different processing information [2]. Click on the name of the processing network to open the processor specific submission form and fill out the required fields.

13.1. General Information

Each of the processor submission form requires the following information:

- **Merchant Account Contact Information** - The contact information for your merchant account service provider should be listed here.
- **Address Verification** - All of the major credit card processors will accept transactions that do not pass AVS. The fact that processors do not reject non-AVS transactions is a great concern of ours. Because of this, we've introduced one of the first AVS Auto-Void systems for the Internet. Our system allows the software to void transactions that are allowed through the processor without passing AVS based upon the requirement level set by the merchant. Remember, the gateway does not provide the AVS Responses. Those responses are generated by the credit card issuing bank and reported by the processing network based on information located in the bank's AVS database (which may or may not match the bank's statement database). However, the gateway system will perform the auto-void according to the requirements you choose. You can modify these settings at any time. Keep in mind, a(n) void/auto-void of an authorized transaction cancels the charge, but does not cancel an authorization. An authorization freezes funds in an account, so that a completed charge can withdraw those frozen funds. A voided authorization may "freeze" the funds in the customer's account for up to 10 days. The following auto-void settings are available:
  - No AVS Auto-Void - This will allow any approved transaction to process regardless of the address verification response.
  - Void Unless ZIP Matches - This will void any approved transaction for which the processor indicates that the ZIP Code entered does not match the ZIP Code listed in the bank's AVS database (even if the street address matches).
  - Void Unless Addr Matches - This will void any approved transaction for which the processor indicates that the street address entered does not match the street address listed in the bank's AVS database (even if the ZIP Code matches).

  - Void Unless Both Match - This setting requires that both the address and the ZIP Code match exactly what the issuing bank's AVS database has on file for the customer. If either the address or ZIP Code, or both, come back as a non-match, that approved transaction will be voided.

- **CVV Verification** - The CVV code is a security feature for "card not present" transactions (e.g., Internet transactions), and now appears on most (but not all) major credit and debit cards. This feature is a three or four digit code which provides a cryptographic check of the information embossed on the card. Therefore, the CVV code is not part of the card number itself. This setting allows you to have sale transactions automatically voided if the processing network approves the transaction but indicates that the CVV entered does not match the CVV database at the customer's credit card issuing bank. Most issuing banks do not require a CVV number to be entered for a transaction to process. However, a small group of banks do require correct CVV entry for Internet based transactions. The gateway system will perform the auto-void according to the requirements you choose. Modify these settings at any time. Keep in mind, a(n) void/auto-void of an authorized transaction cancels the charge, but does not cancel an authorization. An authorization freezes funds in an account, so that a completed charge can withdraw those frozen funds. A voided authorization may “freeze” the funds in the customer's account for up to 10 days. The following levels are available:

  - No Auto-Void Required - This will allow any approved transaction to process regardless of the CVV verification response.

  - Void Unless Full CVV Matches - This setting will void any authorized transaction which is returned with a non-matching or empty CVV response.

  - Void If CVV Not Entered - With this setting a customer's transaction will be voided if the bank indicates that a CVV code should exist on the card, but was not entered.

  NOTE: *The iTransact gateway does not perform CVV auto-voids for American Express transactions.*

- **Authorized Card Types** - Set these buttons to `Yes` or `No` depending on your status with each of the card issuers. If you are not authorized by the issuer to accept a card type, the merchant should leave the `No` button selected.

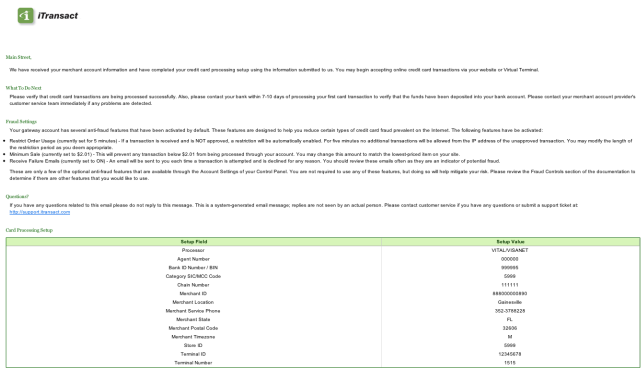
- **Other Information** - Any message may be included here to pass information to the gateway support team concerning your account.

14. Emails from the Gateway

The gateway system uses emails as the primary source of communication. Many different types of emails are generated for different purposes. Settings for some of the merchant [22] and customer [24] emails are available in the Account Settings [19].

14.1. CC Processing Account Activation

Figure 2.61. CC Processing Account Activation Email



iTransact

Main Street,

We have received your merchant account information and have completed your credit card processing setup using the information submitted to us. You may begin accepting online credit card transactions via your website or Virtual Terminal.

What To Do Next

Please verify that credit card transactions are being processed successfully. Also, please contact your bank within 7-10 days of processing your first card transaction to verify that the funds have been deposited into your bank account. Please contact your merchant account provider's customer service team immediately if any problems are detected.

Fraud Settings

Your gateway account has several anti-fraud features that have been activated by default. These features are designed to help you reduce certain types of credit card fraud prevalent on the Internet. The following features have be activated:

- Restrict Order Usage (currently set for 5 minutes) - If a transaction is received and is NOT approved, a restriction will be automatically enabled. For five minutes no additional transactions will be allowed from the IP address of the unapproved transaction. You may modify the length of the restriction period as you deem appropriate.
- Minimum Sale (currently set to $2.01) - This will prevent any transaction below $2.01 from being processed through your account. You may change this amount to match the lowest-priced item on your site.
- Receive Failure Emails (currently set to ON) - An email will be sent to you each time a transaction is attempted and is declined for any reason. You should review these emails often as they are an indicator of potential fraud.

These are only a few of the optional anti-fraud features that are available through the Account Settings of your Control Panel. You are not required to use any of these features, but doing so will help mitigate your risk. Please review the Fraud Controls section of the documentation to determine if there are other features that you would like to use.

Questions?

If you have any questions related to this email please do not reply to this message. This is a system-generated email message; replies are not seen by an actual person. Please contact customer service if you have any questions or submit a support ticket at http://support.itransact.com

Card Processing Setup:

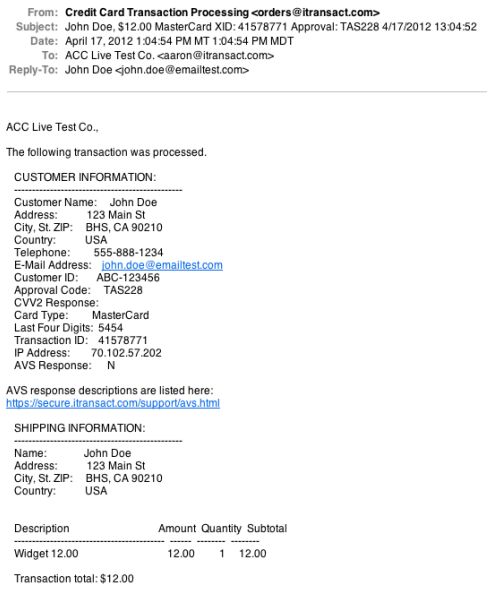
Setup Field	Setup Value
Processor	VITAL/VISANET
Agent Number	000000
Bank ID Number / BIN	999995
Category SIC/MCC Code	5968
Chain Number	111111
Merchant ID	888000000890
Merchant Location	Gainesville
Merchant Service Phone	352-3766228
Merchant State	FL
Merchant Postal Code	32606
Merchant Timezone	M
Store ID	0068
Terminal ID	12345678
Terminal Number	1515

When credit card processing is activated in the gateway this email is sent. It includes some instructions and details the merchant account information that has been built into the software. Once this is received, you can begin to accept credit card transactions.

14.2. Sale Transaction Confirmation Emails

- Merchant CC Sale Confirmation - Sent to you with each credit card transaction

Figure 2.62. Merchant CC Sale Confirmation



From: **Credit Card Transaction Processing <orders@itransact.com>**
Subject: John Doe, $12.00 MasterCard XID: 41578771 Approval: TAS228 4/17/2012 13:04:52
Date: April 17, 2012 1:04:54 PM MT 1:04:54 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: John Doe <john.doe@emailtest.com>

ACC Live Test Co.,

The following transaction was processed.

CUSTOMER INFORMATION:

Customer Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: john.doe@emailtest.com
Customer ID: ABC-123456
Approval Code: TAS228
CVV2 Response:
Card Type: MasterCard
Last Four Digits: 5454
Transaction ID: 41578771
IP Address: 70.102.57.202
AVS Response: N

AVS response descriptions are listed here:
https://secure.itransact.com/support/avs.html

SHIPPING INFORMATION:

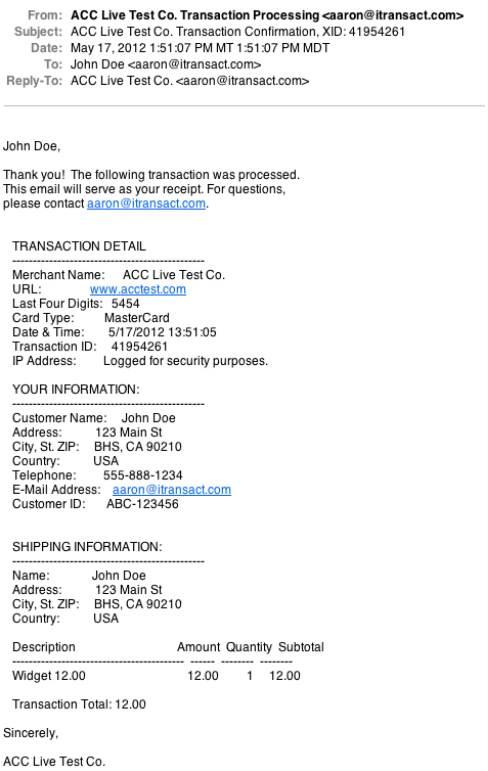
Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Widget 12.00	12.00	1	12.00

Transaction total: $12.00

- Customer CC Sale Confirmation - Sent to your customer if your settings [24] are activated

Figure 2.63. Customer CC Sale Confirmation



From: **ACC Live Test Co. Transaction Processing <aaron@itransact.com>**
Subject: ACC Live Test Co. Transaction Confirmation, XID: 41954261
Date: May 17, 2012 1:51:07 PM MT 1:51:07 PM MDT
To: John Doe <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

John Doe,

Thank you! The following transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

TRANSACTION DETAIL

Merchant Name: ACC Live Test Co.
URL: www.acctest.com
Last Four Digits: 5454
Card Type: MasterCard
Date & Time: 5/17/2012 13:51:05
Transaction ID: 41954261
IP Address: Logged for security purposes.

YOUR INFORMATION:

Customer Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: aaron@itransact.com
Customer ID: ABC-123456

SHIPPING INFORMATION:

Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Widget 12.00	12.00	1	12.00

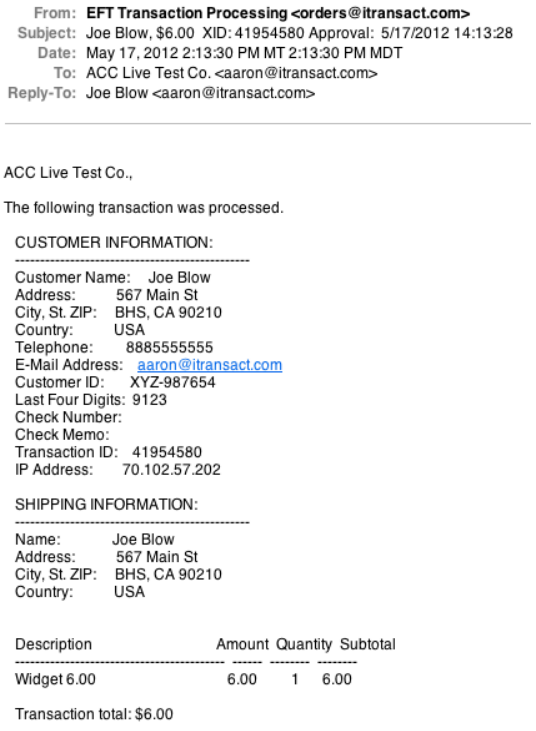
Transaction Total: 12.00

Sincerely,

ACC Live Test Co.

- Merchant Check Sale Confirmation - Sent to you with each check transaction

Figure 2.64. Merchant Check Sale Confirmation



From: **EFT Transaction Processing <orders@itransact.com>**
Subject: Joe Blow, $6.00 XID: 41954580 Approval: 5/17/2012 14:13:28
Date: May 17, 2012 2:13:30 PM MT 2:13:30 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Joe Blow <aaron@itransact.com>

ACC Live Test Co.,

The following transaction was processed.

CUSTOMER INFORMATION:

Customer Name: Joe Blow
Address: 567 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885555555
E-Mail Address: aaron@itransact.com
Customer ID: XYZ-987654
Last Four Digits: 9123
Check Number:
Check Memo:
Transaction ID: 41954580
IP Address: 70.102.57.202

SHIPPING INFORMATION:

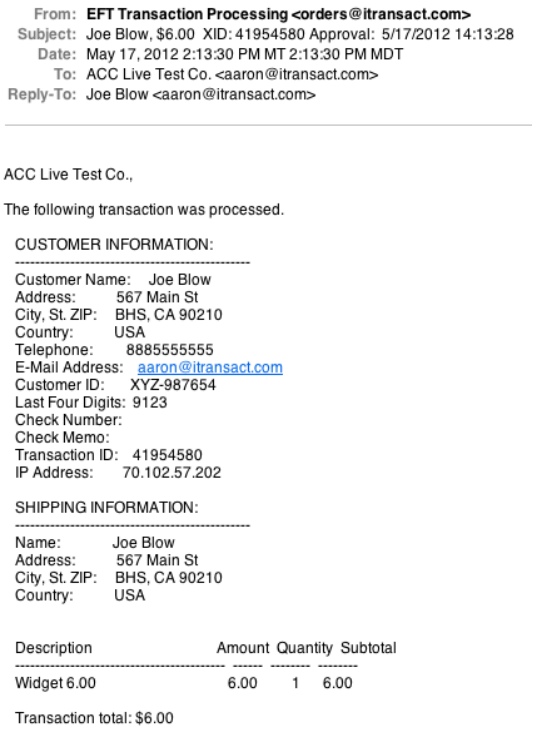
Name: Joe Blow
Address: 567 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Widget 6.00	6.00	1	6.00

Transaction total: $6.00

- Customer Check Sale Confirmation - Sent to your customer if your settings [24] are activated

Figure 2.65. Customer Check Sale Confirmation



From: **EFT Transaction Processing <orders@itransact.com>**
Subject: Joe Blow, $6.00 XID: 41954580 Approval: 5/17/2012 14:13:28
Date: May 17, 2012 2:13:30 PM MT 2:13:30 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Joe Blow <aaron@itransact.com>

ACC Live Test Co.,

The following transaction was processed.

CUSTOMER INFORMATION:

Customer Name: Joe Blow
Address: 567 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885555555
E-Mail Address: aaron@itransact.com
Customer ID: XYZ-987654
Last Four Digits: 9123
Check Number:
Check Memo:
Transaction ID: 41954580
IP Address: 70.102.57.202

SHIPPING INFORMATION:

Name: Joe Blow
Address: 567 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

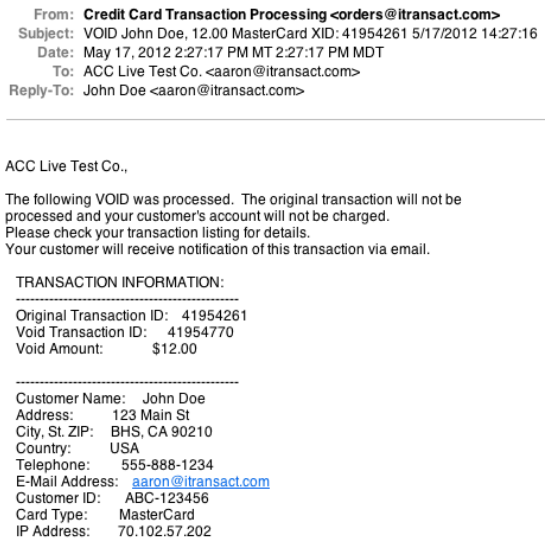
Description	Amount	Quantity	Subtotal
Widget 6.00	6.00	1	6.00

Transaction total: $6.00

14.3. Void Transaction Confirmation Emails

- Merchant Void Confirmation - Sent to you with each voided transaction

Figure 2.66. Merchant Void Confirmation



From: **Credit Card Transaction Processing <orders@itransact.com>**
Subject: VOID John Doe, 12.00 MasterCard XID: 41954261 5/17/2012 14:27:16
Date: May 17, 2012 2:27:17 PM MT 2:27:17 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: John Doe <aaron@itransact.com>

ACC Live Test Co.,

The following VOID was processed. The original transaction will not be processed and your customer's account will not be charged.
Please check your transaction listing for details.
Your customer will receive notification of this transaction via email.

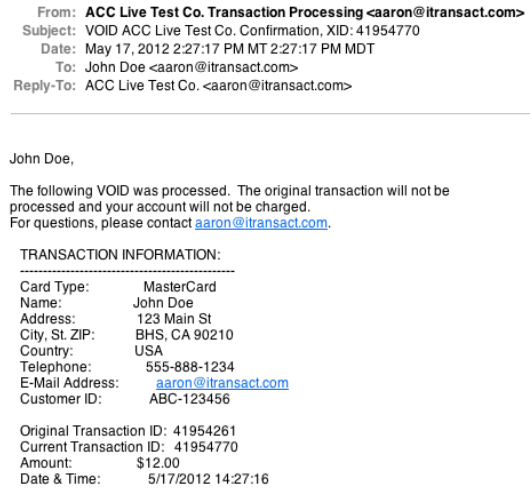
TRANSACTION INFORMATION:

Original Transaction ID: 41954261
Void Transaction ID: 41954770
Void Amount: $12.00

Customer Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: aaron@itransact.com
Customer ID: ABC-123456
Card Type: MasterCard
IP Address: 70.102.57.202

- Customer Void Confirmation - Sent to your customer if your settings [24] are activated

Figure 2.67. Customer Void Confirmation



From: **ACC Live Test Co. Transaction Processing <aaron@itransact.com>**
Subject: VOID ACC Live Test Co. Confirmation, XID: 41954770
Date: May 17, 2012 2:27:17 PM MT 2:27:17 PM MDT
To: John Doe <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

John Doe,

The following VOID was processed. The original transaction will not be processed and your account will not be charged.
For questions, please contact aaron@itransact.com.

TRANSACTION INFORMATION:

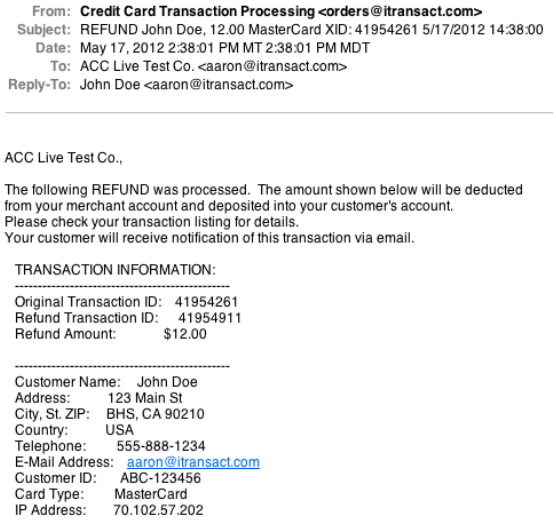
Card Type: MasterCard
Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: aaron@itransact.com
Customer ID: ABC-123456

Original Transaction ID: 41954261
Current Transaction ID: 41954770
Amount: $12.00
Date & Time: 5/17/2012 14:27:16

14.4. Refund Transaction Confirmation Emails

- Merchant Refund Confirmation - Sent to you with each refund transaction

Figure 2.68. Merchant Refund Confirmation



From: **Credit Card Transaction Processing <orders@itransact.com>**
Subject: REFUND John Doe, 12.00 MasterCard XID: 41954261 5/17/2012 14:38:00
Date: May 17, 2012 2:38:01 PM MT 2:38:01 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: John Doe <aaron@itransact.com>

ACC Live Test Co.,

The following REFUND was processed. The amount shown below will be deducted from your merchant account and deposited into your customer's account.
Please check your transaction listing for details.
Your customer will receive notification of this transaction via email.

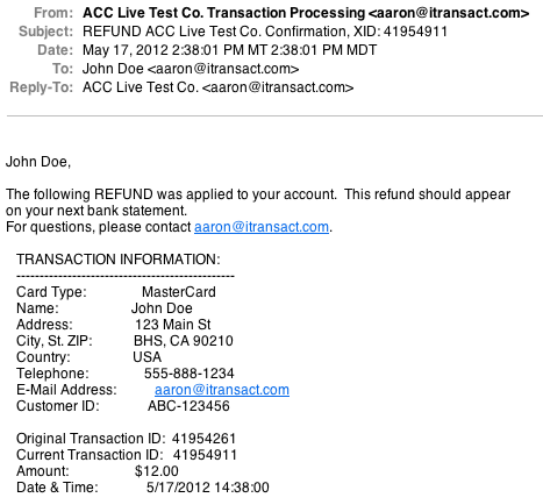
TRANSACTION INFORMATION:

Original Transaction ID: 41954261
Refund Transaction ID: 41954911
Refund Amount: $12.00

Customer Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: aaron@itransact.com
Customer ID: ABC-123456
Card Type: MasterCard
IP Address: 70.102.57.202

- Customer Refund Confirmation - Sent to your customer if your settings [24] are activated

Figure 2.69. Customer Refund Confirmation



From: **ACC Live Test Co. Transaction Processing <aaron@itransact.com>**
Subject: REFUND ACC Live Test Co. Confirmation, XID: 41954911
Date: May 17, 2012 2:38:01 PM MT 2:38:01 PM MDT
To: John Doe <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

John Doe,

The following REFUND was applied to your account. This refund should appear on your next bank statement.
For questions, please contact aaron@itransact.com.

TRANSACTION INFORMATION:

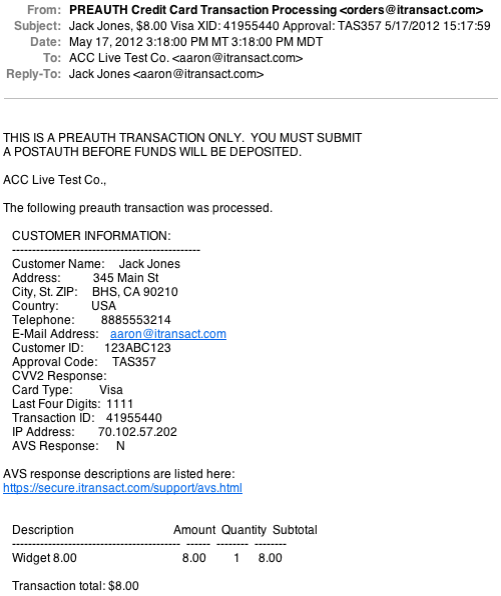
Card Type: MasterCard
Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: aaron@itransact.com
Customer ID: ABC-123456

Original Transaction ID: 41954261
Current Transaction ID: 41954911
Amount: $12.00
Date & Time: 5/17/2012 14:38:00

14.5. Preauth Transaction Confirmation Emails

- Merchant Preauth Confirmation - Sent to you with each preauth transaction

Figure 2.70. Merchant Preauth Confirmation



From: **PREAUTH Credit Card Transaction Processing <orders@itransact.com>**
Subject: Jack Jones, $8.00 Visa XID: 41955440 Approval: TAS357 5/17/2012 15:17:59
Date: May 17, 2012 3:18:00 PM MT 3:18:00 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Jack Jones <aaron@itransact.com>

THIS IS A PREAUTH TRANSACTION ONLY. YOU MUST SUBMIT
A POSTAUTH BEFORE FUNDS WILL BE DEPOSITED.

ACC Live Test Co.,

The following preauth transaction was processed.

CUSTOMER INFORMATION:

Customer Name: Jack Jones
Address: 345 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885553214
E-Mail Address: aaron@itransact.com
Customer ID: 123ABC123
Approval Code: TAS357
CVV2 Response:
Card Type: Visa
Last Four Digits: 1111
Transaction ID: 41955440
IP Address: 70.102.57.202
AVS Response: N

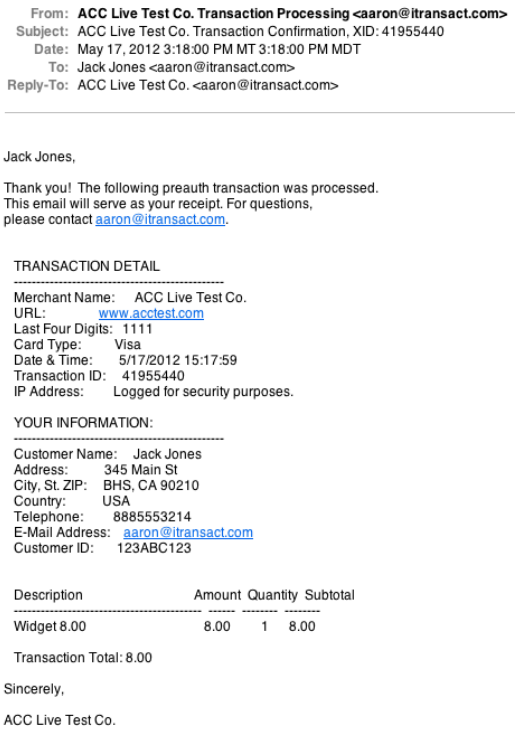
AVS response descriptions are listed here:
https://secure.itransact.com/support/avs.html

Description	Amount	Quantity	Subtotal
Widget 8.00	8.00	1	8.00

Transaction total: $8.00

- Customer Preauth Confirmation - Sent to your customer if your settings [22] are activated

Figure 2.71. Customer Preauth Confirmation



From: **ACC Live Test Co. Transaction Processing <aaron@itransact.com>**
Subject: ACC Live Test Co. Transaction Confirmation, XID: 41955440
Date: May 17, 2012 3:18:00 PM MT 3:18:00 PM MDT
To: Jack Jones <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

Jack Jones,

Thank you! The following preauth transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

TRANSACTION DETAIL

Merchant Name: ACC Live Test Co.
URL: www.acctest.com
Last Four Digits: 1111
Card Type: Visa
Date & Time: 5/17/2012 15:17:59
Transaction ID: 41955440
IP Address: Logged for security purposes.

YOUR INFORMATION:

Customer Name: Jack Jones
Address: 345 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885553214
E-Mail Address: aaron@itransact.com
Customer ID: 123ABC123

Description	Amount	Quantity	Subtotal
Widget 8.00	8.00	1	8.00

Transaction Total: 8.00

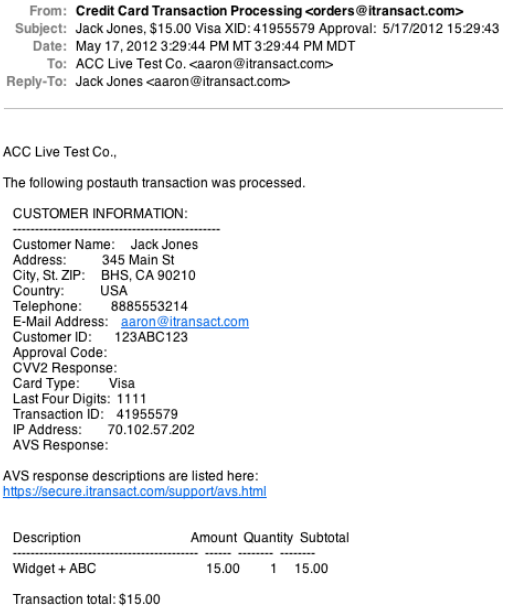
Sincerely,

ACC Live Test Co.

14.6. Postauth Transaction Confirmation Emails

- Merchant Postauth Confirmation - Sent to you with each postauth transaction

Figure 2.72. Merchant Postauth Transaction Confirmation



From: **Credit Card Transaction Processing <orders@itransact.com>**
Subject: Jack Jones, $15.00 Visa XID: 41955579 Approval: 5/17/2012 15:29:43
Date: May 17, 2012 3:29:44 PM MT 3:29:44 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Jack Jones <aaron@itransact.com>

ACC Live Test Co.,

The following postauth transaction was processed.

CUSTOMER INFORMATION:

Customer Name: Jack Jones
Address: 345 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885553214
E-Mail Address: aaron@itransact.com
Customer ID: 123ABC123
Approval Code:
CVV2 Response:
Card Type: Visa
Last Four Digits: 1111
Transaction ID: 41955579
IP Address: 70.102.57.202
AVS Response:

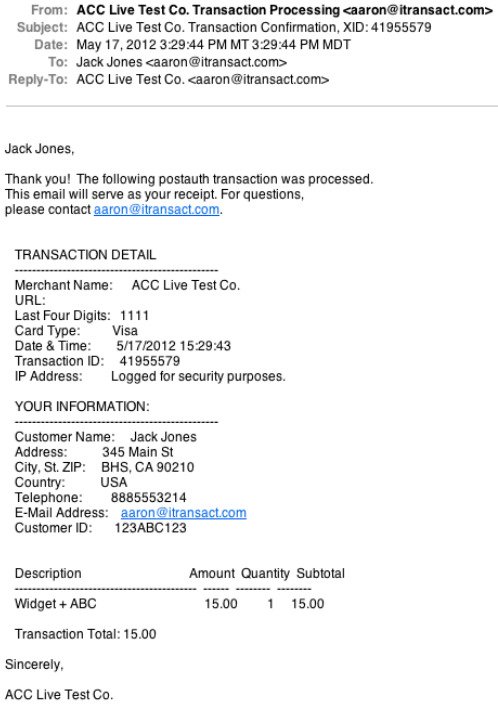
AVS response descriptions are listed here:
https://secure.itransact.com/support/avs.html

Description	Amount	Quantity	Subtotal
Widget + ABC	15.00	1	15.00

Transaction total: $15.00

- Customer Postauth Confirmation - Sent to your customer if your settings [24] are activated

Figure 2.73. Customer Postauth Confirmation



From: **ACC Live Test Co. Transaction Processing <aaron@itransact.com>**
Subject: ACC Live Test Co. Transaction Confirmation, XID: 41955579
Date: May 17, 2012 3:29:44 PM MT 3:29:44 PM MDT
To: Jack Jones <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

Jack Jones,

Thank you! The following postauth transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

TRANSACTION DETAIL

Merchant Name: ACC Live Test Co.
URL:
Last Four Digits: 1111
Card Type: Visa
Date & Time: 5/17/2012 15:29:43
Transaction ID: 41955579
IP Address: Logged for security purposes.

YOUR INFORMATION:

Customer Name: Jack Jones
Address: 345 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885553214
E-Mail Address: aaron@itransact.com
Customer ID: 123ABC123

Description	Amount	Quantity	Subtotal
Widget + ABC	15.00	1	15.00

Transaction Total: 15.00

Sincerely,

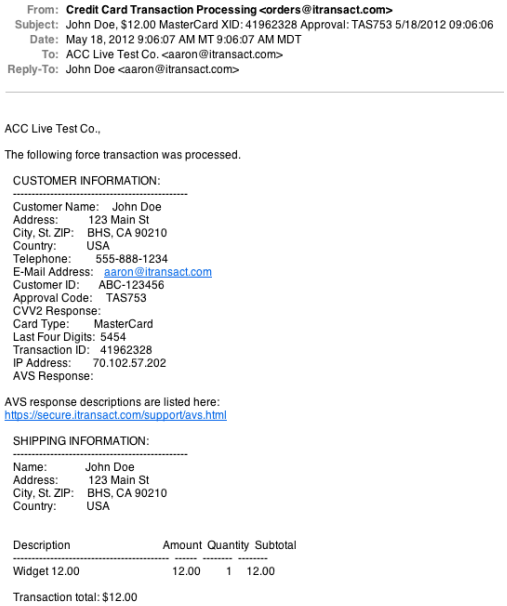
ACC Live Test Co.

.

14.7. Force Transaction Confirmation Emails

- Merchant Force Confirmation - Sent to you with each Force transaction

Figure 2.74. Merchant Force Confirmation



From: **Credit Card Transaction Processing <orders@itransact.com>**
Subject: John Doe, $12.00 MasterCard XID: 41962328 Approval: TAS753 5/18/2012 09:06:06
Date: May 18, 2012 9:06:07 AM MT 9:06:07 AM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: John Doe <aaron@itransact.com>

ACC Live Test Co.,

The following force transaction was processed.

CUSTOMER INFORMATION:

Customer Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: aaron@itransact.com
Customer ID: ABC-123456
Approval Code: TAS753
CVV2 Response:
Card Type: MasterCard
Last Four Digits: 5454
Transaction ID: 41962328
IP Address: 70.102.57.202
AVS Response:

AVS response descriptions are listed here:
https://secure.itransact.com/support/avs.html

SHIPPING INFORMATION:

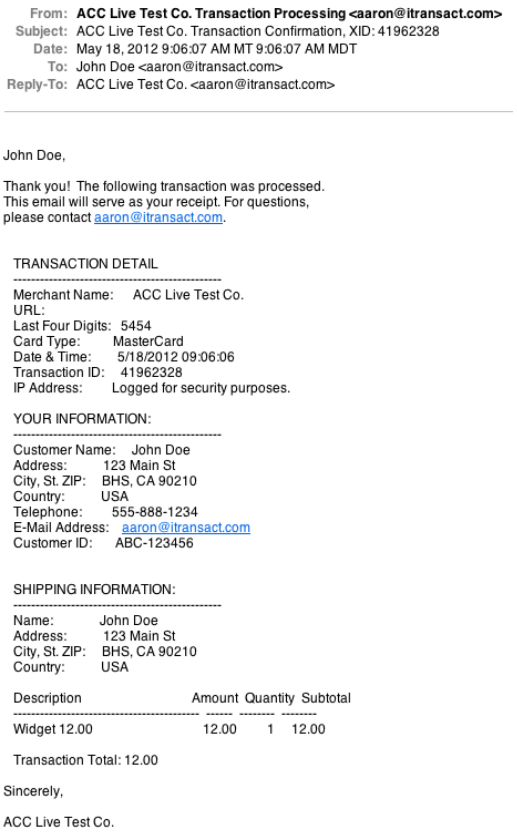
Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Widget 12.00	12.00	1	12.00

Transaction total: $12.00

- Customer Force Confirmation - Sent to your customer if your settings [24] are activated

Figure 2.75. Customer Force Confirmation



From: **ACC Live Test Co. Transaction Processing <aaron@itransact.com>**
Subject: ACC Live Test Co. Transaction Confirmation, XID: 41962328
Date: May 18, 2012 9:06:07 AM MT 9:06:07 AM MDT
To: John Doe <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

John Doe,

Thank you! The following transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

TRANSACTION DETAIL

Merchant Name: ACC Live Test Co.
URL:
Last Four Digits: 5454
Card Type: MasterCard
Date & Time: 5/18/2012 09:06:06
Transaction ID: 41962328
IP Address: Logged for security purposes.

YOUR INFORMATION:

Customer Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: aaron@itransact.com
Customer ID: ABC-123456

SHIPPING INFORMATION:

Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Widget 12.00	12.00	1	12.00

Transaction Total: 12.00

Sincerely,

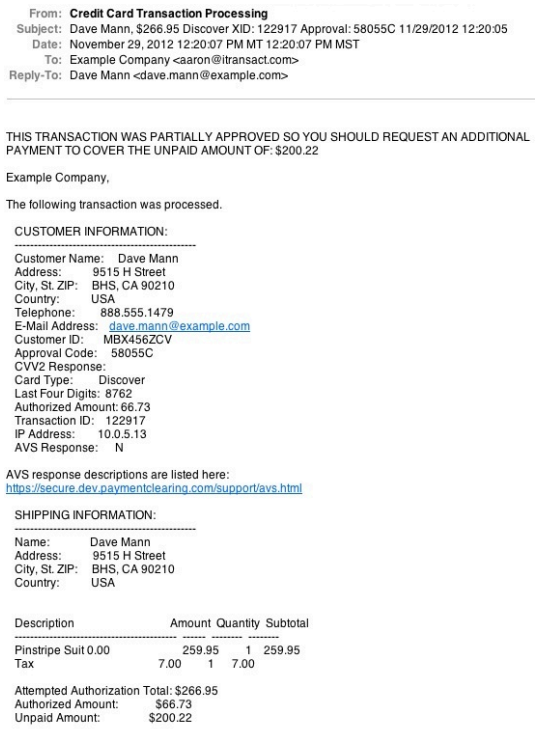
ACC Live Test Co.

.

14.8. Partial Authorization Notification Emails

- Merchant Partial Authorization - Sent to you when a retail transaction is partially authorized

Figure 2.76. Merchant Partial Authorization



From: **Credit Card Transaction Processing**
Subject: Dave Mann, $266.95 Discover XID: 122917 Approval: 58055C 11/29/2012 12:20:05
Date: November 29, 2012 12:20:07 PM MT 12:20:07 PM MST
To: Example Company <aaron@itransact.com>
Reply-To: Dave Mann <dave.mann@example.com>

THIS TRANSACTION WAS PARTIALLY APPROVED SO YOU SHOULD REQUEST AN ADDITIONAL PAYMENT TO COVER THE UNPAID AMOUNT OF: $200.22

Example Company,

The following transaction was processed.

CUSTOMER INFORMATION:

Customer Name: Dave Mann
Address: 9515 H Street
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 888.555.1479
E-Mail Address: dave.mann@example.com
Customer ID: MBX456ZCV
Approval Code: 58055C
CVV2 Response:
Card Type: Discover
Last Four Digits: 8762
Authorized Amount: 66.73
Transaction ID: 122917
IP Address: 10.0.5.13
AVS Response: N

AVS response descriptions are listed here:
https://secure.dev.paymentclearing.com/support/avs.html

SHIPPING INFORMATION:

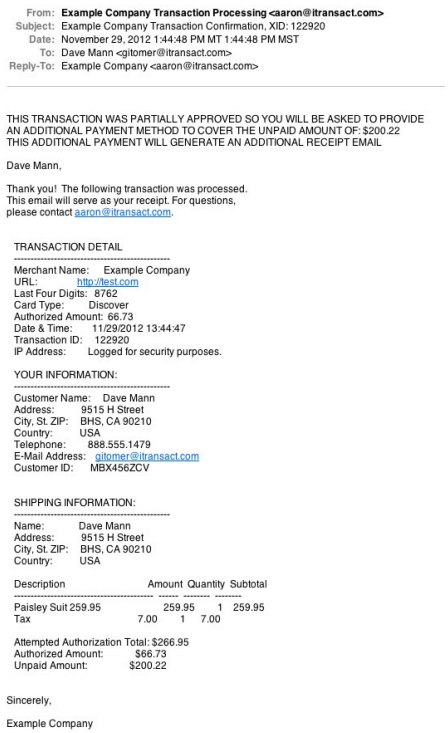
Name: Dave Mann
Address: 9515 H Street
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Pinstripe Suit 0.00	259.95	1	259.95
Tax	7.00	1	7.00

Attempted Authorization Total: $266.95
Authorized Amount: $66.73
Unpaid Amount: $200.22

- Customer Partial Authorization - Sent to your customer if your settings are activated

Figure 2.77. Customer Partial Authorization



From: **Example Company Transaction Processing <aaron@itransact.com>**
Subject: Example Company Transaction Confirmation, XID: 122920
Date: November 29, 2012 1:44:48 PM MT 1:44:48 PM MST
To: Dave Mann <gitomer@itransact.com>
Reply-To: Example Company <aaron@itransact.com>

THIS TRANSACTION WAS PARTIALLY APPROVED SO YOU WILL BE ASKED TO PROVIDE AN ADDITIONAL PAYMENT METHOD TO COVER THE UNPAID AMOUNT OF: $200.22
THIS ADDITIONAL PAYMENT WILL GENERATE AN ADDITIONAL RECEIPT EMAIL

Dave Mann,

Thank you! The following transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

TRANSACTION DETAIL

Merchant Name: Example Company
URL: http://test.com
Last Four Digits: 8762
Card Type: Discover
Authorized Amount: 66.73
Date & Time: 11/29/2012 13:44:47
Transaction ID: 122920
IP Address: Logged for security purposes.

YOUR INFORMATION:

Customer Name: Dave Mann
Address: 9515 H Street
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 888.555.1479
E-Mail Address: gitomer@itransact.com
Customer ID: MBX456ZCV

SHIPPING INFORMATION:

Name: Dave Mann
Address: 9515 H Street
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Paisley Suit 259.95	259.95	1	259.95
Tax	7.00	1	7.00

Attempted Authorization Total: $266.95
Authorized Amount: $66.73
Unpaid Amount: $200.22

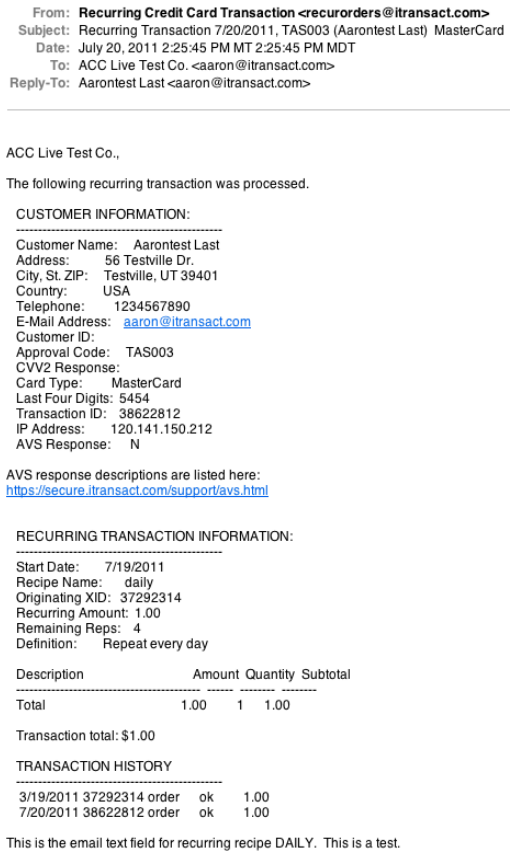
Sincerely,

Example Company

14.9. Recurring Transaction Confirmation Emails

- Merchant Recurring CC Confirmation - Sent to you with each recurring transaction

Figure 2.78. Merchant Recurring CC Confirmation



From: **Recurring Credit Card Transaction <recurorders@itransact.com>**
Subject: Recurring Transaction 7/20/2011, TAS003 (Aarontest Last) MasterCard
Date: July 20, 2011 2:25:45 PM MT 2:25:45 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Aarontest Last <aaron@itransact.com>

ACC Live Test Co.,

The following recurring transaction was processed.

CUSTOMER INFORMATION:

Customer Name: Aarontest Last
Address: 56 Testville Dr.
City, St. ZIP: Testville, UT 39401
Country: USA
Telephone: 1234567890
E-Mail Address: aaron@itransact.com
Customer ID:
Approval Code: TAS003
CVV2 Response:
Card Type: MasterCard
Last Four Digits: 5454
Transaction ID: 38622812
IP Address: 120.141.150.212
AVS Response: N

AVS response descriptions are listed here:
https://secure.itransact.com/support/avs.html

RECURRING TRANSACTION INFORMATION:

Start Date: 7/19/2011
Recipe Name: daily
Originating XID: 37292314
Recurring Amount: 1.00
Remaining Reps: 4
Definition: Repeat every day

Description	Amount	Quantity	Subtotal
Total	1.00	1	1.00

Transaction total: $1.00

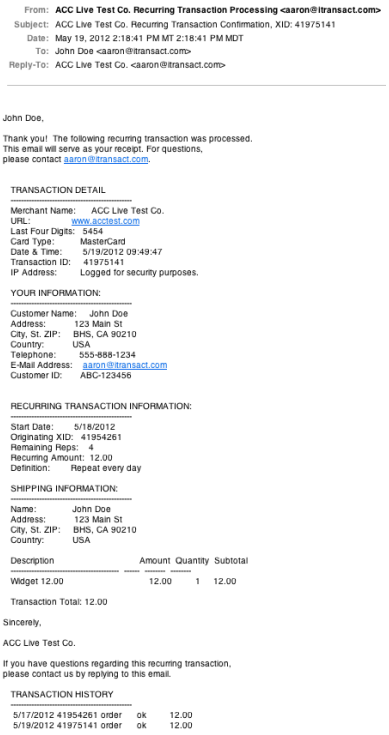
TRANSACTION HISTORY

3/19/2011 37292314 order ok 1.00
7/20/2011 38622812 order ok 1.00

This is the email text field for recurring recipe DAILY. This is a test.

- Customer Recurring CC Confirmation - Sent to your customer if your settings [24] are activated

Figure 2.79. Customer Recurring CC Confirmation



From: **ACC Live Test Co. Recurring Transaction Processing <aaron@itransact.com>**
Subject: ACC Live Test Co. Recurring Transaction Confirmation, XID: 41975141
Date: May 19, 2012 2:18:41 PM MT 2:18:41 PM MDT
To: John Doe <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

John Doe,

Thank you! The following recurring transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

TRANSACTION DETAIL
Merchant Name: ACC Live Test Co.
URL: www.acctest.com
Last Four Digits: 5454
Card Type: MasterCard
Date & Time: 5/19/2012 09:49:47
Transaction ID: 41975141
IP Address: Logged for security purposes.

YOUR INFORMATION:
Customer Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 555-888-1234
E-Mail Address: aaron@itransact.com
Customer ID: ABC-123456

RECURRING TRANSACTION INFORMATION:
Start Date: 5/18/2012
Originating XID: 41954261
Remaining Reps: 4
Recurring Amount: 12.00
Definition: Repeat every day

SHIPPING INFORMATION:
Name: John Doe
Address: 123 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Widget 12.00	12.00	1	12.00

Transaction Total: 12.00

Sincerely,

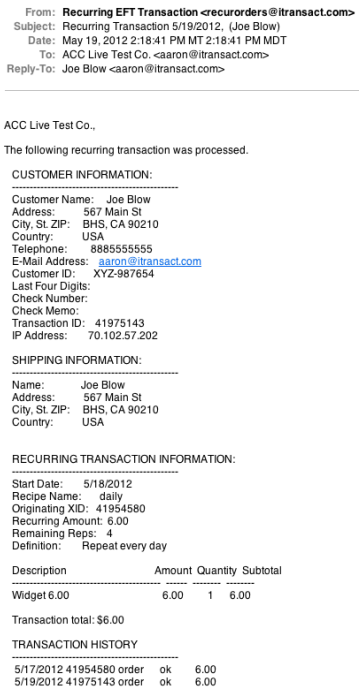
ACC Live Test Co.

If you have questions regarding this recurring transaction,
please contact us by replying to this email.

TRANSACTION HISTORY
5/17/2012 41954261 order ok 12.00
5/19/2012 41975141 order ok 12.00

- Merchant Recurring Check Confirmation - Sent to you with each recurring transaction

Figure 2.80. Merchant Recurring Check Confirmation



From: **Recurring EFT Transaction <recurorders@itransact.com>**
Subject: Recurring Transaction 5/19/2012, (Joe Blow)
Date: May 19, 2012 2:18:41 PM MT 2:18:41 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Joe Blow <aaron@itransact.com>

ACC Live Test Co.,

The following recurring transaction was processed.

CUSTOMER INFORMATION:
Customer Name: Joe Blow
Address: 567 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885555555
E-Mail Address: aaron@itransact.com
Customer ID: XYZ-987654
Last Four Digits:
Check Number:
Check Memo:
Transaction ID: 41975143
IP Address: 70.102.57.202

SHIPPING INFORMATION:
Name: Joe Blow
Address: 567 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

RECURRING TRANSACTION INFORMATION:
Start Date: 5/18/2012
Recipe Name: daily
Originating XID: 41954580
Recurring Amount: 6.00
Remaining Reps: 4
Definition: Repeat every day

Description	Amount	Quantity	Subtotal
Widget 6.00	6.00	1	6.00

Transaction total: $6.00

TRANSACTION HISTORY
5/17/2012 41954580 order ok 6.00
5/19/2012 41975143 order ok 6.00

- Customer Recurring Check Confirmation - Sent to your customer if your settings are activated

Figure 2.81. Customer Recurring Check Confirmation

From: **ACC Live Test Co. Recurring Transaction Processing <aaron@itransact.com>**
Subject: ACC Live Test Co. Recurring Transaction Confirmation, XID: 41975143
Date: May 19, 2012 2:18:41 PM MT 2:18:41 PM MDT
To: Joe Blow <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

Joe Blow,

Thank you! The following recurring transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

This is the email text field for recurring recipe DAILY. This is a test.

TRANSACTION DETAIL

Merchant Name: ACC Live Test Co.
URL: www.acctest.com
Last Four Digits:
Last Four Digits:
Check Number:
Check Memo:
Date & Time: 5/19/2012 09:49:49
Transaction ID: 41975143
IP Address: Logged for security purposes.

YOUR INFORMATION:

Customer Name: Joe Blow
Address: 567 Main St
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885555555
E-Mail Address: aaron@itransact.com
Customer ID: XYZ-987654

RECURRING TRANSACTION INFORMATION:

Start Date: 5/18/2012
Originating XID: 41954580
Remaining Reps: 4
Recurring Amount: 6.00
Definition: Repeat every day

SHIPPING INFORMATION:

Name: Joe Blow
Address: 567 Main St
City, St. ZIP: BHS, CA 90210
Country: USA

Description	Amount	Quantity	Subtotal
Widget 6.00	6.00	1	6.00

Transaction Total: 6.00

Sincerely,

ACC Live Test Co.

If you have questions regarding this recurring transaction,
please contact us by replying to this email.

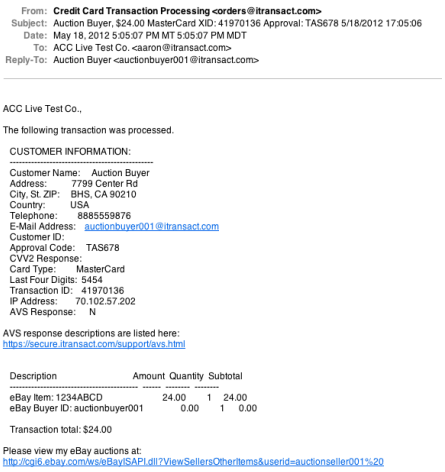
TRANSACTION HISTORY

5/17/2012 41954580 order ok 6.00
5/19/2012 41975143 order ok 6.00

14.10. Recurring Transaction Confirmations with Update Links

- Merchant Recurring CC Confirmation - Sent to you with each recurring transaction

Figure 2.82. Merchant Recurring CC Confirmation



From: **Credit Card Transaction Processing <orders@itransact.com>**
Subject: Auction Buyer, $24.00 MasterCard XID: 41970136 Approval: TAS678 5/18/2012 17:05:06
Date: May 18, 2012 5:05:07 PM MT 5:05:07 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Auction Buyer <auctionbuyer001@itransact.com>

ACC Live Test Co.,

The following transaction was processed.

CUSTOMER INFORMATION:

Customer Name: Auction Buyer
Address: 7799 Center Rd
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885559876
E-Mail Address: auctionbuyer001@itransact.com
Customer ID:
Approval Code: TAS678
CVV2 Response:
Card Type: MasterCard
Last Four Digits: 5454
Transaction ID: 41970136
IP Address: 70.102.57.202
AVS Response: N

AVS response descriptions are listed here:
https://secure.itransact.com/support/avs.html

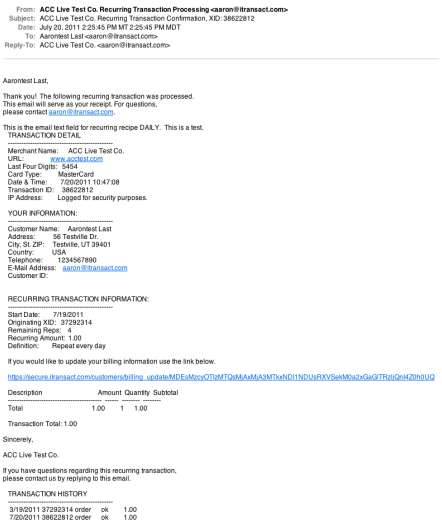
Description	Amount	Quantity	Subtotal
eBay Item: 1234ABCD	24.00	1	24.00
eBay Buyer ID: auctionbuyer001	0.00	1	0.00

Transaction total: $24.00

Please view my eBay auctions at:
http://cgi6.ebay.com/ws/eBayISAPI.dll?ViewSellersOtherItems&userid=auctionseller001%20

- Customer Recurring CC Confirmation - Sent to your customer if your settings [24] are activated

Figure 2.83. Customer Recurring CC Confirmation



From: **ACC Live Test Co. Recurring Transaction Processing <aaron@itransact.com>**
Subject: ACC Live Test Co. Recurring Transaction Confirmation, XID: 38622812
Date: July 20, 2011 2:25:45 PM MT 2:25:45 PM MDT
To: Aarontest Last <aaron@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

Aarontest Last,

Thank you! The following recurring transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

This is the email test field for recurring recipe DAILY. This is a test.
TRANSACTION DETAIL

Merchant Name: ACC Live Test Co.
URL: www.acctest.com
Last Four Digits: 5454
Card Type: MasterCard
Date & Time: 7/20/2011 10:47:08
Transaction ID: 38622812
IP Address: Logged for security purposes.

YOUR INFORMATION:

Customer Name: Aarontest Last
Address: 56 Testville Dr.
City, St, ZIP: Testville, UT 39401
Country: USA
Telephone: 1234567890
E-Mail Address: aaron@itransact.com
Customer ID:

RECURRING TRANSACTION INFORMATION:

Start Date: 7/19/2011
Originating XID: 37292314
Remaining Reps: 4
Recurring Amount: 1.00
Definition: Repeat every day

If you would like to update your billing information use the link below.

https://secure.itransact.com/customers/billing_update/MDEsMzcyOTIzMTQsMjAxMiA3MTkxND1tNDUsRXVSekM0a2xGaGlTRzljQnI4Z0h0UQ

Description	Amount	Quantity	Subtotal
Total	1.00	1	1.00

Transaction Total: 1.00

Sincerely,

ACC Live Test Co.

If you have questions regarding this recurring transaction,
please contact us by replying to this email.

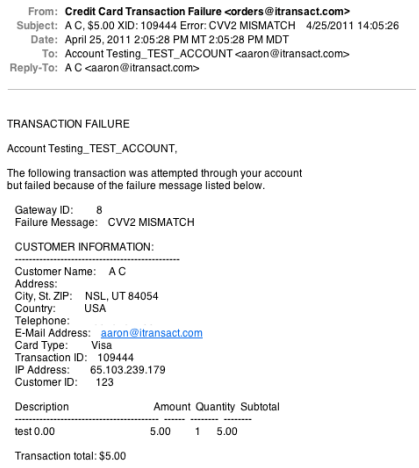
TRANSACTION HISTORY

3/19/2011 37292314 order ok 1.00
7/20/2011 38622812 order ok 1.00

14.11. Failed Transaction Notification Emails

- Merchant Notification of Failed CC Transaction - Sent to you if your settings are activated

Figure 2.84. Merchant Notification of Failed CC Transaction



From: **Credit Card Transaction Failure <orders@itransact.com>**
Subject: A C, $5.00 XID: 109444 Error: CVV2 MISMATCH 4/25/2011 14:05:26
Date: April 25, 2011 2:05:28 PM MT 2:05:28 PM MDT
To: Account Testing_TEST_ACCOUNT <aaron@itransact.com>
Reply-To: A C <aaron@itransact.com>

TRANSACTION FAILURE

Account Testing_TEST_ACCOUNT,

The following transaction was attempted through your account
but failed because of the failure message listed below.

Gateway ID: 8
Failure Message: CVV2 MISMATCH

CUSTOMER INFORMATION:

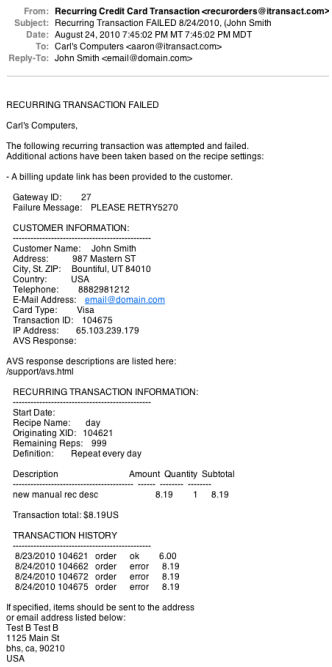
Customer Name: A C
Address:
City, St, ZIP: NSL, UT 84054
Country: USA
Telephone:
E-Mail Address: aaron@itransact.com
Card Type: Visa
Transaction ID: 109444
IP Address: 65.103.239.179
Customer ID: 123

Description	Amount	Quantity	Subtotal
test 0.00	5.00	1	5.00

Transaction total: $5.00

- Merchant Notification of Failed Recurring Transaction - Sent to you if your settings [22] are activated

Figure 2.85. Merchant Notification of Failed Recurring Transaction



From: **Recurring Credit Card Transaction <recurorders@itransact.com>**
Subject: Recurring Transaction FAILED 8/24/2010, (John Smith
Date: August 24, 2010 7:45:02 PM MT 7:45:02 PM MDT
To: Carl's Computers <aaron@itransact.com>
Reply-To: John Smith <email@domain.com>

RECURRING TRANSACTION FAILED

Carl's Computers,

The following recurring transaction was attempted and failed.
Additional actions have been taken based on the recipe settings:

- A billing update link has been provided to the customer.

Gateway ID: 27
Failure Message: PLEASE RETRY5270

CUSTOMER INFORMATION:
Customer Name: John Smith
Address: 987 Mastern ST
City, St. ZIP: Bountiful, UT 84010
Country: USA
Telephone: 8882981212
E-Mail Address: email@domain.com
Card Type: Visa
Transaction ID: 104675
IP Address: 65.103.239.179
AVS Response:

AVS response descriptions are listed here:
/support/avs.html

RECURRING TRANSACTION INFORMATION:
Start Date:
Recipe Name: day
Originating XID: 104621
Remaining Reps: 999
Definition: Repeat every day

Description	Amount	Quantity	Subtotal
new manual rec desc	8.19	1	8.19

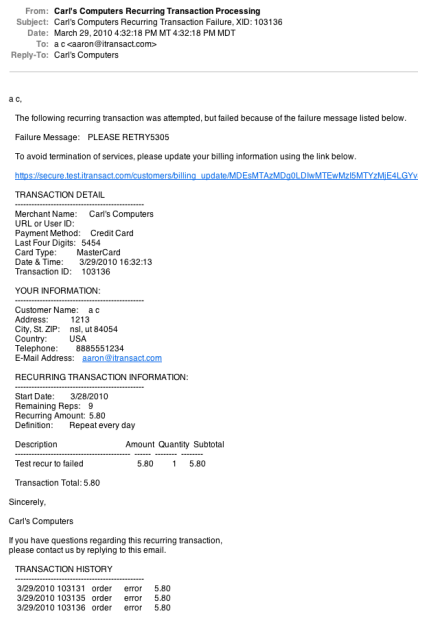
Transaction total: $8.19 US

TRANSACTION HISTORY
8/23/2010 104621 order ok 6.00
8/24/2010 104662 order error 8.19
8/24/2010 104672 order error 8.19
8/24/2010 104675 order error 8.19

If specified, items should be sent to the address
or email address listed below:
Test B Test B
1125 Main St
bhs, ca, 90210
USA

- Customer Notification of Failed Recurring - Sent if you are using Customer Update [63]

Figure 2.86. Customer Notification of Failed Recurring



From: **Carl's Computers Recurring Transaction Processing**
Subject: Carl's Computers Recurring Transaction Failure, XID: 103136
Date: March 29, 2010 4:32:18 PM MT 4:32:18 PM MDT
To: a c <aaron@itransact.com>
Reply-To: Carl's Computers

a c,

The following recurring transaction was attempted, but failed because of the failure message listed below.

Failure Message: PLEASE RETRY5305

To avoid termination of services, please update your billing information using the link below.

https://secure.test.itransact.com/customers/billing_update/MDEsMTAzMDg0LDIwMTEwMzI5MTYzMjE4LGYv

TRANSACTION DETAIL
Merchant Name: Carl's Computers
URL or User ID:
Payment Method: Credit Card
Last Four Digits: 5454
Card Type: MasterCard
Date & Time: 3/29/2010 16:32:13
Transaction ID: 103136

YOUR INFORMATION:
Customer Name: a c
Address: 1213
City, St. ZIP: nsl, ut 84054
Country: USA
Telephone: 8885551234
E-Mail Address: aaron@itransact.com

RECURRING TRANSACTION INFORMATION:
Start Date: 3/28/2010
Remaining Reps: 9
Recurring Amount: 5.80
Definition: Repeat every day

Description	Amount	Quantity	Subtotal
Test recur to failed	5.80	1	5.80

Transaction Total: 5.80

Sincerely,

Carl's Computers

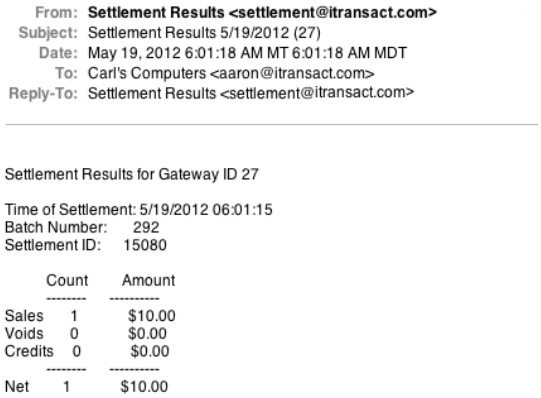
If you have questions regarding this recurring transaction,
please contact us by replying to this email.

TRANSACTION HISTORY
3/29/2010 103131 order error 5.80
3/29/2010 103135 order error 5.80
3/29/2010 103136 order error 5.80

14.12. Credit Card Settlement Notification Emails

The payment gateway sends notification emails each time a credit card settlement takes place. The information listed in the emails is generated by the response from the credit card processing network. The gateway system only reports the data that the processing network responds with. Any discrepancies need to be discussed with the processor or the merchant service provider.

Figure 2.87. Credit Card Settlement Notification



From: **Settlement Results <settlement@itransact.com>**
Subject: Settlement Results 5/19/2012 (27)
Date: May 19, 2012 6:01:18 AM MT 6:01:18 AM MDT
To: Carl's Computers <aaron@itransact.com>
Reply-To: Settlement Results <settlement@itransact.com>

Settlement Results for Gateway ID 27

Time of Settlement: 5/19/2012 06:01:15
Batch Number: 292
Settlement ID: 15080

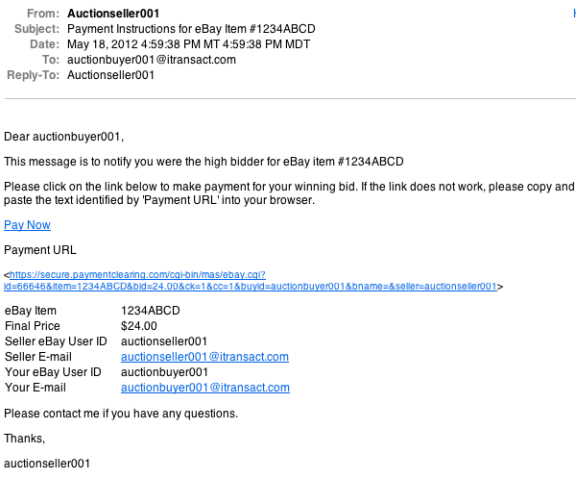
	Count	Amount
Sales	1	$10.00
Voids	0	$0.00
Credits	0	$0.00
Net	1	$10.00

14.13. Auction Payment Notification Emails

The Auction Payment [59] system uses emails to direct a winning bidder to a secure page (built dynamically when the Auction Payment request is sent) where they can complete the payment for the auction item.

- Auction Payment e-Invoice

Figure 2.88. Auction Payment e-Invoice



From: **Auctionseller001**
Subject: Payment Instructions for eBay Item #1234ABCD
Date: May 18, 2012 4:59:38 PM MT 4:59:38 PM MDT
To: auctionbuyer001@itransact.com
Reply-To: Auctionseller001

Dear auctionbuyer001,

This message is to notify you were the high bidder for eBay item #1234ABCD

Please click on the link below to make payment for your winning bid. If the link does not work, please copy and paste the text identified by 'Payment URL' into your browser.

Pay Now

Payment URL

<https://secure.paymentclearing.com/cgi-bin/mas/ebay.cgi?id=66646&item=1234ABCD&bid=24.00&ck=1&cc=1&buyid=auctionbuyer001&bname=&seller=auctionseller001>

eBay Item	1234ABCD
Final Price	$24.00
Seller eBay User ID	auctionseller001
Seller E-mail	auctionseller001@itransact.com
Your eBay User ID	auctionbuyer001
Your E-mail	auctionbuyer001@itransact.com

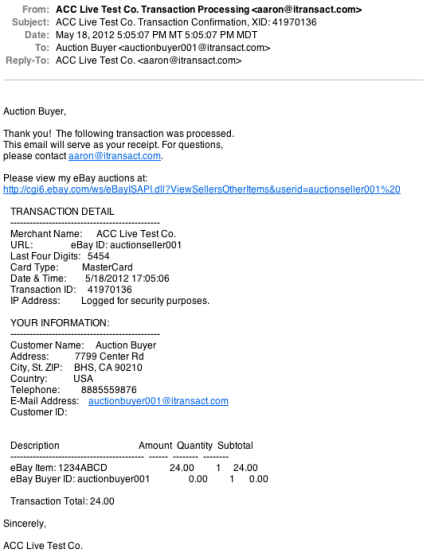
Please contact me if you have any questions.

Thanks,

auctionseller001

- Merchant Auction Payments Confirmation

Figure 2.89. Merchant Auction Payments Confirmation



From: **ACC Live Test Co. Transaction Processing <aaron@itransact.com>**
Subject: ACC Live Test Co. Transaction Confirmation, XID: 41970136
Date: May 18, 2012 5:05:07 PM MT 5:05:07 PM MDT
To: Auction Buyer <auctionbuyer001@itransact.com>
Reply-To: ACC Live Test Co. <aaron@itransact.com>

Auction Buyer,

Thank you! The following transaction was processed.
This email will serve as your receipt. For questions,
please contact aaron@itransact.com.

Please view my eBay auctions at:
http://cgi6.ebay.com/ws/eBayISAPI.dll?ViewSellersOtherItems&userid=auctionseller001%20

TRANSACTION DETAIL

Merchant Name: ACC Live Test Co.
URL: eBay ID: auctionseller001
Last Four Digits: 5454
Card Type: MasterCard
Date & Time: 5/18/2012 17:05:06
Transaction ID: 41970136
IP Address: Logged for security purposes.

YOUR INFORMATION:

Customer Name: Auction Buyer
Address: 7799 Center Rd
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885559876
E-Mail Address: auctionbuyer001@itransact.com
Customer ID:

Description	Amount	Quantity	Subtotal
eBay Item: 1234ABCD	24.00	1	24.00
eBay Buyer ID: auctionbuyer001	0.00	1	0.00

Transaction Total: 24.00

Sincerely,

ACC Live Test Co.

- Customer Auction Payment Confirmation

Figure 2.90. Customer Auction Payment Confirmation

From: **Credit Card Transaction Processing <orders@itransact.com>**
Subject: Auction Buyer, $24.00 MasterCard XID: 41970136 Approval: TAS678 5/18/2012 17:05:06
Date: May 18, 2012 5:05:07 PM MT 5:05:07 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Auction Buyer <auctionbuyer001@itransact.com>

ACC Live Test Co.,

The following transaction was processed.

CUSTOMER INFORMATION:

Customer Name: Auction Buyer
Address: 7799 Center Rd
City, St. ZIP: BHS, CA 90210
Country: USA
Telephone: 8885559876
E-Mail Address: auctionbuyer001@itransact.com
Customer ID:
Approval Code: TAS678
CVV2 Response:
Card Type: MasterCard
Last Four Digits: 5454
Transaction ID: 41970136
IP Address: 70.102.57.202
AVS Response: N

AVS response descriptions are listed here:
https://secure.itransact.com/support/avs.html

Description	Amount	Quantity	Subtotal
eBay Item: 1234ABCD	24.00	1	24.00
eBay Buyer ID: auctionbuyer001	0.00	1	0.00

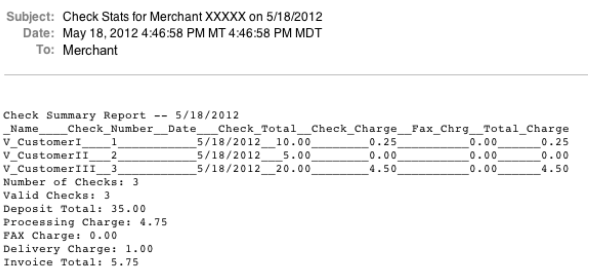
Transaction total: $24.00

Please view my eBay auctions at:
http://cgi6.ebay.com/ws/eBayISAPI.dll?ViewSellersOtherItems&userid=auctionseller001%20

14.14. Check Stats Email

This is sent to people who use the RediCheck printing system.

Figure 2.91. Check Stats Email



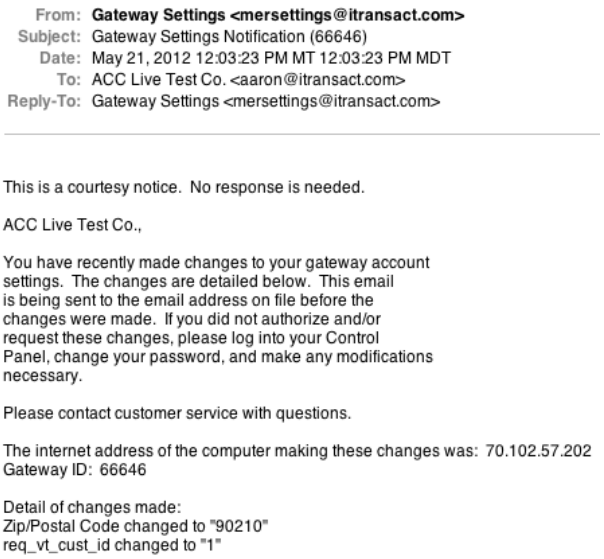
Subject: Check Stats for Merchant XXXXX on 5/18/2012
Date: May 18, 2012 4:46:58 PM MT 4:46:58 PM MDT
To: Merchant

```
Check Summary Report -- 5/18/2012
_Name____Check_Number__Date___Check_Total__Check_Charge__Fax_Chrg__Total_Charge
V_CustomerI____1_____5/18/2012__10.00_____0.25_____0.00_____0.25
V_CustomerII___2_____5/18/2012___5.00_____0.00_____0.00_____0.00
V_CustomerIII__3_____5/18/2012__20.00_____4.50_____0.00_____4.50
Number of Checks: 3
Valid Checks: 3
Deposit Total: 35.00
Processing Charge: 4.75
FAX Charge: 0.00
Delivery Charge: 1.00
Invoice Total: 5.75
```

14.15. Gateway Notification Emails

- Account Settings Change Notification

Figure 2.92. Account Settings Change Notification



From: **Gateway Settings <mersettings@itransact.com>**
Subject: Gateway Settings Notification (66646)
Date: May 21, 2012 12:03:23 PM MT 12:03:23 PM MDT
To: ACC Live Test Co. <aaron@itransact.com>
Reply-To: Gateway Settings <mersettings@itransact.com>

This is a courtesy notice. No response is needed.

ACC Live Test Co.,

You have recently made changes to your gateway account settings. The changes are detailed below. This email is being sent to the email address on file before the changes were made. If you did not authorize and/or request these changes, please log into your Control Panel, change your password, and make any modifications necessary.

Please contact customer service with questions.

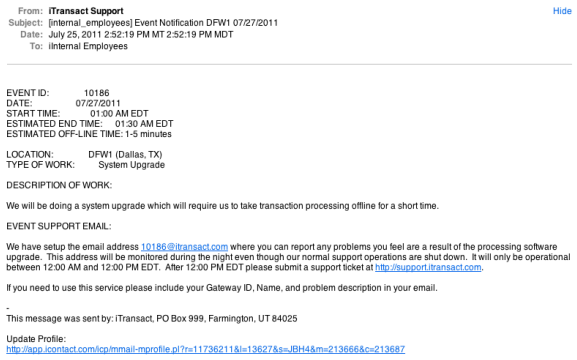
The internet address of the computer making these changes was: 70.102.57.202
Gateway ID: 66646

Detail of changes made:
Zip/Postal Code changed to "90210"
req_vt_cust_id changed to "1"

When any change is made in the Account Settings the gateway will send you an email notifying you of the change. The changes are explained and the IP address of the computer submitting the change is logged for your records. Changes to any of the settings through a session will generate an email.

- MerchantUpdate Email

Figure 2.93. MerchantUpdate Email



From: **iTransact Support** Hide
Subject: [internal_employees] Event Notification DFW1 07/27/2011
Date: July 25, 2011 2:52:19 PM MT 2:52:19 PM MDT
To: iInternal Employees

EVENT ID: 10186
DATE: 07/27/2011
START TIME: 01:00 AM EDT
ESTIMATED END TIME: 01:30 AM EDT
ESTIMATED OFF-LINE TIME: 1-5 minutes

LOCATION: DFW1 (Dallas, TX)
TYPE OF WORK: System Upgrade

DESCRIPTION OF WORK:

We will be doing a system upgrade which will require us to take transaction processing offline for a short time.

EVENT SUPPORT EMAIL:

We have setup the email address 10186@itransact.com where you can report any problems you feel are a result of the processing software upgrade. This address will be monitored during the night even though our normal support operations are shut down. It will only be operational between 12:00 AM and 12:00 PM EDT. After 12:00 PM EDT please submit a support ticket at http://support.itransact.com.

If you need to use this service please include your Gateway ID, Name, and problem description in your email.

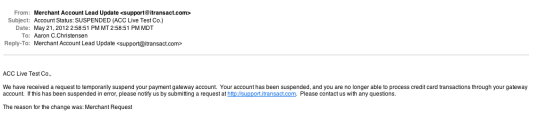
-
This message was sent by: iTransact, PO Box 999, Farmington, UT 84025

Update Profile:
http://app.icontact.com/icp/mmail-mprofile.pl?r=11736211&l=13627&s=JBH4&m=213666&c=213687

When you sign up for the *MerchantUpdates* email list, you will be sent notifications about updates being made to the gateway software, scheduled maintenance, and other system wide information.

- Gateway Status Updates

Figure 2.94. Gateway Status Updates



From: **Merchant Account Lead Update <support@itransact.com>**
Subject: Account Status: SUSPENDED (ACC Live Test Co.)
Date: May 21, 2012 2:58:51 PM MT 2:58:51 PM MDT
To: Aaron C.Christensen
Reply-To: Merchant Account Lead Update <support@itransact.com>

ACC Live Test Co.,

We have received a request to temporarily suspend your payment gateway account. Your account has been suspended, and you are no longer able to process credit card transactions through your gateway account. If this has been suspended in error, please notify us by submitting a request at http://support.itransact.com. Please contact us with any questions.

The reason for the change was: Merchant Request

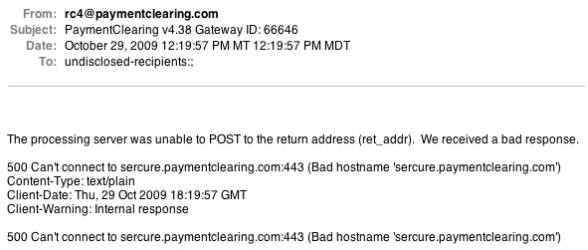
An email like this is sent when there is a change to your account's status. One will be sent if your account is suspended, closed, or re-opened.

14.16. Order Form Error Emails

These emails (sent to your errors email address [22]) notify you when there is an error in your order form coding or with the credentials of the account. These emails are sent from *rc4@itransact.com* or *rc4@paymentclearing.com*. The examples below are the most common form error emails, but other issues could result in an email. Please read the email and contact the *support team* with any questions.

- Return Address POST Error Email

Figure 2.95. Return Address POST Error Email



From: **rc4@paymentclearing.com**
Subject: PaymentClearing v4.38 Gateway ID: 66646
Date: October 29, 2009 12:19:57 PM MT 12:19:57 PM MDT
To: undisclosed-recipients:;

The processing server was unable to POST to the return address (ret_addr). We received a bad response.

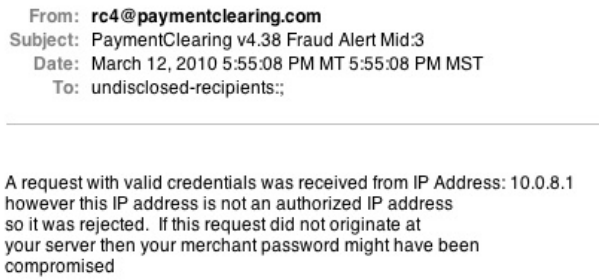
500 Can't connect to sercure.paymentclearing.com:443 (Bad hostname 'sercure.paymentclearing.com')
Content-Type: text/plain
Client-Date: Thu, 29 Oct 2009 18:19:57 GMT
Client-Warning: Internal response

500 Can't connect to sercure.paymentclearing.com:443 (Bad hostname 'sercure.paymentclearing.com')

An email similar to this will be sent if a requested POST of lookup [143] or passback [143] variables is unable to complete the communication back to the ret_addr on your server. The email will generally include a standard server error number (i.e. 403, 404, 500, etc.) that your server administrators can review and fix the issue.

- IP Error Email

Figure 2.96. IP Error Email



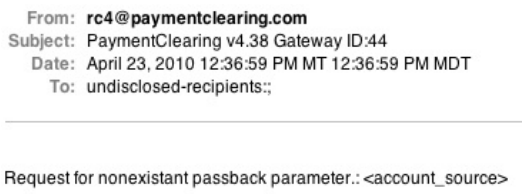
From: **rc4@paymentclearing.com**
Subject: PaymentClearing v4.38 Fraud Alert Mid:3
Date: March 12, 2010 5:55:08 PM MT 5:55:08 PM MST
To: undisclosed-recipients:;

A request with valid credentials was received from IP Address: 10.0.8.1
however this IP address is not an authorized IP address
so it was rejected. If this request did not originate at
your server then your merchant password might have been
compromised

An email similar to this will be sent if a transaction request is submitted from a non-approved IP address through an account with active IP filters [26].

- Nonexistent Passback Error Email

Figure 2.97. Nonexistent Passback Error Email



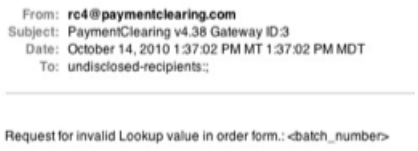
From: **rc4@paymentclearing.com**
Subject: PaymentClearing v4.38 Gateway ID:44
Date: April 23, 2010 12:36:59 PM MT 12:36:59 PM MDT
To: undisclosed-recipients:;

Request for nonexistant passback parameter.: <account_source>

An email similar to this will be sent if a transaction request is submitted asking for a passback [143] for a field value that doesn't exist.

- Invalid Lookup Error Email

Figure 2.98. Invalid Lookup Error Email

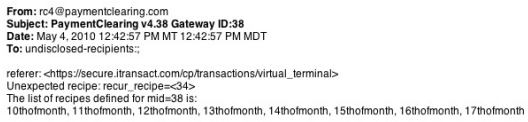

From: rc4@paymentclearing.com
Subject: PaymentClearing v4.38 Gateway ID:3
Date: October 14, 2010 1:37:02 PM MT 1:37:02 PM MDT
To: undisclosed-recipients:;

Request for invalid Lookup value in order form.: <batch_number>

An email similar to this will be sent if a transaction request is submitted asking for *lookup* for a field value that isn't specified as a lookup [143]. If you need that field value returned to you, ask for it as a passback [143].

- Unexpected Recipe Error Email

Figure 2.99. Unexpected Recipe Error Email


From: rc4@paymentclearing.com
Subject: PaymentClearing v4.38 Gateway ID:38
Date: May 4, 2010 12:42:57 PM MT 12:42:57 PM MDT
To: undisclosed-recipients:;

referer: <https://secure.itransact.com/cp/transactions/virtual_terminal>
Unexpected recipe: recur_recipe=<34>
The list of recipes defined for mid=38 is:
10thofmonth, 11thofmonth, 12thofmonth, 13thofmonth, 14thofmonth, 15thofmonth, 16thofmonth, 17thofmonth

An email similar to this will be sent if a recurring transaction request is submitted using a recipe name [61] that has not been setup.

15. Validating a Card

Why is this done?

Many merchants need to validate the card being used for payment before they begin work on preparing and shipping their product. It allows them to check the address verification and CVV information so that they can be assured that the information is verified. Other merchants need to submit card data to the gateway for future recurring transactions without actually charging a card. The gateway allows for this.

Using an AVSOnly

If your credit card processing account supports it, an AVSOnly is the best way to run a validation on a card without charging it. To run an AVSOnly, you can submit a sale transaction with a $0.00 amount. Use any transaction method to accomplish this.

Using a Preauth

If your credit card processing account does not support AVSOnly transactions, use a Preauth for a minimal amount to validate the card instead. Preauths verify cardholder information, but actually puts a freeze on the account for the amount submitted. The charge will not complete unless a Postauth is run. If a Postauth is not run, the authorized amount will drop off in approximately ten days.

16. Phone and Mobile Transactions

The gateway supports multiple methods for accepting transactions. Most merchants utilize the web-based transaction methods. Other merchants use the online methods of the gateway along with non-web based methods. Some of these alternate methods require the use of smart phone, the other method can be accessed by any type of telephone.

16.1. Call-A-Charge

iTransact's proprietary Call-A-Charge service gives you the ability to accept card payments by simply dialing them in by telephone. It is perfect for those occasions when you are away from an Internet connection. These transactions, like all of your other transactions, will show up in the Transaction Listing [32] in your Control Panel. Please continue to check your transactions daily to make sure that everything is accurate. You can continue using the existing features of the Transaction Options [42]to issue voids and credits, to resubmit transactions, etc. Follow these steps to use this service:

1. Dial the access number: 801-951-8260
2. Enter your Gateway ID [21] followed by #
3. Enter your PIN code [21] followed by #
4. Follow the prompts to complete your transaction.

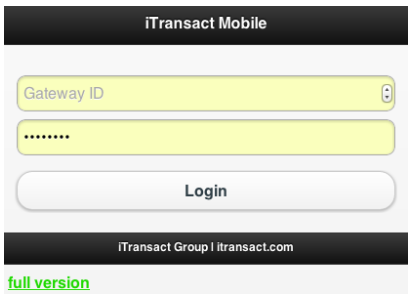
16.2. iTransact Mobile Web Interface

The iTransact Mobile Web interface can be used on any smart phone that has web capabilities. The *mobile interface* is a streamlined version of the iTransact Virtual Terminal with the ability to view a transaction history and generate refunds. It is very simple and intuitive to use.

16.2.1. Instructions

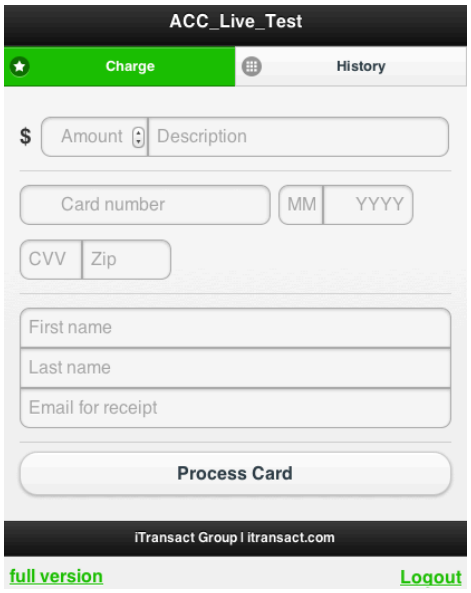
1. Access *http://m.itransact.com* in your web browser. Log in using your Gateway ID [21] and password.

Figure 2.100. Mobile Login Page



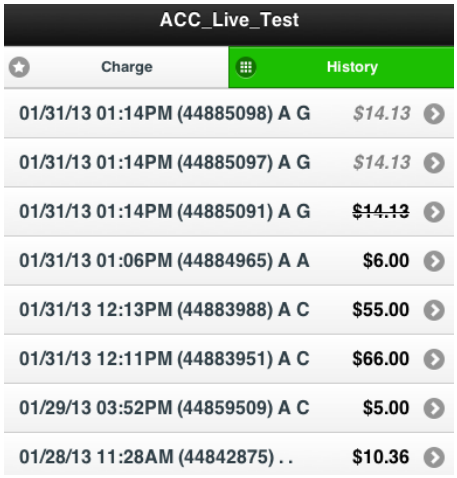
2. To process a sale transaction, complete the form in the Charge window.

Figure 2.101. Mobile Charge Window



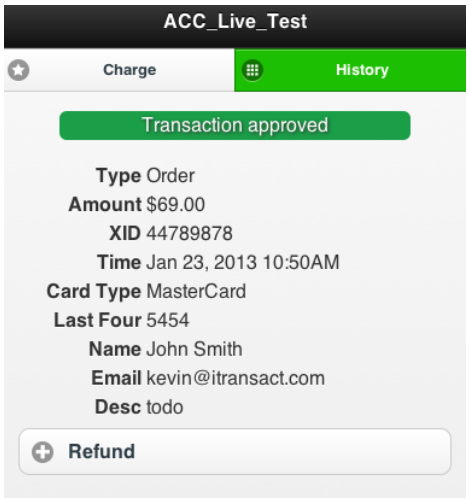
3. All transaction attempts will be listed in the History window, which can be accessed by clicking on the `History` tab. Bold transactions are successful. Grayed transactions are failures. Crossed out amounts indicate that a transaction has been voided or refunded.

Figure 2.102. Mobile History Window



4. To refund or cancel a transaction, click the arrow next to the transaction to be taken to the detail page where a refund can be generated by clicking the `Refund` link and following the displayed instructions.

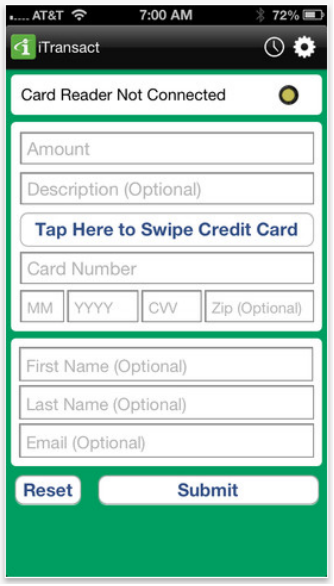
Figure 2.103. Mobile Detail Window



16.3. The iTransact Mobile App

iTransact has developed a mobile app that allows for processing card transactions on the Apple iOS devices and Android devices. The app itself may be used with or without a card reader to accept transactions. The iTransact Mobile card reader dongle plugs directly into the headphone jack of your Apple iOS or Google Android device. Most iOS and many Android devices are compatible with the app. Please view the current list of compatible devices *here*.

Figure 2.104. iTransact Mobile App Example



16.3.1. iTransact Mobile App for Android

Many phones with Google's Android operating system are compatible with out app, so you can use a simple (free) app to run transactions where ever you have a cell or data connection. You can access the *iTransact Mobile* app in the Google Play store directly from your device. To use this app, you must have XML API access enabled on your gateway. To activate XML access, please submit a request to our *support team*. Additionally, you can use the iTransact Mobile swipe dongle device. Please contact our *support team* for information about the dongle.

Many Android devices are compatible. Your device must have:

- Google Android 2.2 or higher (modified versions of the OS are not supported)
- A 3.5mm headset jack with integrated microphone
- Access to Google Play

While the iTransact app itself may work with Android tablets for keyed transactions, the card reader generally will not work because of the limitations of the Android devices.

There are other third-party apps available for Android that are compatible with the iTransact gateway. However, since they are third-party apps, we do not provide support for those apps.

16.3.2. iTransact Mobile App for Apple iOS

Most Apple devices running iOS 5.0 or higher are compatible. You can use a simple (free) *app* to run transactions where ever you have a cell or data connection. You can access the *iTransact Mobile* app in the iTunes store directly from your device. To use this app, you must have XML API access enabled on your gateway. To activate XML access, please submit a request to our support team. Additionally, you can use the iTransact Mobile swipe dongle device. Please contact our *support team* for information about the dongle.

16.3.3. Using the Card Reader Dongle with the iTransact Mobile App

Follow these instructions to use the dongle:

- Open the iTransact Mobile app on your device.
- Place the card reader securely into the headphone jack of your mobile device. The top line of the app window will display `Card Reader Connected`.

- In the app, tap the field that says `Tap Here to Swipe Credit Card`.
- Run the card through the reader with the magnetic stripe inside the reader and against the reader's thick side.
- Swipe in a steady motion at a normal pace. You can swipe either direction, but make sure to keep the card straight all the way through the reader so that it reads the entire magnetic stripe.
- You will know the swipe was successful when the app displays the card number and expiration date.
- Complete the remainder of the fields and press `Submit`.
- If you encounter errors swiping the card, you can manually complete all fields.

16.3.4. Using the iTransact Mobile App to Process a Refund

Within the iTransact Mobile app, tap the `History` icon (the icon in the top right corner that looks like a clock). This displays your device's transaction history.

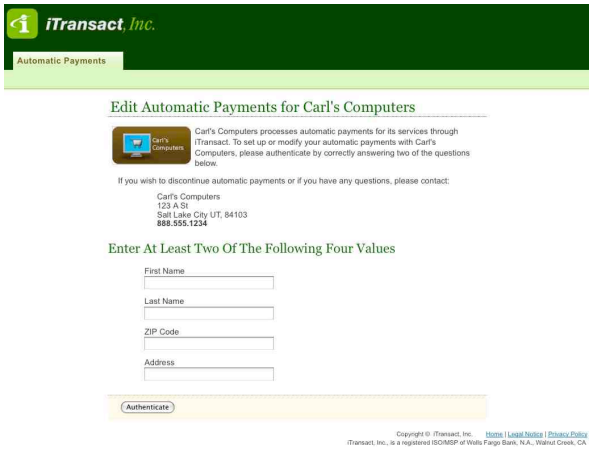
- Select the transaction you wish to refund.
- Select the transaction you wish to refund.
- Confirm the request by tapping `OK`.
- In your device's transaction history, the refund will show as a credit for the refunded amount.

17. Customer Recurring Update

Merchants can opt to let their customers update their own credit card billing information for recurring charges. The tool is comprised of several features that make this possible. This is only available on recurring recipes that have been enabled for Allow Customer Update [63] - and can ONLY be used for credit card transactions. This can be enabled for new and existing recipes. When a recurring transaction is successful or fails, an email including a link for the secure update interface will be generated to the cardholder. Following the simple instructions, they can update their own billing information. If the update was necessary because a recurring attempt had failed, a transaction will be attempted to bring the missed recurring billing up-to-date.

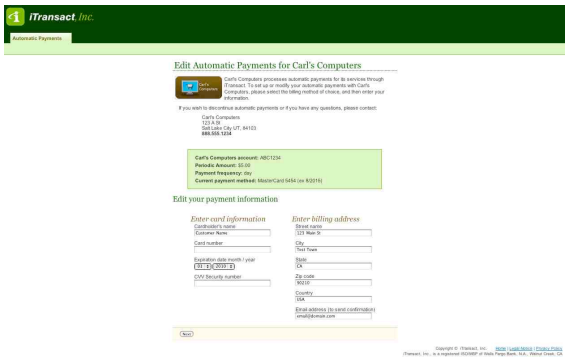
Most merchants choose to use this feature in conjunction with the Hold on Failure recurring function - so that when a transaction fails, the recurring is put on hold (to prevent future attempts), and the update link is sent out. A merchant can also resend confirmation emails including the link to their customers from the Transaction Options [42] area of the Transaction Listing. When a customer follows the link, they will be taken to this page:

Figure 2.105. Customer Recurring Update - Credential Window



This verification page is a feature used to verify that the person accessing the tool is using the correct information for a specific transaction. If the customer enters valid verification information, they will be taken to the following page where they can update their information:

Figure 2.106. Customer Recurring Update - Update Page



17.1. Update Tool Details

The Allow Customer Update [63] feature makes it possible for a merchant to allow cardholders to update their own recurring billing information for credit card transactions attached to recipes where the update option is enabled. The Recurring On-Hold feature provides the ability to put future recurring attempts "on hold" temporarily without removing the value of the remaining repetitions. Pro-active cardholders will also be able to update their own card information without requiring the merchant to have to manually make the changes themselves. Merchants will also be able to temporarily stop specific recurring transactions from attempting.

When this feature is in use, a declined or failed recurring transaction will generate an email to the cardholder instructing them to click a link which will take them to a secure web page where they will be able to update their billing information. The recurring customer confirmation emails for successful transactions will also include a link for updating card information if they want to update the account information that is being used. Additionally, the merchant will be notified of the update link via the billing_update_token in the Recurring Postback, so that if they'd like to be able to provide the customer with the update access through their own system, they'll have the correct security token.

The Recurring On-Hold feature allows a merchant to toggle recurring on and off for specific transactions, without changing the number of remaining repetitions for that account. This can be used to prevent future attempts against a recurring transaction temporarily. If the Hold on Failure function of the Recurring On-Hold feature is enabled in a recipe, a failed recurring credit card billing will place a transaction on-hold to prevent future billing attempts so that account information can be updated (either by the merchant or the customer). When either of these features are triggered by a failed transaction, the merchant will be notified by email in the Recurring Failure email and in the Recurring Postback (using the billing_update_token and on_hold parameters). A value for the billing_update_token designating the update page will be available on a successful recur-initiating transaction for recipes where Allow Customer Update is enabled.

In addition to the update link being displayed in the email, you can opt to display the update link on somewhere on your website so that a specific user can link to their update page. The value of any valid billing_update_token can be appended to

https://secure.itransact.com/customers/billing_update/

For example, a link using a token with a value of "abc123" would be

https://secure.itransact.com/customers/billing_update/abc123

When an update is made to a transaction that is not on hold, an AVSOnly or Preauth transaction is attempted to validate the updated information.

17.2. Activating These Features

Recurring On-Hold is available for any recurring transaction. This can be toggled in the Transaction Details for a specific transaction or through the Recurring History for a specific transaction in the Recurring Transactions

interface. Those interfaces display the On-Hold toggle field with a link to toggle on (`Yes`) or off (`No`) the hold setting.

Allow Customer Update and Hold on Failure can be enabled when a recurring recipe is created in the Recipe Builder [61] by selecting the checkboxes. These features can be enabled independent of each other. If you would like to activate this for existing recipes, you may edit your recipes by logging into your Control Panel, accessing the Recurring Transaction area, and using the edit feature for the necessary recipes. You will see the checkboxes that you can enable. Activate the desired features and click the `Save Recipe` button.

18. What Does the Error Mean?

18.1. Link to Errors and Failures

The gateway system uses emails as the primary source of communication. Many different types of emails are generated for different purposes. Settings for some of the merchant [22] and customer [24] emails are available in the Account Settings. Emails for a failed transaction will include the error verbiage sent back from the processor. A comprehensive list of potential errors may be reviewed here [178].

18.2. General Guidelines

- **For Credit Cards**

  For any error that mentions “decline”, know that you will need to request a different card to complete the transaction. There are many reasons that a card can be declined. Unfortunately, most issuing banks don't explain the specific reason for a decline. Trust that a decline is a valid reason to ask for a different card to attempt. If you are receiving an error that indicates that your merchant account is invalid, it usually means that your merchant account has been shut down or closed. Please check with your credit card merchant account provider.

- **For EFTs**

  The EFT processor verifies account data against several databases before they will allow a transaction to process. Most errors for EFTs are linked to bad verification information or a limited credit history for the account.

Chapter 3. Developer Information

1. Which Connection Method is for Me?

iTransact offers multiple methods for integrating your site or app with the gateway. You can communicate with the gateway via an HTML [122] form post or via an XML [90] query request. Please use this section to help determine which method suits your needs best.

1.1. XML API - Preferred Method

This type of integration gives the gives you complete control of the transaction process, since requests and responses are handled within the same HTTPS connection. The use of XML allows developers to create their own Windows COM objects, Java apps, PHP routines, Perl libraries, standardized Web Services, etc. If an application can generate XML, it can process transactions. Since the XML method [90] is simply another method for processing transactions, all gateway features remain available.

1.2. HTML Methods

The HTML Connection method [122]utilizes an HTTPS form post to pass information securely from your order form to our transaction servers. This method can be easily integrated into any website. This method can be used by merchants who have their own secure servers and by merchants who do not have secure servers. Merchants who do not have a secure server should use our BuyNow [135] or Split Form [129]-based protocol and merchants who have a secure server should use our Standard Form [123]-based protocol. The Form Creation Wizard within the *Control Panel* may be used to create an HTML order form for the merchant's site, which can be modified as needed. Two advanced features available with the HTML methods are the Lookup [143] and Passback [143] functions, which will send transaction data to a dynamic script on your server via an HTTPS post.

1.3. Deprecated XML Method

You should not use this. This is a deprecated method. Information in included here [171] for the reference of users still utilizing this older version. Future versions of this documentation may not include information regarding this method.

1.4. Compatible Shopping Carts

Developers are constantly building modules that integrate the gateway system with various shopping carts. There is a list that is being added to often. To receive access to the latest list, please submit a request through our *support ticket system*.

If the cart you would like to use is not currently compatible, please speak to the cart provider to determine if they could add support for our gateway.

2. XML API Connection Method

The XML API connection method allows submission of XML transaction requests using API key credentials. With one set of credentials, a merchant can submit transactions through any of the gateways that they own. This feature removes the necessity to have to track multiple sets of credentials and IP filters.

Authentication is handled using the combination of a username and a HMAC digest. A HMAC digest verifies both the authenticity and data integrity for a request payload. Our implementation is based on HMAC-SHA-1 using a 160 bit key. You can read more about HMAC at Wikipedia [*http://en.wikipedia.org/wiki/HMAC*] or for the adventurous the RFC [*http://www.faqs.org/rfcs/rfc2104.html*] is available. By using a signature based authentication scheme we can validate who generated an API request and ensure that the request was not altered using a man-in-the-middle attack [*http://en.wikipedia.org/wiki/Man-in-the-middle_attack*]. To access the reporting tool you will

sign a string containing all the request parameters you are posting to the report and then pass that in along with your API username.

2.1. Obtaining your API Key

The reseller and/or merchant interfaces can be used to enable API access and setup and revoke API credentials. Resellers should click on the *API Access* tab and follow the instructions listed in the *Reseller Interface*. Merchants can setup their API features [23] using the Account Settings tool in the Control Panel.

2.2. Payload Signature Generation

All the XML features require a signature value to be calculated and passed through in the request. This is part of the authentication process and ensures requests cannot be modified after leaving your servers. Each XML interface will require this in a different input field so this chapter is just providing a general discussion of how to generate the signature for a given payload value. The payload format is also a feature of the XML interface being used, so refer to the XML interface specific sections discussing what the payload is.

2.2.1. Generating the PayloadSignature Value

Every module that supports the API username/key authentication requires that some part of the request method is signed and sent as a separate parameter of the request. Once the request is received by our servers, we create the same signature from the request and compare the values. This type of authentication is used by many other companies and the hash functions are free and widely available. Many languages come with these hash functions as standard libraries. We have included examples for many popular programming languages below. The examples all use the same payload to create the hash. That payload is

<RecurDetails><OperationXID>12345</OperationXID></RecurDetails>

Remember, each transaction will have a unique payload. These examples are simply for testing. Do not try to use this payload or the generated PayloadSignature for your real transactions. It will not work.

Figure 3.1. Java HMAC Signature Example

```
import java.util.*;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.binary.Base64;
public class HMACSignature {
  private static final String EXPECTED_SIGNATURE = "4U1OXfCzwOG1IMWHOgalplUcWiE=";
  public static void main(String[] args) throws Exception {
    java.security.Security.addProvider(new com.sun.crypto.provider.SunJCE());
   SecretKeySpec hmac = new SecretKeySpec( "12345678901234567890".getBytes("ASCII"),
     "HmacSHA1" );
    Mac mac = Mac.getInstance( hmac.getAlgorithm() );
    mac.init( hmac );
    String sigLoad = "<RecurDetails><OperationXID>12345</OperationXID></RecurDetails
    byte[] digest = mac.doFinal(sigLoad.getBytes("UTF8"));
     String actualSignature = new String(Base64.encodeBase64(digest), "ASCII");
    if (EXPECTED_SIGNATURE.equals(actualSignature)) {
      System.out.println("Success!");
      System.out.println("Signature: " + actualSignature);
    }
    else {
      System.out.println("Failure!");
      System.out.println("Expected Signature: " + EXPECTED_SIGNATURE);
      System.out.println("  Actual Signature: " + actualSignature);
     }
   }
}
```

Figure 3.2. Perl HMAC Signature Example

```
my $EXPECTED_SIGNATURE = '4U1OXfCzwOG1lMWHOgaIplUcWiE=';
my $payload = "<RecurDetails><OperationXID>12345</OperationXID></RecurDetails>";
use Digest::HMAC_SHA1;
my $hmac = Digest::HMAC_SHA1->new("12345678901234567890");
$hmac->add($payload);
# The Perl Digest lib doesn't provide the trailing '=' character
my $actual_signature = $hmac->b64digest . '=';
if ($EXPECTED_SIGNATURE eq $actual_signature) {
  print "Success!\n";
  print "Signature: '$actual_signature'\n";
} else {
  print "Failure!\n";
  print "Expected Signature: '$EXPECTED_SIGNATURE'\n";
print "  Actual Signature: '$actual_signature'\n";
}
```

Figure 3.3. Ruby HMAC Signature Example

```
require 'openssl'
require 'base64'
EXPECTED_SIGNATURE = '4U1OXfCzwOG1lMWHOgaIplUcWiE='
payload = "<RecurDetails><OperationXID>12345</OperationXID></RecurDetails>"
key = "12345678901234567890"
digest=OpenSSL::HMAC.digest(OpenSSL::Digest::SHA1.new(key), key, payload)
actual_signature = Base64.b64encode(digest)
# For some reason, we end up with a new line character in the actual_signature...
actual_signature.chomp!
if EXPECTED_SIGNATURE.eql?(actual_signature)
  puts 'Success!'
  puts 'Signature: ' + actual_signature
else
  puts 'Failure!'
  puts 'Expected Signature: ' + EXPECTED_SIGNATURE
  puts '  Actual Signature: ' + actual_signature
end
```

Figure 3.4. ColdFusion HMAC Signature Example

```
<cfset expectedSignature = "4U1OXfCzwOG1lMWHOgaIplUcWiE=">
<cfset payload = "<RecurDetails><OperationXID>12345</OperationXID></RecurDetails>">
<cfset key = "12345678901234567890">
<cfset generatedSignature = HMAC_SHA1(key = key, payload = payload)>
<cfoutput>
Payload:<br>
#payload#
<hr>
Expected Signature: #expectedSignature#<br>
Generated Signature: #generatedSignature#
</cfoutput>
```

Figure 3.5. PHP HMAC Signature Example

```
<?php
$expected = '4U1OXfCzwOG1lMWHOgaIplUcWiE=';
$payload = "<RecurDetails><OperationXID>12345</OperationXID></RecurDetails>";
$key = "12345678901234567890";
#Using built in PHP5 functions
$digest = hash_hmac('sha1', $payload, $key, true);
$actual_signature = base64_encode($digest);
#Using PEAR module
#require_once 'Crypt/HMAC.php';
#$hmac = new Crypt_HMAC($key,"sha1");
#$digest = pack("H40", $hmac->hash(trim($payload)));
#$actual_signature = base64_encode($digest);
if($expected == $actual_signature) {
  echo "Success!\n";
  echo "Signature: $actual_signature\n";
} else {
  echo "Failure!\n";
  echo "Expected Signature: $expected\n";
  echo "  Actual Signature: $actual_signature\n";
} ?>
```

Figure 3.6. .NET/C# HMAC Signature Example

```
using System;
using System.Security.Cryptography;
using Text;
string expected_signature = "4U1OXfCzwOG1lMWHOgaIplUcWiE=";
string payload = "<RecurDetails><OperationXID>12345</OperationXID></RecurDetails>";
string key = "12345678901234567890";
Encoding encoding = new UTF8Encoding();
HMACSHA1 signature = new HMACSHA1(encoding.GetBytes(key));
string actual_signature = Convert.ToBase64String(
  signature.ComputeHash(encoding.GetBytes(
    payload.ToCharArray() ) )
);
if(expected_signature == actual_signature) {
  Console.WriteLine("Success!");
  Console.WriteLine("  Signature:  {0}", actual_signature);
} else {
  Console.WriteLine("Failure!");
  Console.WriteLine("Expected Signature:  {0}", expected_signature);
  Console.WriteLine("  Actual Signature:  {0}", actual_signature);
}
```

2.2.2. Testing Your Signature Process

If you would like to test your signature process without having to hit our servers use the signatures generated in the examples above as a test base. If you use the same set of parameters, your code should generate the same signature values that are shown in the examples.

2.3. The XMLTrans2.cgi Module

The use of this module allows a reseller or merchant to submit Check and Card transactions as well as other requests such as recurring transaction modification. An API request is made up of the XML and Interface declarations, the APICredentials element set (including PayloadSignature) and the action element set. Use this as the address to submit your API calls to:

https://secure.itransact.com/cgi-bin/rc/xmltrans2.cgi

2.3.1. MIME Type Information

The cgi is accessed with an HTTP POST and requires a CONTENT_TYPE header to be specified. Either "application/x-www-form-urlencoded" or "text/xml" must be used. If "application/ x-www-form-urlencoded" is sent, then the HTTP body must contain valid form markup. See WW3 Form Spec for details at *w3.org*. If "text/xml" is used then the HTTP body should only contain the XML request. If the incorrect MIME type is used, the following response will be sent back to your server:

Figure 3.7. MIME Type Error Example

```
<?xml version="1.0" standalone="yes"?>
 <GatewayFailureResponse>
  <Status>FAILED</Status>
   <ErrorCategory>REQUEST_FORMAT</ErrorCategory>
   <ErrorMessage>Unexpected mime type: </ErrorMessage>
 </GatewayFailureResponse>
```

2.3.2. Tools for Testing the XMLTrans2 Module

You can run all requests through xmltrans2 in test mode in order to help with the integration process. However, there are certain types of requests which require editing or using transaction data that exists in the gateway database - and those types of requests may be better tested by using an account in our development environment. If you are interested in that type of access, please submit a request to ticket.liaison@itransact.com.

For a standard, production environment account, there are three different ways to run a XML transaction in test mode:

1. Enable the `Test Mode` checkbox in your Account Settings [19]. Please remember this turns on Test Mode for all transactions.

2. Setup the `Test User First Name` value in your merchant Account Settings [19]. Any xmltrans2 request that you submit with a BillingAddress FirstName tag with this same value will be run as a test transaction. This is only available on some transaction requests such as CreditTransaction and AuthTransaction.

3. Send through a TestMode tag with the value "TRUE". The TestMode tag which is in the TransactionControl structure always overrides the other test settings. This means that sending through TestMode equal to "FALSE" will cause a transaction to always be run as a live transaction (circumventing any other settings).

2.3.3. API Payloads

The API Payload for which a signature is generated is simply the whole XML structure of the "action" in the GatewayInterface (or iTransactInterface) element. For example, the payload section to sign for the following request is highlighted in **bold**:

```
<?xml version="1.0" standalone="yes"?>
<GatewayInterface>
  <APICredentials>
    <Username>username</Username>
    <PayloadSignature>signature</PayloadSignature>
    <TargetGateway>12345</TargetGateway>
  </APICredentials>
  <RecurDetails>
    <OperationXID>12345</OperationXID>
  </RecurDetails>
</GatewayInterface>
```

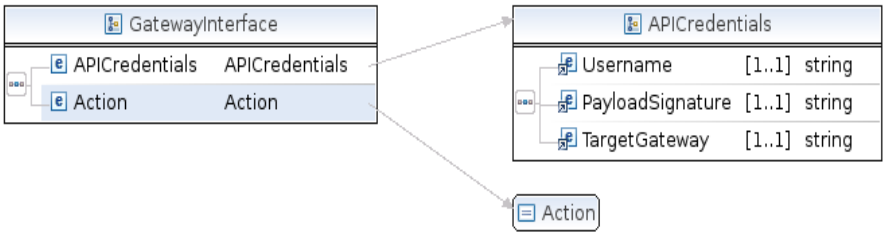
The example shows a request with the RecurDetails “action”. For any type of request, the payload will be the entire “action” element set. The payload passed into your signature generation routines should start with the first < character in the “action” opening element tag and should end with the last > in the closing action element tag. We do not strip out space or carriage return/newline characters when validating the signature on our end. If you are having problems retrieving the XML sub-structure without affecting the spacing that is present compared to how we receive it, you might want to disable XML pretty printing. This will ensure that there are no whitespace

characters in your payload. Use the element section and calculate the value for PayloadSignature - a unique value for each transaction.

2.4. Requests

An API request is made up of the XML and Interface declarations, the APICredentials element set (including PayloadSignature) and the action element set.

Figure 3.8. GatewayInterface Diagram



2.4.1. Credentials

The APICredentials set are made up of the following elements:

- **Username** - This value is found (and can be reset) in the Account Settings [19] of the Control Panel on an API enabled account.
- **PayloadSignature** - This is generated by signing the unique payload of each transaction with the API key, which is found (and can be reset) in the Account Settings [19] of the Control Panel on an API enabled account.
- **TargetGateway** - This is the five digit gateway ID number of the account. This field is designed for use with the iTransact Reseller API, but it is an optional field for the standard Gateway API.

2.4.2. Actions

This section of the request will determine which type of request is generated.

Table 3.1. Actions

Action	Response	Description
AuthTransaction [96]	*TransactionResponse*	Performs a force, preauth, or sale request
CreditTransaction [99]	*TransactionResponse*	Performs a credit transaction without referencing a previous transaction
PostAuthTransaction [100]	*TransactionResponse*	Generates a postauth capture for a previously authorized transaction
RecurDetails [109]	*RecurDetailsResponse*	Returns the number of remaining repetitions, recipe name, and total for a recurring transaction
RecurUpdate [107]	*RecurUpdateResponse*	Modifies billing information, customer information, and/or recurring status for an existing recurring transaction
TranCredTransaction [102]	*TransactionResponse*	Performs a credit transaction referencing billing information from a previous transaction

Action	Response	Description
TranForceTransaction [103]	*TransactionResponse*	Performs a force transaction using the billing information from a previous transaction
TranRefundTransaction [106]	*TransactionResponse*	Performs a force transaction using the billing information from a previous transaction
TranRetryTransaction [104]	*TransactionResponse*	Performs a sale using the billing information from a previous transaction
TransactionStatus [109]	*TransactionResponse*	Returns transaction information based on VendorData elements which can be used to determine the status of an attempt
VoidTransaction [105]	*TransactionResponse*	Voids an auth transaction in an open batch. If batch is already settled, use TranCredTransaction [102]

2.4.2.1. AuthTransaction

The AuthTransaction can be used to perform four different authorization requests. A sale request is the default authorization type performed which is an authorization that will automatically be captured during the settlement process. An AVSOnly credit card transaction request is a sale request run with a zero amount and can be used to validate the AVS and CVV information on a card without actually running an authorization. A pre-auth request is performed when the PreAuth field is included. This performs an authorization that will not be captured during the settlement process until a post-auth transaction is run. A force transaction is run when an AuthCode field is included in the request. A force transaction can be used when you have been provided an authorization code over the phone by a processing network.

Figure 3.9. AuthTransaction Diagram

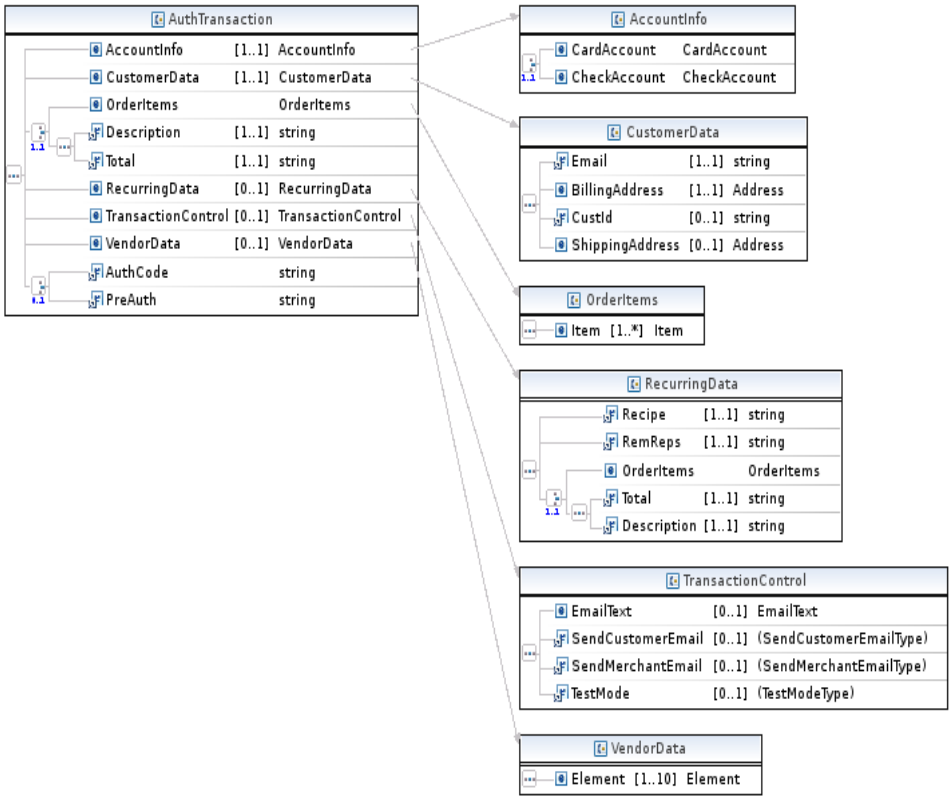


Table 3.2. AuthTransaction Elements

Element	Description	Data Type	Required
AccountInfo	Contains either card or check information	pc:AccountInfo	Yes
AuthCode	Initiates forced transaction	CustomerData	No (2)
CustomerData	Customer Information	pc:CustomerData	Yes
Description	Transaction Description	xs:string	Yes (1)
OrderItems	Transaction Line Items	pc:OrderItems	Yes (1)
Preauth	Initiates a preauthorization	xs:string	No (2)
RecurringData	Recurring transaction information	pc:RecurringDetail	No
Total	Total transaction amount	xs:string	Yes (1)
TransactionControl	Transaction controls	pc:TransactionControl	No
VendorData	Merchant supplied metadata	pc:VendorData	No

(1) *Either OrderItems or Total and Description must be provided.*

(2) *PreAuth and AuthCode can not be used in the same request.*

For a full explanation of everything that makes up those elements, please see the schema details [114].

AuthTransaction Request Example

```
<?xml version="1.0"?>
<GatewayInterface>
  <APICredentials>
    <Username>username</Username>
    <PayloadSignature>signature</PayloadSignature>
    <TargetGateway>12345</TargetGateway>
  </APICredentials>
  <AuthTransaction>
<!-- Optional. Supplying AuthCode results in Force transaction -->
  <AuthCode>12345</AuthCode>
<!-- Optional. Supplying Preauth results in Pre-Auth transaction -->
  <Preauth/>
    <CustomerData>
    <Email>test@example.com</Email>
    <BillingAddress>
      <Address1>test</Address1>
      <Address2>test2</Address2>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
      <City>Bountiful</City>
      <State>UT</State>
      <Zip>84032</Zip>
      <Country>USA</Country>
      <Phone>801-555-1212</Phone>
    </BillingAddress>
<!-- Optional ShippingAddress -->
    <ShippingAddress>
      <Address1>test</Address1>
      <Address2>test2</Address2>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
      <City>Bountiful</City>
      <State>UT</State>
      <Zip>84032</Zip>
      <Country>USA</Country>
      <Phone>801-555-1212</Phone>
    </ShippingAddress>
```

```
<!-- Optional Customer ID -->
    <CustId>12345</CustId>
  </CustomerData>
<!-- Can either supply OrderItems or Total and Description -->
  <OrderItems>
    <Item>
      <Description>test</Description>
      <Cost>10.00</Cost>
      <Qty>1</Qty>
    </Item>
  </OrderItems>
  <Total>10.00</Total>
  <Description>desc</Description>
  <AccountInfo>
<!-- Can supply either CardAccount or CheckAccount -->
    <CardAccount>
<!-- Supply AccountNumber, ExpirationMonth and ExpirationYear or TrackData -->
      <AccountNumber>5454545454545454</AccountNumber>
      <ExpirationMonth>01</ExpirationMonth>
      <ExpirationYear>2000</ExpirationYear>
<!-- Optional CVV code -->
      <CVVNumber>123</CVVNumber>
<!-- Optional Track Data if running swipe transaction -->
      <TrackData>TRACK DATA</TrackData>
<!-- Supply Ksn, Pin along with TrackData for Debit transactions -->
      <Ksn>12345</Ksn>
      <Pin>1234</Pin>
    </CardAccount>
    <CheckAccount>
      <AccountNumber>123456</AccountNumber>
      <ABA>324377516</ABA>
<!-- SecCode if required by processor -->
      <SecCode>PPD</SecCode>
<!-- AccountSource if required by processor. Can be "checking" or "savings” -->
      <AccountSource>checking</AccountSource>
<!-- AccountType if required by processor. Can be "personal" or "business” -->
      <AccountType>personal</AccountType>
    </CheckAccount>
  </AccountInfo>
<!-- Optional recurring commands -->
  <RecurringData>
    <Recipe>text</Recipe>
    <RemReps>1</RemReps>
<!-- Optional - If used, supply OrderItems or Total and Description -->
    <OrderItems>
      <Item>
        <Description>test</Description>
        <Cost>10.00</Cost>
        <Qty>1</Qty>
      </Item>
    </OrderItems>
    <Total>10.00</Total>
    <Description>desc</Description>
    </RecurringData>
<!-- Optional transaction commands -->
  <TransactionControl>
    <SendCustomerEmail>TRUE</SendCustomerEmail> <!-- TRUE/FALSE -->
    <SendMerchantEmail>TRUE</SendMerchantEmail> <!-- TRUE/FALSE -->
    <TestMode>TRUE</TestMode> <!-- TRUE/FALSE -->
    <EmailText>
<!-- Supports up to 10 EmailTextItem fields -->
      <EmailTextItem>test1</EmailTextItem> </EmailText>
  </TransactionControl>
<!-- Optional - This is saved and is available in the XML transaction report.-->
  <VendorData>
    <Element>
      <Name>repId</Name>
      <Value>1234567</Value>
    </Element>
  </VendorData>
```

```
  </AuthTransaction>
</GatewayInterface>
```

This request will receive a TransactionResponse [110].

2.4.2.2. CreditTransaction

Generate a credit/refund transaction for a transaction that was not originally processed through the gateway. This works for both credit cards and EFTs.

Figure 3.10. CreditTransaction Diagram

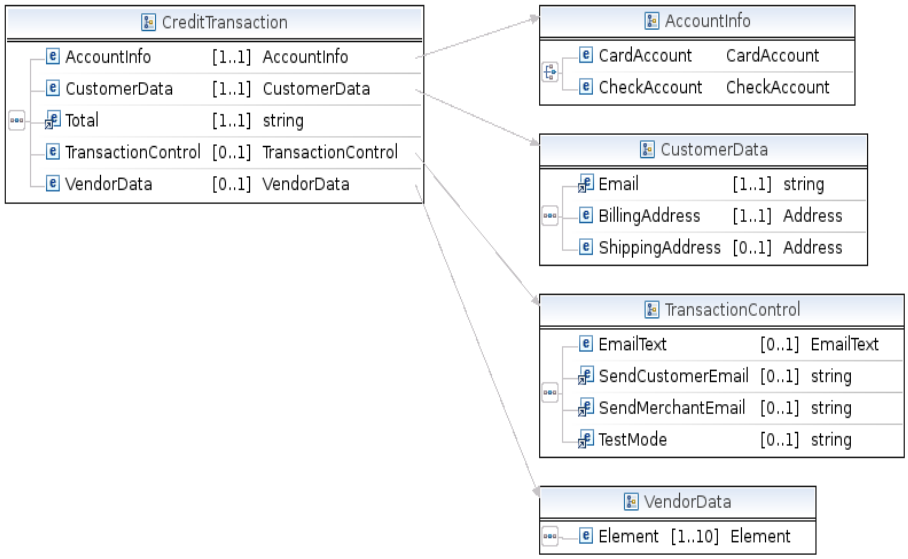


Table 3.3. CreditTransaction Elements

Element	Description	Data Type	Required
AccountInfo	Contains either card or check information	pc:AccountInfo	Yes
CustomerData	Customer Information	pc:CustomerData	Yes
Total	Total transaction amount	xs:string	Yes
TransactionControl	Transaction controls	pc:TransactionControl	No
VendorData	Merchant supplied meta-data	pc:VendorData	No

For a full explanation of everything that makes up those elements, please see the schema details [114].

CreditTransaction Request Example

```
<?xml version="1.0"?>
<GatewayInterface>
  <APICredentials>
    <Username>username</Username>
    <PayloadSignature>signature</PayloadSignature>
    <TargetGateway>12345</TargetGateway>
  </APICredentials>
  <CreditTransaction>
    <Total>5.00</Total>
    <CustomerData>
      <Email>demo@demo.com</Email>
<!-- Optional custom id -->
```

```
      <CustId>12345</CustId>
      <BillingAddress>
        <Address1>test</Address1>
        <FirstName>John</FirstName>
        <LastName>Smith</LastName>
        <City>Bountiful</City>
        <State>UT</State>
        <Zip>84032</Zip>
        <Country>USA</Country>
        <Phone>801-555-1212</Phone>
      </BillingAddress>
<!-- Optional shipping-->
      <ShippingAddress>
        <Address1>test</Address1>
        <FirstName>John</FirstName>
        <LastName>Smith</LastName>
        <City>Bountiful</City>
        <State>UT</State>
        <Zip>84032</Zip>
        <Country>USA</Country>
      </ShippingAddress>
    </CustomerData>
    <AccountInfo>
<!-- Use for Credit card transaction. -->
      <CardAccount>
        <AccountNumber>5454545454545454</AccountNumber>
        <ExpirationMonth>01</ExpirationMonth>
        <ExpirationYear>2000</ExpirationYear>
      </CardAccount>
<!-- For EFT transactions. -->
      <CheckAccount>
        <AccountNumber>123456</AccountNumber>
        <ABA>324377516</ABA>
      </CheckAccount>
<!-- For NACHA transactions. -->
      <CheckAccount>
        <AccountNumber>123456</AccountNumber>
        <ABA>324377516</ABA>
<!-- AccountSource if required by processor. Can be "checking" or "savings" -->
        <AccountSource>checking</AccountSource>
      </CheckAccount>
    </AccountInfo>
<!-- All TransactionControl elements are optional -->
    <TransactionControl>
      <SendCustomerEmail>TRUE</SendCustomerEmail> <!-- TRUE/FALSE -->
      <SendMerchantEmail>TRUE</SendMerchantEmail> <!-- TRUE/FALSE -->
      <TestMode>TRUE</TestMode> <!-- TRUE/FALSE -->
      <EmailText>
<!-- Up to 10 EmailTextItem elements allowed -->
        <EmailTextItem>test1</EmailTextItem>
      </EmailText>
    </TransactionControl>
<!-- Optional - This is saved and is available in the XML transaction report.-->
    <VendorData>
      <Element>
        <Name>repId</Name>
        <Value>1234567</Value>
      </Element>
    </VendorData>
  </CreditTransaction>
</GatewayInterface>
```

This request will receive a TransactionResponse [110].

2.4.2.3. PostAuthTransaction

This will generate a postauth (capture) for a previously run pre-auth transaction. The OperationXID field should contain the XID for the original preauth.

Figure 3.11. PostAuthTransaction Diagram

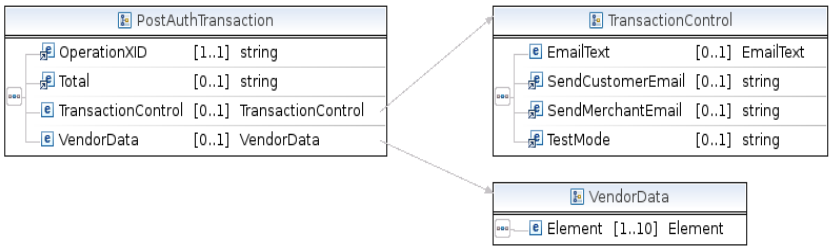


Table 3.4. PostAuthTransaction Elements

Element	Description	Data Type	Required
OperationXID	XID of preauth transaction	xs:string	Yes
Total	Total transaction amount if different from preauth amount	xs:string	No
TransactionControl	Transaction controls	pc:TransactionControl	No
VendorData	Merchant supplied meta-data	pc:VendorData	No

For a full explanation of everything that makes up those elements, please see the schema details [114].

PostAuthTransaction Request Example

```
<?xml version="1.0"?>
<GatewayInterface>
  <APICredentials>
    <Username>username</Username>
    <PayloadSignature>signature</PayloadSignature>
    <TargetGateway>12345</TargetGateway>
  </APICredentials>
  <PostAuthTransaction>
    <OperationXID>3535353</OperationXID>
<!-- Optional - Will use original transaction amount if not specified in Total -->
    <Total>5.00</Total>
<!-- All TransactionControl elements are optional including TransactionControl -->
    <TransactionControl>
      <SendCustomerEmail>TRUE</SendCustomerEmail> <!-- TRUE/FALSE -->
      <SendMerchantEmail>TRUE</SendMerchantEmail> <!-- TRUE/FALSE -->
      <TestMode>TRUE</TestMode> <!-- TRUE/FALSE -->
      <EmailText>
<!-- Up to 10 EmailTextItem elements allowed -->
        <EmailTextItem>test1</EmailTextItem>
        <EmailTextItem>test2</EmailTextItem>
        <EmailTextItem>test3</EmailTextItem>
        <EmailTextItem>test4</EmailTextItem>
        <EmailTextItem>test5</EmailTextItem>
        <EmailTextItem>test6</EmailTextItem>
        <EmailTextItem>test7</EmailTextItem>
        <EmailTextItem>test8</EmailTextItem>
        <EmailTextItem>test9</EmailTextItem>
        <EmailTextItem>test10</EmailTextItem>
      </EmailText>
    </TransactionControl>
<!-- Optional - This is saved is available in the XML transaction report. -->
    <VendorData>
      <Element>
        <Name>repId</Name>
        <Value>1234567</Value>
      </Element>
    </VendorData>
  </PostAuthTransaction>
```

```
</GatewayInterface>
```

This request will receive a TransactionResponse [110].

2.4.2.4. TranCredTransaction

This will generate a refund for a previously run transaction. The OperationXID field should contain the XID for the original transaction.

Figure 3.12. TranCredTransaction Diagram

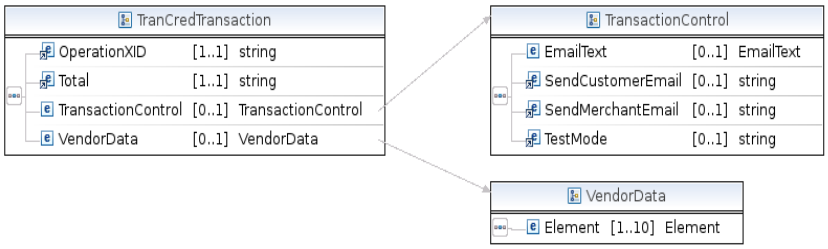


Table 3.5. TranCredTransaction Elements

Element	Description	Data Type	Required
OperationXID	XID of preauth transaction	xs:string	Yes
Total	Total transaction amount if different from preauth amount	xs:string	Yes
TransactionControl	Transaction controls	pc:TransactionControl	No
VendorData	Merchant supplied meta-data	pc:VendorData	No

For a full explanation of everything that makes up those elements, please see the schema details [114].

TranCredTransaction Request Example

```
<?xml version="1.0"?>
<GatewayInterface>
  <APICredentials>
    <Username>username</Username>
    <PayloadSignature>signature</PayloadSignature>
    <TargetGateway>12345</TargetGateway>
  </APICredentials>
  <TranCredTransaction>
    <OperationXID>3535353</OperationXID>
    <Total>5.00</Total>
<!-- All TransactionControl elements are optional including TransactionControl -->
    <TransactionControl>
      <SendCustomerEmail>TRUE</SendCustomerEmail> <!-- TRUE/FALSE -->
      <SendMerchantEmail>TRUE</SendMerchantEmail> <!-- TRUE/FALSE -->
      <TestMode>TRUE</TestMode> <!-- TRUE/FALSE -->
      <EmailText> <!-- Up to 10 EmailTextItem elements allowed -->
        <EmailTextItem>test1</EmailTextItem>
        <EmailTextItem>test2</EmailTextItem>
        <EmailTextItem>test3</EmailTextItem>
        <EmailTextItem>test4</EmailTextItem>
        <EmailTextItem>test5</EmailTextItem>
        <EmailTextItem>test6</EmailTextItem>
        <EmailTextItem>test7</EmailTextItem>
        <EmailTextItem>test8</EmailTextItem>
        <EmailTextItem>test9</EmailTextItem>
        <EmailTextItem>test10</EmailTextItem>
      </EmailText>
    </TransactionControl>
<!-- Optional - This is saved is available in the XML transaction report. -->
```

```
    <VendorData>
      <Element>
        <Name>repId</Name>
        <Value>1234567</Value>
      </Element>
    </VendorData>
  </TranCredTransaction>
</GatewayInterface>
```

This request will receive a TransactionResponse [110].

2.4.2.5. TranForceTransaction

If you can obtain a valid authorization code, this can be used to generate a Force (capture) for a previously failed transaction. Obtain a voice approval from the credit card processor's voice approval center and use that as value for the AuthCode. The OperationXID field should contain the XID for the original failed transaction.

Figure 3.13. TranForceTransaction Diagram

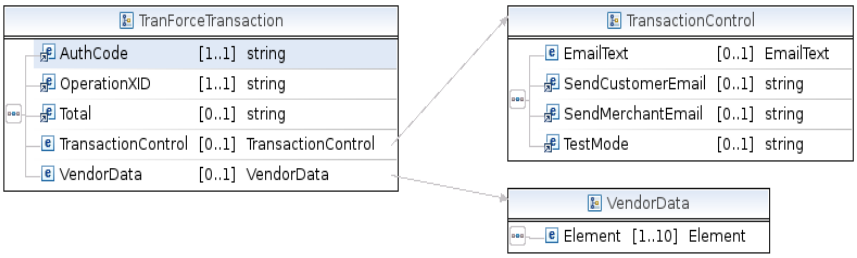


Table 3.6. TranForceTransaction Elements

Element	Description	Data Type	Required
AuthCode	Authorization code from processor	xs:string	Yes
OperationXID	XID of previous transaction	xs:string	Yes
Total	Total transaction amount	xs:string	No
TransactionControl	Transaction controls	pc:TransactionControl	No
VendorData	Merchant supplied metadata	pc:VendorData	No

For a full explanation of everything that makes up those elements, please see the schema details [114].

TranForceTransaction Request Example

```
<?xml version="1.0"?>
<GatewayInterface>
  <APICredentials>
    <Username>username</Username>
    <PayloadSignature>signature</PayloadSignature>
    <TargetGateway>12345</TargetGateway>
  </APICredentials>
  <TranForceTransaction>
    <OperationXID>3535353</OperationXID>
    <AuthCode>1234</AuthCode>
<!-- Optional - Will use original transaction amount if not specified here -->
    <Total>5.00</Total>
    <TransactionControl>
      <SendCustomerEmail>TRUE</SendCustomerEmail> <!-- TRUE/FALSE -->
      <SendMerchantEmail>TRUE</SendMerchantEmail> <!-- TRUE/FALSE -->
      <TestMode>TRUE</TestMode> <!-- TRUE/FALSE -->
      <EmailText> <!-- Up to 10 EmailTextItem elements allowed -->
        <EmailTextItem>test1</EmailTextItem>
```

```
        <EmailTextItem>test2</EmailTextItem>
        <EmailTextItem>test3</EmailTextItem>
        <EmailTextItem>test4</EmailTextItem>
        <EmailTextItem>test5</EmailTextItem>
        <EmailTextItem>test6</EmailTextItem>
        <EmailTextItem>test7</EmailTextItem>
        <EmailTextItem>test8</EmailTextItem>
        <EmailTextItem>test9</EmailTextItem>
        <EmailTextItem>test10</EmailTextItem>
      </EmailText>
    </TransactionControl>
<!-- Optional. This is saved and is available in the XML transaction report. -->
    <VendorData>
      <Element>
        <Name>repId</Name>
        <Value>1234567</Value>
      </Element>
    </VendorData>
  </TranForceTransaction>
</GatewayInterface>
```

This request will receive a TransactionResponse [110].

2.4.2.6. TranRetryTransaction

This will generate a sale transaction from a previously failed or successful transaction. The OperationXID field should contain the XID for the original transaction.

Figure 3.14. TranRetryTransaction Diagram

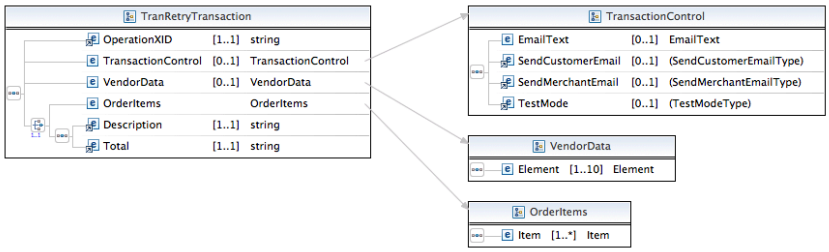


Table 3.7. TranRetryTransaction Elements

Element	Description	Data Type	Required
Description	Transaction description	xs:string	No (1)
OperationXID	XID of previous transaction	xs:string	Yes
Total	Total transaction amount	xs:string	No (1)
Description	Transaction Description	xs:string	No (1)
OrderItems	Transaction Line Items	pc:OrderItems	Yes
TransactionControl	Transaction controls	pc:TransactionControl	No
VendorData	Merchant supplied metadata	pc:VendorData	No

(1)*Total and Description are optional, but if one is used the other must be provided.*

For a full explanation of everything that makes up those elements, please see the schema details [114].

TranRetryTransaction Request Example

```
<?xml version="1.0"?>
 <GatewayInterface>
   <APICredentials>
```

```
    <Username>username</Username>
    <PayloadSignature>signature</PayloadSignature>
    <TargetGateway>12345</TargetGateway>
   </APICredentials>
   <TranRetryTransaction>
    <OperationXID>3535353</OperationXID>
     <!-- Optional.  Description and Total are optional but if used,
      both fields must be passed through -->
     <Description>1.00</Description>
     <Total>1.00</Total>
      <!-- All TransactionControl elements are optional including TransactionControl -->
      <TransactionControl>
       <SendCustomerEmail>TRUE</SendCustomerEmail> <!-- TRUE/FALSE -->
       <SendMerchantEmail>TRUE</SendMerchantEmail> <!-- TRUE/FALSE -->
       <TestMode>TRUE</TestMode> <!-- TRUE/FALSE -->
       <EmailText> <!-- Up to 10 EmailTextItem elements allowed -->
        <EmailTextItem>test1</EmailTextItem>
        <EmailTextItem>test2</EmailTextItem>
        <EmailTextItem>test3</EmailTextItem>
        <EmailTextItem>test4</EmailTextItem>
        <EmailTextItem>test5</EmailTextItem>
        <EmailTextItem>test6</EmailTextItem>
        <EmailTextItem>test7</EmailTextItem>
        <EmailTextItem>test8</EmailTextItem>
        <EmailTextItem>test9</EmailTextItem>
        <EmailTextItem>test10</EmailTextItem>
       </EmailText>
      </TransactionControl>
<!-- Optional.  This is saved and is available in the XML transaction report. -->
       <VendorData>
        <Element>
         <Name>repId</Name>
         <Value>1234567</Value>
        </Element>
       </VendorData>
   </TranRetryTransaction>
 </GatewayInterface>
```

This request will receive a *TransactionResponse* [110].

2.4.2.7. VoidTransaction

This can be used to void any sale, credit, or refund transaction if processed prior to the daily batch settlement.

Figure 3.15. VoidTransaction Diagram

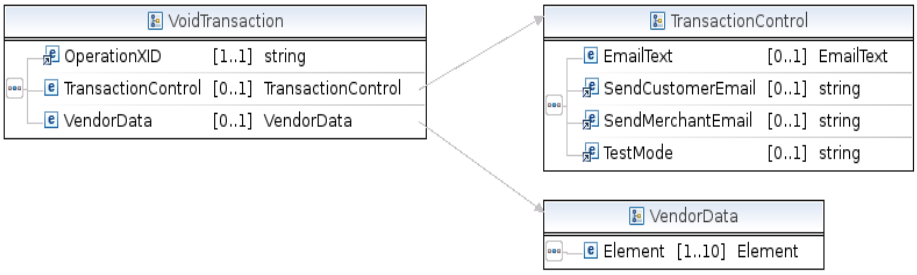


Table 3.8. VoidTransaction Elements

Element	Description	Data Type	Required
OperationXID	XID of previous transaction	xs:string	Yes
TransactionControl	Transaction controls	pc:TransactionControl	No
VendorData	Merchant supplied metadata	pc:VendorData	No

For a full explanation of everything that makes up those elements, please see the schema details [114].

VoidTransaction Request Example

```
<?xml version="1.0"?>
 <GatewayInterface>
   <APICredentials>
     <Username>username</Username>
     <PayloadSignature>signature</PayloadSignature>
     <TargetGateway>12345</TargetGateway>
   </APICredentials>
   <VoidTransaction>
    <OperationXID>3535353</OperationXID>
     <!-- All TransactionControl elements are optional including TransactionControl -->
     <TransactionControl>
      <SendCustomerEmail>TRUE</SendCustomerEmail> <!-- TRUE/FALSE -->
      <SendMerchantEmail>TRUE</SendMerchantEmail> <!-- TRUE/FALSE -->
      <TestMode>TRUE</TestMode> <!-- TRUE/FALSE -->
      <EmailText> <!-- Up to 10 EmailTextItem elements allowed -->
       <EmailTextItem>test1</EmailTextItem>
       <EmailTextItem>test2</EmailTextItem>
       <EmailTextItem>test3</EmailTextItem>
       <EmailTextItem>test4</EmailTextItem>
       <EmailTextItem>test5</EmailTextItem>
       <EmailTextItem>test6</EmailTextItem>
       <EmailTextItem>test7</EmailTextItem>
       <EmailTextItem>test8</EmailTextItem>
       <EmailTextItem>test9</EmailTextItem>
       <EmailTextItem>test10</EmailTextItem>
      </EmailText>
     </TransactionControl>
<!-- Optional. This is saved and is available in the XML transaction report. -->
      <VendorData>
       <Element>
        <Name>repId</Name>
        <Value>1234567</Value>
       </Element>
      </VendorData>
   </VoidTransaction>
 </GatewayInterface>
```

This request will receive a *TransactionResponse [110]*.

2.4.2.8. TranRefundTransaction

This can be used to void any sale, credit, or refund transaction if processed prior to the daily batch settlement.

Figure 3.16. TranRefundTransaction Diagram

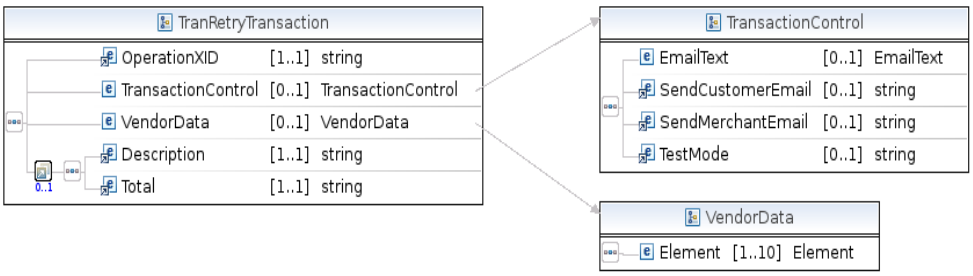


Table 3.9. TranRefundTransaction Elements

Element	Description	Data Type	Required
OperationXID	XID of previous transaction	xs:string	Yes
TransactionControl	Transaction controls	pc:TransactionControl	No

Element	Description	Data Type	Required
VendorData	Merchant supplied meta-data	pc:VendorData	No

For a full explanation of everything that makes up those elements, please see the schema details [114].

TranRefundTransaction Request Example

```
<?xml version="1.0"?>
 <GatewayInterface>
   <APICredentials>
     <Username>username</Username>
     <PayloadSignature>signature</PayloadSignature>
     <TargetGateway>12345</TargetGateway>
   </APICredentials>
   <TranRefundTransaction>
    <OperationXID>3535353</OperationXID>
     <!-- All TransactionControl elements are optional including TransactionControl -->
     <TransactionControl>
      <SendCustomerEmail>TRUE</SendCustomerEmail> <!-- TRUE/FALSE -->
      <SendMerchantEmail>TRUE</SendMerchantEmail> <!-- TRUE/FALSE -->
      <TestMode>TRUE</TestMode> <!-- TRUE/FALSE -->
      <EmailText> <!-- Up to 10 EmailTextItem elements allowed -->
       <EmailTextItem>test1</EmailTextItem>
       <EmailTextItem>test2</EmailTextItem>
       <EmailTextItem>test3</EmailTextItem>
       <EmailTextItem>test4</EmailTextItem>
       <EmailTextItem>test5</EmailTextItem>
       <EmailTextItem>test6</EmailTextItem>
       <EmailTextItem>test7</EmailTextItem>
       <EmailTextItem>test8</EmailTextItem>
       <EmailTextItem>test9</EmailTextItem>
       <EmailTextItem>test10</EmailTextItem>
      </EmailText>
     </TransactionControl>
<!-- Optional. This is saved and is available in the XML transaction report. -->
      <VendorData>
       <Element>
        <Name>repId</Name>
        <Value>1234567</Value>
       </Element>
      </VendorData>
   </TranRefundTransaction>
 </GatewayInterface>
```

This request will receive a TransactionResponse [110].

2.4.2.9. RecurUpdate

This request allows you to modify the transaction information and recurring commands for a recurring transaction.

Figure 3.17. RecurUpdate Diagram

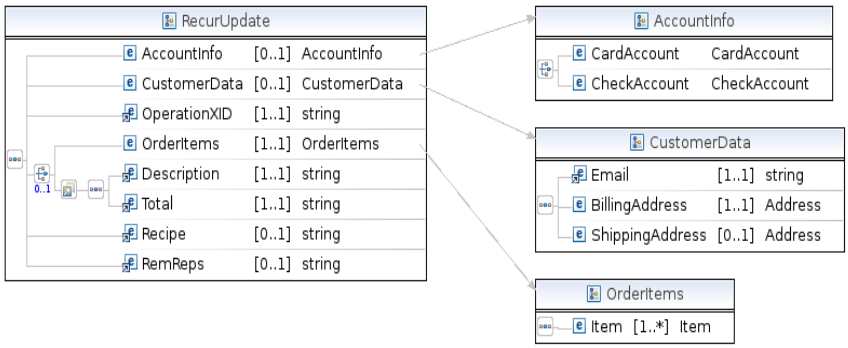


Table 3.10. RecurUpdate Elements

Element	Description	Data Type	Required
AccountInfo	Contains either card or check information	pc:AccountInfo	No (2)
CustomerData	Customer information	pc:CustomerData	No (2,3)
Description	Transaction description	xs:string	No (1,2)
OperationXID	XID of previous transaction	xs:string	Yes
OrderItems	Transaction line items	pc:OrderItems	No (1,2)
Recipe	Recurring recipe name	xs:string	No (2)
RemReps	Remaining recurring repetitions	pc:VendorData	No (2)
Total	Total transaction amount if different from previous amounts	xs:string	No (1,2)

(1) *If used only one of OrderItems or Total and Description must be provided although all are optional.*

(2) *Other than OperationXID, all of the child elements of RecurUpdate are individually optional but you must pass one of Recipe, RemReps, CustomerData, OrderItems or Total.*

(3) *The CustomerData element in RecurUpdate only requires one of the following sub-elements to be provided: AccountInfo, Email, BillingAddress, CustId or ShippingAddress.*

For a full explanation of everything that makes up those elements, please see the schema details [114].

RecurUpdate Request Example

```
<?xml version="1.0"?>
<GatewayInterface>
  <APICredentials>
    <Username>username</Username>
    <PayloadSignature>signature</PayloadSignature>
    <TargetGateway>12345</TargetGateway>
  </APICredentials>
<!-- Other than OperationXID, all of the child elements of RecurUpdate
are individually optional but you must pass
one of Recipe, RemReps, CustomerData, OrderItems or Total -->
  <RecurUpdate>
    <OperationXID>3535353</OperationXID>
<!-- Optional.-->
    <RemReps>123</RemReps>
<!-- Optional.-->
    <Recipe>Recipe Name</Recipe>
<!-- Optional.  Will update customer info tied to recurring transaction if passed -->
    <CustomerData>
      <Email>demo@demo.com</Email>
      <CustId>12345</CustId>
<!-- Optional -->
      <BillingAddress>
        <Address1>test</Address1>
        <FirstName>John</FirstName>
        <LastName>Smith</LastName>
        <City>Bountiful</City>
        <State>UT</State>
        <Zip>84032</Zip>
        <Country>USA</Country>
        <Phone>801-555-1212</Phone>
      </BillingAddress>
<!-- Optional -->
```

```
      <ShippingAddress>
        <Address1>test</Address1>
        <FirstName>John</FirstName>
        <LastName>Smith</LastName>
        <City>Bountiful</City>
        <State>UT</State>
        <Zip>84032</Zip>
        <Country>USA</Country>
      </ShippingAddress>
<!-- Optional.  Will update customer info tied to recurring transaction if passed-->
      <AccountInfo>
<!-- For Credit card transactions. -->
        <CardAccount>
          <AccountNumber>5454545454545454</AccountNumber>
          <ExpirationMonth>01</ExpirationMonth>
          <ExpirationYear>2000</ExpirationYear>
          <CVVNumber>123</CVVNumber><!-- CVV is Optional -->
        </CardAccount>
<!-- For EFT transactions. -->
        <CheckAccount>
          <AccountNumber>123456</AccountNumber>
          <ABA>324377516</ABA>
        </CheckAccount>
      </AccountInfo>
    </CustomerData>
<!-- Only one of OrderItems or Total elements may be passed in but neither is required -->
    <OrderItems>
      <Item>
        <Description>item1</Description>
        <Cost>5</Cost>
        <Qty>1</Qty>
      </Item>
    </OrderItems>
<!-- To use the Total element the original transaction can only have
one item associated with it  -->
    <Total>5.00</Total>
  </RecurUpdate>
</GatewayInterface>
```

This request will receive a RecurUpdateResponse [113].

2.4.2.10. RecurDetails

This request allows you to query for details on an existing recurring transaction. This includes information about the recurring details as well as information that helps determine if the credit card tied to the recurring transaction is expired or set to expire soon.

RecurDetails Request Example

```
<?xml version="1.0"?>
 <GatewayInterface>
   <APICredentials>
     <Username>username</Username>
     <PayloadSignature>signature</PayloadSignature>
     <TargetGateway>12345</TargetGateway>
   </APICredentials>
   <RecurDetails>
     <OperationXID>3535353</OperationXID>
   </RecurDetails>
 </GatewayInterface>
```

This request will receive a RecurDetailsResponse [113].

2.4.2.11. TransactionStatus

The TransactionStatus request provides a way to determine the status of a request that was interrupted for some reason. For a transaction where the response was lost, the transaction XID would be unknown to the client. To be

able to use this request you needed to have passed through a unique transaction identifier of your own using the VendorData elements which can be passed through with the original request. Ideally you would pass through a unique ID with every request, although this request will do the request based on multiple VendorData elements. If more than one record is matched based on the passed VendorData the first matching transaction will be used to generate the response. The response will contain a warning message in the WarningMessage field indicating that multiple transactions were matched in this case. Please note that this is NOT a search tool since it does not return multiple responses. This tool was also not written to automate the mass retrieval of past transaction history. Please use the Transaction Report API [163] interface for that purpose.

Figure 3.18. TransactionStatus Diagram

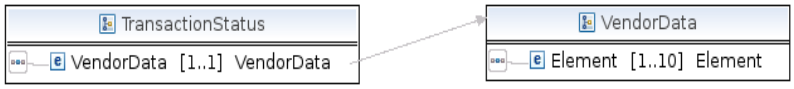


Table 3.11. TransactionStatus Elements

Element	Description	Data Type	Required
TransactionControl	Transaction controls	pc:TransactionControl	No but only TestMode is supported
VendorData	Merchant supplied meta-data	pc:VendorData	Yes

TransactionStatus Request Example

```
<?xml version="1.0"?>
 <GatewayInterface>
   <APICredentials>
     <Username>username</Username>
     <PayloadSignature>signature</PayloadSignature>
     <TargetGateway>12345</TargetGateway>
   </APICredentials>
   <TransactionStatus>
    <VendorData>
     <Element>
      <Name>field1</Name>
      <Value>1234567</Value>
     </Element>
     <Element>
      <Name>field2</Name>
      <Value>6789123</Value>
     </Element>
    </VendorData>
   </TransactionStatus>
  <TransactionControl>
     <TestMode>FALSE</TestMode>
   </TransactionControl>
 </GatewayInterface>
```

This request will receive a TransactionResponse [109].

2.5. Responses

All API requests will receive some type of response providing data or an indication of the the outcome of the request. These responses are returned as a string of XML in the same connection as the request.

2.5.1. TransactionResponse

All of the “transaction” requests including TransactionStatus return the TransactionResponse [110] structure. The primary element is TransactionResult. The customer data fields are populated with data from the request and/or as designated by the OperationXID field in the request.

Figure 3.19. TransactionResponse - TransactionResult Diagram

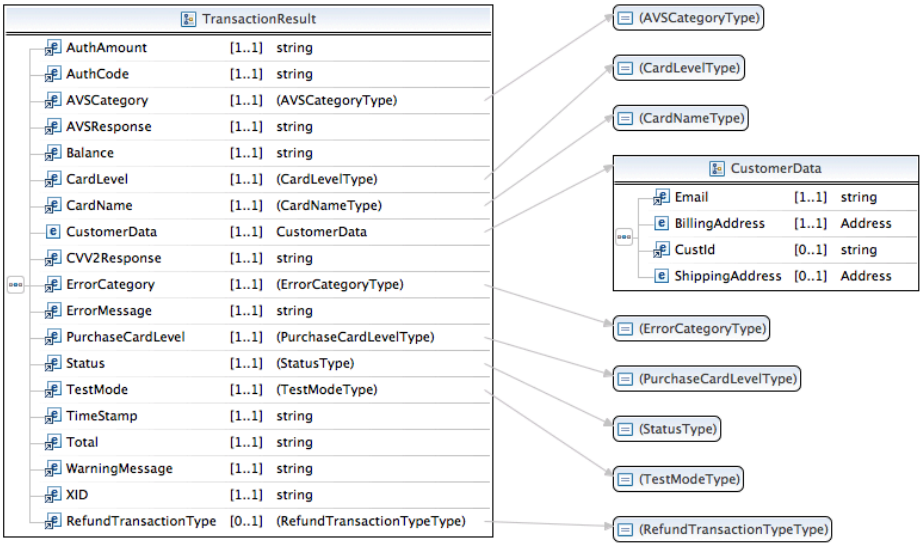


Table 3.12. TransactionResult Elements

Element	Description	Data Type	Provided
AuthAmount	Amount authorized	xs:string	Yes
AuthCode	Transaction authorization code	xs:string	Yes
AVSCategory	Address verification response category	pc:AVSCategory	Yes
AVSResponse	Address verification response	xs:string	Yes
Balance	Funds available on card account	xs:string	If Available
CardLevel	The card service level	pc:CardLevelType	If Available
CardName	Type of credit card used	pc:CardName	Yes
CustomerData	Customer information	pc:CustomerData	Yes
CVV2Response	Security code verification response	xs:string	Yes
ErrorCategory	Error category	pc:ErrorCategory	Yes
ErrorMessage	Error message	xs:string	Yes
PurchaseCardLevel	Indicates if card is a purchase card	pc:PurcahseCardLevel	If Available
RefundTransactionType	Indicates refund type	pc:RefundTransactionType	If Available
Status	Transaction status	pc:Status	Yes
TestMode	Indicates if transaction was run in test mode	pc:TrueFalse	Yes
TimeStamp	Date and time of transaction	xs:string	Yes
Total	Total transaction amount if different from previous amounts	xs:string	Yes

Element	Description	Data Type	Provided
WarningMessage	Warning message	xs:string	Yes (1)
XID	Unique transaction identifier assigned by the gateway	xs:string	Yes

(1) *This is currently only used with the TransactionStatus action in the case that more than one transaction is matched by the supplied VendorData elements.*

TransactionResponse Example

```
<?xml version="1.0" standalone="yes"?>
<GatewayInterface>
  <TransactionResponse>
    <TransactionResult>
      <Status>text</Status> <!-- Will be one of: error, fail, FAILED, ok -->
      <ErrorCategory>text</ErrorCategory>
<!-- ErrorCategory will be one of :
AVS_FAILURE - Transaction will be automatically voided.
CVV2_FAILURE - Transaction will be automatically voided.
INTERNAL_ERROR - Something unexpected happened. Refer to the message
PROCESSOR_ERROR - Something such as DECLINED, etc .
PROCESSOR_FAIL -
REQUEST_FORMAT - Request received has an invalid format.
REQUEST_VALIDATION - XML content is invalid. -->
      <ErrorMessage>text</ErrorMessage>
<!-- ErrorMessage could be anything. -->
      <AuthAmount>5.00</AuthAmount>
<!-- Actual authorized amount. -->
      <AuthCode></AuthCode>
<!-- Authorization code received from processing network. -->
      <AVSCategory></AVSCategory>
<!-- AVSCategory will be one of :
address - Address Matched
address_postal - Address and postal patched
address_zip5 - Address and five digit zip matched
address_zip9 - Address and nine digit zip matched
address_ok_postal_format_error - Address matched, postal format error
global_non_participant - International with no AVS support
international_address_not_verified - International with no AVS support
no_match - No address or postal match
no_response - No response
not_allowed - Not allowed
postal - Postal match
postal_ok_address_format_error - Postal matched, address format error
service_not_supported - AVS service not supported for card
unavailable - AVS service unavailable.
zip5 - Five digit zip matched
zip9 - Nine digit zip matched -->
      <AVSResponse></AVSResponse>
<!-- AVSResponse is actual AVS response received from the processing network. -->
      <Balance></Balance>
<!-- Available account funds if provided by processor -->
      <CardLevel></CardLevel>
<!-- This identifies the card service level if provided by processor. -->
      <CardName></CardName>
<!-- CardName will be one of: American Express
Australian Bankcard
Discover
Diners Club/Carte Blanche
enRoute
Japanese Credit Bureau
MasterCard
Visa -->
      <CVV2Response></CVV2Response>
<!-- CVV2Response is actual CVV response received from the processing network. -->
      <PurchaseCardLevel></PurchaseCardLevel>
```

```
<!-- Identifies purchase card type if provided by processor. -->
      <TimeStamp></TimeStamp>
      <TestMode>FALSE</TestMode>
<!-- TestMode indicates the test status of your gateway account. TRUE/FALSE -->
      <Total></Total>
      <XID></XID>
      <CustomerData>
        <BillingAddress>
          <Address1></Address1>
          <City></City>
          <FirstName></FirstName>
          <LastName></LastName>
          <State></State>
          <Zip></Zip>
          <Country></Country>
          <Phone></Phone>
        </BillingAddress>
        <ShippingAddress>
          <Address1></Address1>
          <City></City>
          <FirstName></FirstName>
          <LastName></LastName>
          <State></State>
          <Zip></Zip>
          <Country></Country>
          <Phone></Phone>
        </ShippingAddress>
      </CustomerData>
    </TransactionResult>
  </TransactionResponse>
</GatewayInterface>
```

2.5.2. RecurUpdateResponse

The RecurUpdate request will generate a response like this:

```
<?xml version="1.0" standalone="yes"?>
 <GatewayInterface>
  <RecurUpdateResponse>
   <Status>ok</Status>
   <ErrorCategory></ErrorCategory>
   <ErrorMessage></ErrorMessage>
   <TimeStamp>20060621154341</TimeStamp>
   <TestMode>FALSE</TestMode> <!-- TRUE/FALSE -->
    <RecurDetails>
     <RemReps>10</RemReps>
     <RecipeName>daily</RecipeName>
     <RecurTotal>1.00</RecurTotal>
    </RecurDetails>
  </RecurUpdateResponse>
 </GatewayInterface>
```

2.5.3. RecurDetailsResponse

The RecurDetails request will generate a response like this:

```
<?xml version="1.0" standalone="yes"?>
 <GatewayInterface>
  <RecurDetailsResponse>
    <Status>ok</Status>
    <ErrorCategory></ErrorCategory>
    <ErrorMessage></ErrorMessage>
    <TimeStamp>20060621154341</TimeStamp>
    <TestMode>FALSE</TestMode> <!-- TRUE/FALSE -->
     <RecurDetails>
      <CardExpired>FALSE</CardExpired> <!-- TRUE/FALSE -->
      <CardExpiresWithinThirty>FALSE</CardExpiresWithinThirty> <!-- TRUE/FALSE -->
      <CardLastFour>1234</CardLastFour>
      <CardName>Visa</CardName>
```

```
      <RemReps>10</RemReps>
      <RecipeName>daily</RecipeName>
      <RecurTotal>1.00</RecurTotal>
     </RecurDetails>
  </RecurDetailsResponse>
</GatewayInterface>
```

2.6. Schema Details

This is a list in alphabetical order of all of the elements that can be used with the API requests and the sub-elements that make up those elements.

2.6.1. AccountInfo

This is used to pass the account information for a payment. It is required for AuthTransaction [96] and CreditTransaction [99]. It is optional for RecurUpdate [107]. If used, only one payment type may be passed.

Figure 3.20. AccountInfo Diagram

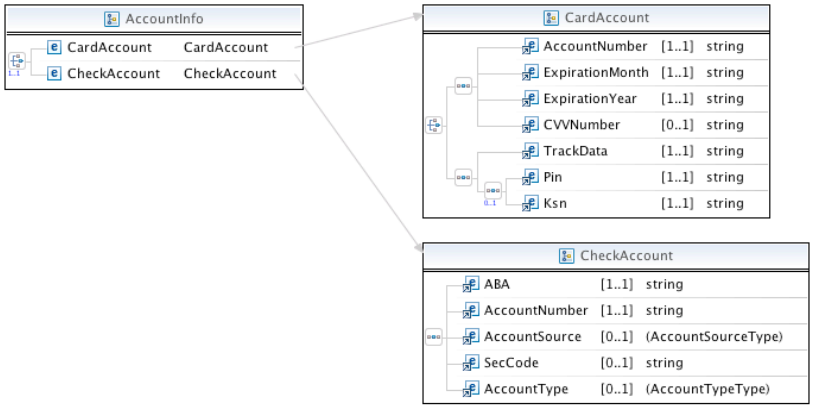


Table 3.13. AccountInfo Elements

Field	Description	Data Type
CardAccount	Card data	pc:CardAccount
CheckAccount	Checking account data	pc:CheckAccount

2.6.2. Address

This set can be used to for BillingAddress and ShippingAddress as sub-sets of CustomerData . Both sets are supported AuthTransaction [96], CreditTransaction [99], and RecurUpdate [107]. ShippingAddress is always optional. BillingAddress is optional for RecurUpdate, but required for the other two requests.

Figure 3.21. Address Diagram

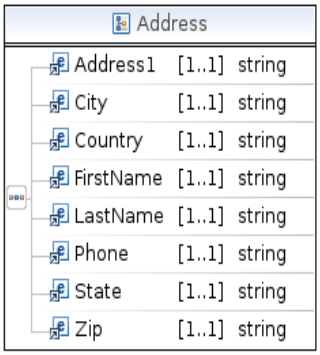


Table 3.14. Address Elements

Field	Description	Data Type
Address1	Street address	xs:string
City	Name of city	xs:string
Country	Name of country	xs:string
FirstName	Cardholder first name	xs:string
LastName	Cardholder last name	xs:string
Phone	Phone number	xs:string
State	Name of state	xs:string
Zip	Postal code	xs:string

2.6.3. APICredentials

This set is required for all requests.

Figure 3.22. APICredentials Diagram

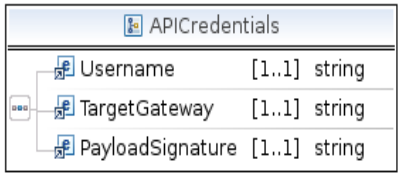


Table 3.15. APICredentials Elements

Field	Description	Data Type
Username	Assigned API Username	xs:string
TargetGateway	5 digit gateway ID	xs:string
PayloadSignature	Calculated signature hash	xs:string

2.6.4. CardAccount

This is used to pass the credit card information for a payment. It is required for AuthTransaction [96] and CreditTransaction [99]. It is optional for RecurUpdate [107]. Pin and Ksn are both required for pin/debit transactions. TrackData is used in place of AccountNumber, ExpirationMonth, and ExpirationYear for swiped transactions. TrackData can include, track1, track2, or both tracks 1 and 2. If both are sent, you must send the start and end sentinel characters for both sets. The formatting of track1 and track2 data is explained at *http://en.wikipedia.org/wiki/ISO/IEC_7813*. CVV is an optional field.

Figure 3.23. CardAccount Diagram

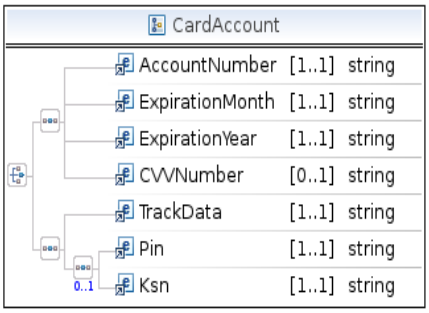


Table 3.16. CardAccount Elements

Field	Description	Data Type
AccountNumber	Credit card account	xs:string
ExpirationMonth	Two digit numeric month	xs:string
ExpirationYear	Four digit numeric year	xs:string
CVVNumber	Cardholder verification number	xs:string
TrackData	Swipe data	xs:string
Pin	Encrypted Debit PIN code	xs:string
Ksn	Encrypted key serial number	xs:string

2.6.5. CheckAccount

This is used to pass the checking account information for a payment. It is required for AuthTransaction [96] and CreditTransaction [99]. It is optional for RecurUpdate [107]. AccountSource, AccountType, and SecCode are conditionally optional, based on what your check processor requires.

Figure 3.24. CheckAccount Diagram

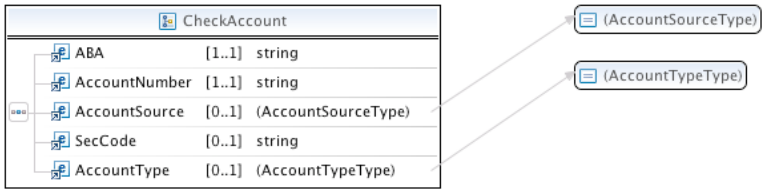


Table 3.17. CardAccount Elements

Field	Description	Data Type
ABA	Nine digit bank ABA routing number	xs:string
AccountNumber	Checking account number	xs:string
AccountSource	Identifies type of account (*checking* or *savings*)	xs:string
AccountType	Identifies type of bank account (*personal* vs *business*)	xs:string
SecCode	Standard Entry Class identifier	xs:string

2.6.6. CustomerData

This is used to pass the customer information for a transaction. It is required for AuthTransaction [96] and CreditTransaction [99]. It is optional for RecurUpdate [107].

Figure 3.25. CustomerData Diagram

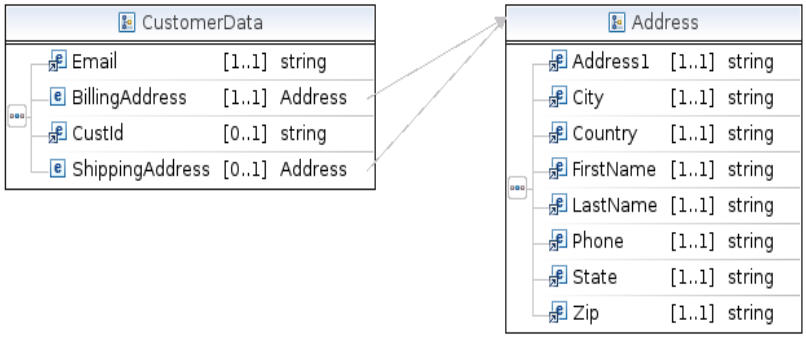


Table 3.18. CustomerData Elements

Field	Description	Data Type
BillingAddress	Account billing address	pc:BillingAddress
CustId	Custom ID	xs:string
Email	Email address	xs:string
ShippingAddress	Shipping address	pc:ShippingAddress

2.6.7. Element

This is a sub-set of VendorData that can be used to pass unique meta-data fields to the gateway.

Figure 3.26. Element Diagram

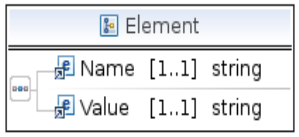


Table 3.19. Element Elements

Field	Description	Data Type
Name	Unique variable name	xs:string
Value	Unique variable value	xs:string

2.6.8. EmailText

Use this to include up to ten separate text fields that will be appended together, separated with a line space, and included in the body of the confirmation emails.

Figure 3.27. EmailText Diagram

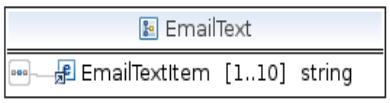


Table 3.20. Element Elements

Field	Description	Data Type
EmailTextItem	Text to be included in emails	xs:string

2.6.9. Item

This is the sub-set of OrderItems used to submit the payment amount and item description.

Figure 3.28. Item Diagram

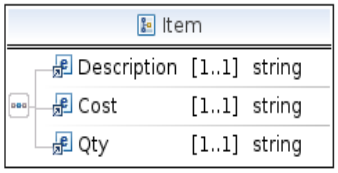


Table 3.21. Item Elements

Field	Description	Data Type
Description	Item identifier	xs:string
Cost	Amount value	xs:string
Qty	Quantity	xs:string

2.6.10. OrderItems

This can be used for AuthTransaction [96] and for RecurUpdate [107] instead of the Total and Description.

Figure 3.29. OrderItems Diagram

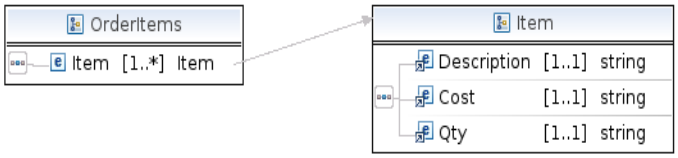


Table 3.22. OrderItems Element

Field	Description	Data Type
Item	Contains cost, quantity, and description of item	pc:Item

2.6.11. RecurringData

This is the set of recurring commands that are optional for use with AuthTransaction [96]. Either OrderItems or Total and Description can be used to pass the order descriptor and payment amount.

Figure 3.30. RecurringData Diagram

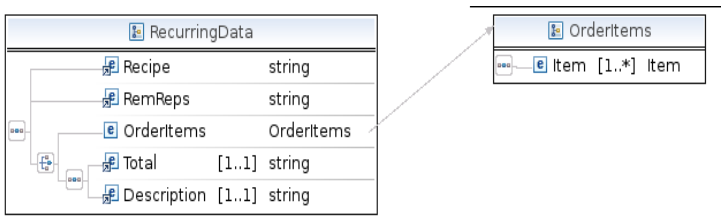


Table 3.23. RecurringData Elements

Field	Description	Data Type
Recipe	Recurring recipe	xs:string

Field	Description	Data Type
RemReps	Recurring repetitions	xs:string
OrderItems	Payment amount and description	xs:string
Total	Recurring amount value	xs:string
Description	Recurring item identifier	xs:string

2.6.12. TransactionControl

These elements can be used to dynamically control email notifications and test mode settings for each transaction. This is available for all “transaction” actions. These are optional.

Figure 3.31. TransactionControl Diagram

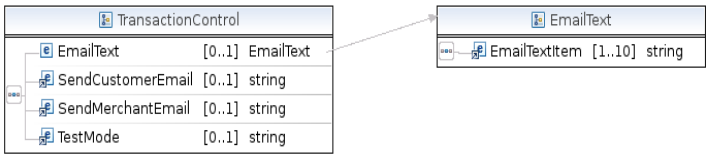


Table 3.24. RecurringData Elements

Field	Description	Data Type
EmailText	Contains up to ten fields that can be passed and included on the email notifications for the transaction	pc:EmailText
SendCustomerEmail	Controls whether customer notification is sent or not	xs:string
SendMerchantEmail	Controls whether merchant notification is sent or not	xs:string
TestMode	Controls whether transaction is run as a live or test transaction	xs:string

2.6.13. VendorData

This can be used to pass up to ten meta-data sets to the gateway that can be queried for at a later time.

Figure 3.32. VendorData Diagram

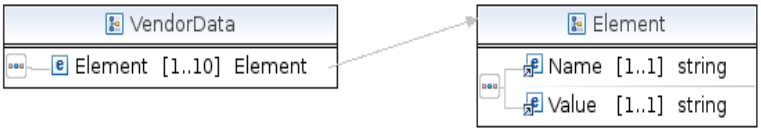


Table 3.25. VendorData Element

Field	Description	Data Type
Element	One meta-data object	pc:Element

2.7. Simple Response Values

These indicators will help you identify things about your transactions when the response comes back.

2.7.1. AVSCategory

Table 3.26. AVSCategory

Value	Definition
address	Street address matched, postal code did not
address_postal	Street address and postal coded matched
address_zip5	Street address and five digit ZIP coded matched
address_zip9	Street address and nine digit ZIP coded matched
address_ok_postal_format_error	Street address matched, postal code formatting error
global_non_particiapant	International with no AVS support
international_address_not_verified	International with no AVS support
no_match	Neither ZIP code nor street address match
no_response	No AVS response was returned by issuer
not_allowed	AVS not allowed
postal	Postal code matched, street address did not
postal_ok_address_format_error	Postal code matched, address formatting error
service_not_supported	AVS service unsupported for the card
unavailable	AVS service unavailable
zip5	Five digit zip code matched
zip9	Nine digit zip code matched

2.7.2. Balance

Table 3.27. Balance

Value
Numeric value of available credit line

2.7.3. CardLevel

Table 3.28. CardLevel

Value
VISA_TRADITIONAL
VISA_TRADITIONAL_REWARDS
VISA_SIGNATURE
VISA_INFINITE
VISA_BUSINESS
VISA_CHECK
VISA_COMMERCE
VISA_CORPORATE
MASTERCARD_EUROCARD_DINERS
PRIVATE_LABEL
PROPRIETARY
VISA_PURCHASE_CARD
INTERLINK

Value
VISA_TRAVELMONEY
VISA_SIGNATURE_BUSINESS
VISA_BUSINESS_CHECK
VISA_GENERAL_PREPAID
VISA_PREPAID_GIFT
VISA_PREPAID_HEALTH
VISA_PREPAID_COMMERCIAL
VISA_GSA_CORPORATE_TANDE
PRIVATE_LABEL_PREPAID
VISA_PURCHASE_FLEET
VISA_GSA_PURCHASE
VISA_GSA_PURCHASE_FLEET
AMEX
DISCOVER

2.7.4. CardName

Table 3.29. CardLevel

Value
American Express
Australian Bankcard
Discover
Diners Club/Carte Blanche
enRoute
Japanese Credit Bureau
Master Card
Visa

2.7.5. PurchaseCardLevel

Table 3.30. PurchaseCardLevel

Value
Text will indicate if the account used was a purchase card.

2.7.6. ErrorCategory

Table 3.31. ErrorCategory

Value	Definition
AVS_FAILURE	Transaction will be auto-voided because of anti-fraud settings
CVV2_FAILURE	Transaction will be auto-voided because of anti-fraud settings
INTERNAL_ERROR	An unexpected error was experienced
PROCESSOR_ERROR	Processor response indicating purpose for non-approval

Value	Definition
PROCESSOR_FAIL	Failure in communication with processing network
REQUEST_FORMAT	Request received has invalid formatting
REQUEST_VALIDATION	XML content is invalid or credentials are incorrect

2.7.7. Status

Table 3.32. Status

Value	Definition
error	Transaction was processed but not approved
fail	Transaction failed prior to processing
FAILED	Transaction failed prior to processing
ok	Transaction was processed successfully

2.7.8. TrueFalse

Table 3.33. Status

Value	Definition
TRUE	Yes or On
FALSE	No or Off

2.8. Considerations

- The XML Connection method is not activated on a gateway account by default. If you desire to use the XML Connection method, an XML activation request needs to be sent to the *support team*. Also, be sure to enable the XML API in the Account Settings [19] section of the Control Panel.
- There are certain characters such as `&` and `<` that may not appear in a XML request unless they are part of the actual XML structure. These characters have to be "escaped" by using the forms `&` and `<`. More information about XML markup can be found at *www.w3.org/TR/REC-xml/#syntax*.
- We suggest that you do not try and create XML structures by simply appending to a string with print statements. Using a XML generation library will ensure that you do not accidentally create badly formatted documents due to user input. The disadvantage of using a library is that it will take a little longer to code. If you do not use a library you need to make sure that any XML node text you are adding that is coming from a user gets escaped as described above. If you do not escape these characters you will have transactions occasionally fail due to a bad XML structure.
- Here are some suggested XML generation libraries for different development environments. This is by no means an authoritative list and there are other good libraries available:
  - Java *http://jakarta.apache.org/ecs/index.html*
  - Perl *http://search.cpan.org/~bholzman/XML-Generator-1.01/Generator.pm*
  - PHP *http://us2.php.net/dom*
  - Ruby *http://www.germane-software.com/software/rexml/*
  - .NET/C# *http://msdn2.microsoft.com/en-us/library/system.xml.xmltextwriter(VS.71).aspx*

3. HTML Connection Methods

These methods provide the simplest method for communicating with the gateway. Simply add an HTML form to your site to begin accepting transactions immediately. This is used by many merchants since it can be done from

any HTML web page. An HTML order for can be created in just a few minutes. These methods are secure and PCI compliant, but there are added layers of additional protection for you if you use the preferred API method for integration instead.

There are three HTML connection methods. They are the Standard [123] format, the Split Form [129] format, and the BuyNow [135] format. You can choose the method that best suits your needs. Merchants who do not have a secure server should use our BuyNow or Split Form-based protocol and merchants who have a secure server should use our Standard Form-based protocol.

3.1. Standard Form

3.1.1. Standard Form - Introduction

If you want to capture cardholder information on your own secure server, we suggest that you utilize our Standard Form protocol. Generate an HTTPS post from your page to submit required and optional fields to our server for each transaction attempt. Submit your post to

https://secure.itransact.com/cgi-bin/rc/ord.cgi

The transaction will either be approved or rejected when the transaction is submitted to the processor. To make the process more seamless, you can utilize the Return Mode function [142] to bypass the gateway's continue page that would normally display after a transaction, and instead, go directly to a page on your site. Other functions (Passback [143] and Lookup [143]) are available to allow data concerning the transactions passed back to your server.

3.1.2. Standard Form - Required Fields

Please pass these fields for all Standard Form transactions (with the exception of the non-used payment type, pass only card data or check data in one request):

- **mername** - The value is the name of your company.
- **vendor_id** - This value is your Order Form UID value or Gateway ID number. For security purposes, we suggest using the randomly generated UID value. This value can be up to 20 characters.
- **home_page** - The url of your website.
- **ret_addr** - This value is the HTML page or dynamic script that a customer and/or data is passed to after an order is submitted. Please remember that any images or information on this page must have the absolute addresses. In most cases, this page will be emulated in our secure server environment.
- ***_desc** - This value is the description of the order item. * indicates the item number; `1_desc`, `2_desc`, `3_desc`, etc. A field separator of underscore [`1_desc`] or dash [`1-desc`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_desc`, `item_2_desc`, etc. Only one naming convention per order form may be used.
- ***_cost** - This value is the numeric amount of the price of the item including dollars, decimal, and cents [`XXX.xx`]. * indicates the item number; `1_cost`, `2_cost`, `3_cost`, etc. A field separator of underscore [`1_cost`] or dash [`1-cost`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_cost`, `item_2_cost`, etc. Only one naming convention per order form may be used.
- ***_qty** - This value is the number of the item desired - * indicates the item number; `1_qty`, `2_qty`, `3_qty`, etc. A field separator of underscore [`1_qty`] or dash [`1-qty`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_qty`, `item_2_qty`, etc. Only one naming convention per order form may be used.
- **first_name** - This value is the customer's first name - 50 maximum characters.
- **last_name** - This value is the customer's last name - 50 maximum characters.
- **address** - This value is the cardholder's billing street address - 100 maximum characters.
- **city** - This value is the cardholder's city of their billing address - 25 maximum characters.

- **state** - This value is the cardholder's state of their billing address - If the transaction is foreign and there is no corresponding state, please enter a comma into this field - 25 maximum characters.
- **zip** - This value is either the five digit or nine digit billing postal code - If the transaction is foreign and there is no corresponding postal code, please enter a comma into this field - 12 maximum characters.
- **country** - This value is the cardholder's country of their billing address - 45 maximum characters.
- **phone** - This value is the customer's phone number - 25 maximum characters.
- **email** - This value is the customer's email address - 255 maximum characters.
- **FOR CREDIT CARD TRANSACTIONS**
  - ccnum - This 15 or 16-digit numeric value is the credit card number, entered without spaces or hyphens.
  - ccmo - This value is the full name of the month of the credit card's expiration date.
  - ccyr - The four digit numeric value of the year of the credit card's expiration date.
- **FOR CHECK/EFT TRANSACTIONS**
  - aba - This nine digit numeric value is the ABA routing number from the checking account.
  - account - This numeric value is the account number of the checking account.
  - account_source - This is used for NACHA authorized accounts and can be the value of checking or savings.
  - account_type - This is used for EFT authorized accounts and can be the value of personal or business.
  - sec_code - This three letter value is the standard entry category for a transaction.

    Potential values:

    - PPD - Prearranged payment and deposit
    - CCD - Corporate credit or debit
    - ARC - Accounts receivable entry
    - BOC - Back office conversion
    - POP - Point of purchase
    - RCK - Returned check entry
    - WEB - Internet initiated entry
    - TEL - Telephone initiated entry

Example

```
<HTML>
<HEAD>
<title>Secure Order Form</title>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<FORM METHOD='POST' ACTION='https://secure.itransact.com/cgi-bin/rc/ord.cgi'
AUTOCOMPLETE ='off'><center> <BRgt; <P>
<input type='hidden' name='cost1' value="1.99">
<input type='hidden' name='desc1' value="ORDER ITEM">
<input type='hidden' name='formtype' value="2">
<input type='hidden' name='home_page' value="http://www.YourHomePage.com">
<input type='hidden' name='items' value="1">
<input type='hidden' name='layout1' value="3">
<input type='hidden' name='mername' value="Your Company">
```

```
<input type='hidden' name='ret_addr' value="http://www.YourReturnAddress.com">
<input type='hidden' name='vendor_id' value="XXXXX">
<table width=590> <TR> <TD>
<P> Select the item(s) below that you would like to order.
Your transaction will be confirmed by email.</P> <!/table>
<INPUT type="checkbox" name='1-desc' value="ORDER ITEM">
<B>ORDER ITEM</B> (&#036 1.99 each)<BR>
<INPUT type = 'hidden' NAME='1-cost' value='1.99'>
<INPUT type = 'hidden' NAME='1-qty' value='1'>
</TD></table><TABLE WIDTH=590>
<td><strong><center><HR><font size=-1 color=blue>
<B>GENERAL INFORMATION</b>
</FONT><HR></strong></center></td></tr> </TABLE>
<TABLE WIDTH=590> <TR><TD ALIGN="right">
<B>First Name: </B> </TD><TD><INPUT NAME="first_name" SIZE=15>
<B>Last Name: </B> <INPUT NAME="last_name" SIZE=15> <BR>
</TD> </TR> <TR> <TD ALIGN="right">
<B>Address: </B> </TD><TD>
<INPUT NAME="address" SIZE=30><BR></TD> </TR>
<TR> <TD ALIGN="right"><B>City:</B> </TD>
<TD><INPUT NAME="city" SIZE=15>
<B> State: </B> <INPUT NAME="state" SIZE=2 MAXLENGTH=2>
<B> ZIP: </B> <INPUT NAME="zip" SIZE=10 maxlength=5>
<BR></TD> </TR>
<TR> <TD ALIGN="right"><B>Country: </B> </TD>
<TD><INPUT NAME="country" SIZE=45 value="USA"><BR></TD> </TR>
<TR> <TD ALIGN="right"><B>Phone Number: </B> </TD>
<TD><strong><INPUT NAME="phone" SIZE=15><BR></TD></TR>
<TR> <TD ALIGN="right"><B>E-Mail Address: </B> </TD>
<TD><INPUT NAME="email" SIZE=30><BR></TD></TR> </TABLE>
<!HR width=590></CENTER><center>
<font size=-1 color=blue>CHECKING INFORMATION</font></strong><hr></center></font>
<I>At the bottom of your check is a series of numbers, separated by symbols.
It is not necessary to enter symbols or spaces in the fields provided below.<br>
<TABLE> <TR> <TD ALIGN="right">
<B> Enter the series of <i>nine</i> numbers<BR><I>between
</I> this symbol </B></TD>
<TD><img src="https://secure.itransact.com/images/1.gif">
<INPUT NAME="aba" SIZE=9 MAXLENGTH=9>
<img src="https://secure.itransact.com/images/1.gif">
<BR></TD></TR><BR>
<TR> <TD ALIGN="right"><B>Enter the series of numbers<BR>
found <I>before</I>this symbol </B></TD>
<TD><img src="https://secure.itransact.com/images/2.gif">
<INPUT NAME="account" SIZE=20 MAXLENGTH=25>
<img src="https://secure.itransact.com/images/2.gif"></TD> </TR>
<TR> <TD ALIGN="right"><B>Check number to use: </B> </TD>
<TD>   <INPUT NAME="check_num" SIZE=10>
<B>(Not required.)</B> </TD></TR>
<TR> </TR> </TABLE></table>
<center> <table width=590>
<TD><font><center><hr><strong>
<FONT SIZE=-1 color=blue>CREDIT CARD INFORMATION</FONT></strong><hr></center></font>
<TABLE> <TR>
<TD ALIGN="right"><B>Card Number: </B> </TD>
<TD><INPUT NAME="ccnum" SIZE=16><B>Exp. Date: </B>
<select NAME="ccmo">
<option>
<option>January
<option>February
<option>March
<option>April
<option>May
<option>June
<option>July
<option>August
<option>September
<option>October
<option>November
<option>December
```

```
</select>
<select name="ccyr">
<option>
<option>2015
<option>2016
<option>2017
<option>2018
<option>2019
</select> </td> </TR></TABLE> </TD> </TABLE>
<HR width=590> <center></center>
<INPUT TYPE="SUBMIT" NAME="submit" VALUE="Submit Securely"> </CENTER> </FORM>
</BODY>
</HTML>
```

3.1.3. Standard Form - Optional Fields

These fields can be used to enhance your Standard forms, support additional data, and allow you to receive data back to your server after a transaction.

- ***_VARIABLE** - This value is an attribute of the order item - * indicates the item number; `1_VARIABLE`, `2_VARIABLE`, `3_VARIABLE`, etc. "`VARIABLE`" can be anything. A field separator of underscore [`1_VARIABLE`] or dash [`1-VARIABLE`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_VARIABLE`, `item_2_VARIABLE`, etc. Only one naming convention per order form may be used.

  ```
  <input name="1_size" value="Large">
  <input name="1_color" value="White">
  ```

- **preauth** - Pass this command if you only want to pre-authorize a credit card to be charged at a later time.

  ```
  <input name="preauth"/>
  ```

- **ret_mode** - The value for this command is either `post` or `redirect`.

  ```
  <input type="hidden" name="ret_mode" value="post">
  ```

  or

  ```
  <input type="hidden" name="ret_mode" value="redirect">
  ```

  Please review all of the features of the Return Mode [142]function [142].

- **lookup** - The value(s) of this command is/are the desired predetermined values of some of the required/non-required fields.

  ```
  <input type="hidden" name="lookup" value="first_name">
  ```

  Please review all of the details of the Lookup function [143].

- **passback** - The value(s) of this command is/are the desired values of most of the non-Lookup fields.

  ```
  <input type="hidden" name="ordernum" value="order#999">
  <input type="hidden" name="passback" value="ordernum">
  ```

  Please review all of the details of the Passback [143] *[143]*function [143].

- **post_back_on_error** - This command is used to request the decline and error responses be sent to the ret_addr on failed transactions. The value for this field should be "1".

  ```
  <input type="hidden" name="post_back_on_error" value="1">
  ```

  Please review all of the details of the error postback system.

- **save** - The value(s) for this command is/are the name of any of the fields (see the section detailing this [126]) that you would like tracked through our meta-data system.

  ```
  <input type="hidden" name="save" value="ordernum">
  ```

- **email_text...email_text10** - This/these command(s) can be used to add up to ten additional messages (up to 255 characters each) on the confirmation email generated by the gateway. This information will appear on both the customer and merchant emails.

```
<input type="hidden" name="email_text" value="Thanks for shopping with us today.">
<input type="hidden" name="email_text2" value="Please shop with us again!">
```

- **cust_id** - The value for this custom field can be a unique alpha-numeric identifier up to 40 characters assigned on the merchant's end for tracking purposes. It is a searchable field.

```
<input type="hidden" name="cust_id" value="ABC123">
```

- **cvv2_number** - This tag should be included if you would like your customers to enter the CVV security code from the back of their credit cards.

```
<input name="cvv2_number" size=”5”>
```

- **Optional Shipping Address Information** - Used to capture shipping information if different from the billing information.
  - sfname - The value for this is the first name of the person receiving the shipment - 50 maximum characters.
  - slname - The value for this is the last name of the person receiving the shipment - 50 maximum characters.
  - saddr - The value for this is the street address of the person receiving the shipment - 100 maximum characters.
  - scity - The value for this is the city of the person receiving the shipment - 25 maximum characters.
  - sstate - The value for this is the state of the person receiving the shipment - 25 maximum characters.
  - szip - The value for this is the postal code of the person receiving the shipment - 12 maximum characters.
  - sctry - The value for this is the country of the person receiving the shipment - 45 maximum characters.

```
<input name=”sfname” size=”15"/>
<input name=”slname” size=”15"/>
<input name=”saddr” size=”15"/>
<input name=”scity” size=”15"/>
<input name=”sstate” size=”15"/>
<input name=”szip” size=”15"/>
<input name=”sctry” size=”15"/>
```

- **check_num** - The value for this is the customer's check number. This field is NOT to be used if you, as the merchant, utilize our EFT system for check acceptance, but is optional for other check processing methods.

```
<input name="check_num" size=”5”>
```

- **check_memo** - This allows a merchant to print information to the Check Memo line on Check transactions. This field is NOT to be used if you, as the merchant, utilize our EFT or NACHA systems, it is optional for RediCheck printing services.

```
<input name="check_memo" size=”25”>
```

- **Recurring Transaction Commands**
  - recur_recipe - The value for this is the Recurring Recipe [61] that you have built in your Control Panel. This must always be paired with the recur_reps tag.
  - recur_reps - This numeric value is the number of times you would like a transaction to follow the recipe which programs the gateway to bill a card on an ongoing basis at the merchant's request. This must always be paired with the recur_recipe tag.
  - recur_total - This can be used if you are using the recurring billing feature to bill a card for a different amount than the initiating transaction. This feature can only be used in conjunction with recipes set for split amount recurring billing [63]. If you pass this, you must pass recur_desc.

- recur_desc - This tag allows a merchant to change the billing description which appears on the confirmation emails for recurring transactions. This feature can only be used in conjunction with recipes set for split amount recurring billing [63]. If you pass this, you must pass recur_total.

```
<input type="hidden" name="recur_recipe" value="monthly13">
<input type="hidden" name="recur_reps" value="6">
<input type="hidden" name="recur_total" value="50.00">
<input type="hidden" name="recur_desc" value="Enter a description here.">
```

- **ccswipe** - This is ONLY to be used if you have a compatible digital swipe reader. This HTML connection methods only support the input of track1 and track2 data combined.

```
<textarea name="ccswipe" rows=2 cols=40>
```

The swipe data will look like this:

```
%B5454545454545454^LASTNAME/FIRSTNAME^0102304000000567000000?;
5454545454545454=01023040000005678000?
```

For more information about the compatible USB swipers, please see this section [14].

- **Discount Commands**

  - Negative Value Item - The format is the same as a normal item. A **_qty*, **_desc*, and **_cost* must be passed for the negative item. The command is only to be used in conjunction with at least one non-negative value item with a positive total. This feature will subtract the amount of negative value item from the transaction total. The gateway will not allow a transaction to process with a negative total. A merchant should use the Post A Credit function if money needs to be credited to a customer's account. * indicates the item number; `1_desc`, `2_desc`, `3_desc`, etc. A field separator of underscore [`1_desc`] or dash [`1-desc`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_desc`, `item_2_desc`, etc. Only one naming convention per order form may be used.

```
<input type = "hidden" name="1_cost" value="100.00">
<input type = "hidden" name="1_desc" value="Item description">
<input type = "hidden" name="1_qty" value="1">
<input type = "hidden" name="2_cost" value="-25.00">
<input type = "hidden" name="2_desc" value="Discount description">
<input type = "hidden" name="2_qty" value="1">
```

The total to be charged to a customer's account in the above example is $75.00.

  - *_discount - The value of this field is the amount in dollars and cents which is amount of the discount given to the customer. This amount will be subtracted from the `*_cost` of the item it is being passed with. * indicates the item number; `1_discount`, `2_discount`, `3_discount`, etc. A field separator of underscore [`1_discount`] or dash [`1-discount`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_discount`, `item_2_discount`, etc. Only one naming convention per order form may be used.

```
<input type = "hidden" name="1_cost" value="100.00">
<input type = "hidden" name="1_desc" value="Item Desc">
<input type = "hidden" name="1_qty" value="1">
<input type = "hidden" name="1_discount" value="10.00">
```

The total to be charged to a customer's account in the above example is $90.00.

  - *_discount_percent - The value of this field is the percentage of the item amount being given to the customer. This percentage will be subtracted from the `*_cost` of the item it is being passed with. * indicates the item number; `1_discount_percent`, `2_discount_percent`, `3_discount_percent`, etc. A field separator of underscore [`1_discount_percent`] or dash [`1-discount_percent`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_discount_percent`, `item_2_discount_percent`, etc. Only one naming convention per order form may be used.

```
<input type = "hidden" name="1_cost" value="200.00">
<input type = "hidden" name="1_desc" value="Item Desc">
```

```
<input type = "hidden" name="1_qty" value="1">
<input type = "hidden" name="1_discount_percent" value="10.00">
```

The total to be charged to a customer's account in the above example is $180.00.

3.2. Split Form

3.2.1. Split Form - Introduction

Communicating transaction information with your gateway software is simple. If you do not have your own secure server, we suggest you utilize our Split Form protocol. The Split Form allows you to meet all of the necessary CISP/PCI Security requirements laid out by the credit card issuing banks for accepting credit cards through a gateway, without enduring the hassle and expense of meeting those requirements on your own server. A form post must be generated from your page that submits case sensitive fields to the second half of the Split Form that resides on our secure server.

For an English language *Split Form*, submit your post to

https://secure.itransact.com/cgi-bin/mas/split.cgi

For a Spanish language *Split Form*, submit your post to

https://secure.itransact.com/cgi-bin/mas/split.cgi?tpl_code=es

For a French language *Split Form*, submit your post to

https://secure.itransact.com/cgi-bin/mas/split.cgi?tpl_code=fr

Your post will generate the secure payment page. You can use the optional style settings or form tag commands to include images and colors to make the secure page look and feel more like your own site.

Example

Figure 3.33. Split Form - Payment Page Example

3.2.2. Split Form - Required Fields

Please pass these fields for all Split Form transactions.

- **mername** - The value is the name of your company.

- **vendor_id** - This value is your Order Form UID value or Gateway ID number. For security purposes, we suggest using the randomly generated UID value. This value can be up to 20 characters.
- **ret_addr** - This value is the HTML page or dynamic script that a customer and/or data is passed to after an order is submitted. Please remember that any images or information on this page must have the absolute addresses. In most cases, this page will be emulated in our secure server environment.
- **home_page** - The url of your website.
- **first_name** - This value is the customer's first name - 50 maximum characters.
- **last_name** - This value is the customer's last name - 50 maximum characters.
- **address** - This value is the cardholder's billing street address - 100 maximum characters.
- **city** - This value is the cardholder's city of their billing address - 25 maximum characters.
- **state** - This value is the cardholder's state of their billing address - If the transaction is foreign and there is no corresponding state, please enter a comma into this field - 25 maximum characters.
- **zip** - This value is either the five digit or nine digit billing postal code - If the transaction is foreign and there is no corresponding postal code, please enter a comma into this field - 12 maximum characters.
- **country** - This value is the cardholder's country of their billing address - 45 maximum characters.
- **phone** - This value is the customer's phone number - 25 maximum characters.
- **email** - This value is the customer's email address - 255 maximum characters.
- **Item Cost Methods** - There are two methods for setting the amount to be billed to the customer. One of these must be used with each split form transaction. If neither of these methods are included, the transaction will not be allowed.
  - Pre-Set Amount Method - Used when you have pre-set the amount of the transaction
    - *_desc - This value is the description of the order item. The * indicates the item number; `1_desc`, `2_desc`, `3_desc`, etc. A field separator of underscore [`1_desc`] or dash [`1-desc`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_desc`, `item_2_desc`, etc. Only one naming convention per order form may be used.
    - *_cost - This value is the numeric amount of the price of the item including dollars, decimal, and cents [XXX.xx]. The * indicates the item number; *1_cost*, *2_cost*, *3_cost*, etc. A field separator of underscore [`1_cost`] or dash [`1-cost`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_cost`, `item_2_cost`, etc. Only one naming convention per order form may be used.
    - *_qty - This value is the number of the item desired. * indicates the item number; `1_qty`, `2_qty`, `3_qty`, etc. A field separator of underscore [`1_qty`] or dash [`1-qty`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_qty`, `item_2_qty`, etc. Only one naming convention per order form may be used.
  - Amount Entry Method - Used when you allow your customers to enter the amount that they are paying or donating.
    - total_input - This value should be "1" if you want to allow the customer to enter their own payment amount.
    - total_input_desc - This value is the description of the order item. The localized version of transaction total will be used if not passed.

```
<HTML> <HEAD> <title> Order Form</title> </HEAD>
<BODY BGCOLOR="#FFFFFF">
<FORM Method="POST" Action="https://secure.itransact.com/cgi-bin/mas/split.cgi"
AUTOCOMPLETE = "off"><center> <BR> <P>
<input type="hidden" name="home_page" value="YourHomePage.com">
```

```
<input type="hidden" name="mername" value="Your Company">
<input type="hidden" name="ret_addr" value="http://www.YourReturnAddress.com">
<input type="hidden" name="vendor_id" value="XXXXX">
<table width=590> <TR> <TD> <P>
Select the item(s) below that you would like to order.
Your transaction will be confirmed by email.</P> <!/table>
<INPUT type="checkbox" name="1-desc" value="ORDER ITEM">
<B>ORDER ITEM</B> (&#036 1.99 each)<BR>
<INPUT type = "hidden" NAME="1-cost" value="1.99">
<INPUT type = "hidden" NAME="1-qty" value="1">
</TD></table>
<TABLE WIDTH=590> <td>
<strong><center><HR><font size=-1 color=blue>
<B>GENERAL INFORMATION</b></FONT><HR></strong></center></td>
</tr> </TABLE>
<TABLE WIDTH=590> <TR> <TD ALIGN="right">
<B>First Name: </B> </TD>
<TD><INPUT NAME="first_name" SIZE=15>
<B>Last Name: </B> <INPUT NAME="last_name" SIZE=15> <BR> </TD> </TR>
<TR> <TD ALIGN="right"><B>Address:</B> </TD>
<TD><INPUT NAME="address"SIZE=30><BR></TD> </TR>
<TR> <TD ALIGN="right"><B>City: </B></TD>
<TD><INPUT NAME="city" SIZE=15>
<B> State: </B> <INPUT NAME="state" SIZE=2MAXLENGTH=2>
<B> ZIP: </B> <INPUT NAME="zip" SIZE=10 maxlength=5><BR></TD> </TR>
<TR> <TD ALIGN="right"><B>Country:</B> </TD><TD><INPUT NAME="country"
SIZE=45 value="USA"><BR></TD> </TR>
<TR> <TD ALIGN="right"><B>Phone Number:</B> </TD>
<TD><strong><INPUT NAME="phone" SIZE=15><BR></TD> </TR>
<TR> <TD ALIGN="right">
<B>E-Mail Address: </B> </TD>
<TD><INPUT NAME="email" SIZE=30><BR></TD>
</TR> </TABLE> <!HR width=590></CENTER>
<P><CENTER>
<INPUT TYPE="submit" VALUE="Proceed to Secure Server"> </FORM> </BODY> </HTML>
```

3.2.3. Split Form - Optional Fields

These fields can be used to enhance your split forms, support additional data, and allow you to receive data back to your server after a transaction.

- ***_VARIABLE** - This value is an attribute of the order item. The * indicates the item number; `1_VARIABLE`, `2_VARIABLE`, `3_VARIABLE`, etc. "`VARIABLE`" can be anything. A field separator of underscore [`1_VARIABLE`] or dash [`1-VARIABLE`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_VARIABLE`, `item_2_VARIABLE`, etc. Only one naming convention per order form may be used.

  ```
  <input name="1_size" value="Large">
  <input name="1_color" value="White">
  ```

- **preauth** - Pass this command if you only want to pre-authorize a credit card to be charged at a later time.

  ```
  <input name="preauth"/>
  ```

- **ret_mode** - The value for this command is either `post` or `redirect`.

  ```
  <input type="hidden" name="ret_mode" value="redirect">
  ```

  or

  ```
  <input type="hidden" name="ret_mode" value="post">
  ```

  Please review all of the features of the Return Mode function [142].

- **lookup** - The value(s) of this command is/are the desired predetermined values of some of the required/non-required fields.

  ```
  <input type="hidden" name="lookup" value="first_name">
  ```

Please review all of the details of the Lookup function [143].

- **passback** - The value(s) of this command is/are the desired values of most of the non-*Lookup* fields.

```
<input type="hidden" name="ordernum" value="order#999">
<input type="hidden" name="passback" value="ordernum">
```

Please review all of the details of the Passback function [143].

- **post_back_on_error** - This command is used to request the decline and error responses be sent to the ret_addr on failed transactions. The value for this field should be "1".

```
<input type="hidden" name="post_back_on_error" value="1">
```

Please review all of the details of the error postback system [142].

- **save** - The value(s) for this command is/are the name of any of the fields (see the section detailing this [126]) that you would like tracked through our meta-data system.

```
<input type="hidden" name="save" value="ordernum">
```

- **email_text...email_text10** - This/these command(s) can be used to add up to ten additional messages (up to 255 characters each) on the confirmation email generated by the gateway. This information will appear on both the customer and merchant emails.

```
<input type="hidden" name="email_text" value="Thanks for shopping with us today.">
<input type="hidden" name="email_text2" value="Please shop with us again!">
```

- **cust_id** - The value for this custom field can be a unique alpha-numeric identifier up to 40 characters assigned on the merchant's end for tracking purposes. It is a searchable field.

```
<input type="hidden" name="cust_id" value="ABC123">
```

- **Formatting Commands** - These optional fields (including the form tags) will allow you to change the way the secure payment page will display. Many of these tags have corresponding settings [30] that can be controlled in the Account Settings [19]. Use of any of the formatting commands will circumvent the corresponding settings on your gateway.

  - showaddr - The value of this field must be “1” if you wish for the address information that was entered on the first half of Split Form to display.

    ```
    <input type="hidden" name="showaddr" value="1">
    ```

  - altaddr - The value for this is "1" if you would like the customer to enter a shipping address if different from the billing address.

    ```
    <input type="hidden" name="altaddr" value="1">
    ```

  - show_items - This displays order items on secure page instead of just the transaction total if you pass a value of “1”.

    ```
    <input type="hidden" name="show_items" value="1">
    ```

  - showcvv - The value of this field must be “1” if you would like to allow the customer to enter the CVV number printed on their credit card for verification purposes.

    ```
    <input type="hidden" name="showcvv" value="1">
    ```

  - nonum - Use a value of “1” to remove the check number field from the payment page for a check transaction.

    ```
    <input type="hidden" name="nonum" value="1">
    ```

  - mertext - This allows you to add a line of text at the bottom of the secure payment page.

    ```
    <input type="hidden" name="mertext" value="Thank you. Please come again!">
    ```

- Card Image Commands - Do not use for any card types you are not authorized to receive.
  - visaimage - If used, this value should be "1" if you are authorized to accept Visa if you want it to display, and "0" if you are not authorized for Visa acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="visaimage" value="0">
    ```

  - mcimage - This value should be "1" if you are authorized to accept MasterCard if you want it to display, and "0" if you are not authorized for MasterCard acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="mcimage" value="0">
    ```

  - ameximage - This value should be "1" if you are authorized to accept AMEX if you want it to display, and "0" if you are not authorized for AMEX acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="ameximage" value="0">
    ```

  - discimage - This value should be "1" if you are authorized to accept Discover if you want it to display, and "0" if you are not authorized for Discover acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="discimage" value="0">
    ```

  - dinerimage - This value should be "1" if you are authorized to accept Diners Card if you want it to display, and "0" if you are not authorized for Diners Card acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="dinerimage" value="0">
    ```

- Payment Type Commands - Do not use for any card types you are not authorized to receive. A payment type will only work correctly if you have been activated with the correct processor. If you would like to accept additional payment types, please contact our *support team.*
  - acceptcards - This value should be "1" if you want card acceptance fields to display.

    ```
    <input type="hidden" name="acceptcards" value="0">
    ```

  - acceptchecks - This value should be "1" if you want check acceptance fields to display.

    ```
    <input type="hidden" name="acceptchecks" value="0">
    ```

  - accepteft - This value should be "1" if you want EFT acceptance fields to display.

    ```
    <input type="hidden" name="accepteft" value="0">
    ```

  - disable_cards - This tag allows you to remove the card acceptance fields from the final check out page if you pass the value of “1”. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="disable_cards" value="1">
    ```

  - disable_checks - This tag allows you to remove the check acceptance fields from the final check out page if you pass the value of “1”.This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="disable_checks" value="1">
    ```

- Image Commands - You may submit images to us to host on our server.
  - background - This command allows your submission to call an image that you have uploaded to our server to display as the background of the final checkout page. The background image should be named like `12345bg.gif` (or .jpg), where “`12345`” is your Gateway ID. This command circumvents the setting in the gateway.

```
<INPUT type="hidden" name="background" value=
 "https://secure.itransact.com/images/background/12345bg.gif">
```

- Header Graphic Commands - These lines allow you to call an order form that displays your company's logo across the top of the final check out page. This same image will be used on Customer Billing Update page for recurring transactions if you use that feature. These commands circumvent the setting in the gateway.
  - mastimage - The value for this is “1”. This must always be paired with the mast tag.
  - mast - This command allows your script to call an image that you have uploaded to our server to display as the header graphic of the final checkout page. Maximum mast width is 800 pixels. This must always be paired with the mastimage tag. The header image should be named `12345mst.gif` (or .jpg), where “`12345`” is your Gateway ID. This command circumvents the setting in the gateway.

    ```
    <INPUT type="hidden" name="mastimage" value="1">
    <INPUT type="hidden" name="mast" value=
     "https://secure.itransact.com/images/mast/12345mst.gif">
    ```

- Color Commands - Change the coloring used in the template of your secure payment page. These commands circumvents the settings in the gateway.
  - bgcolor - This tag allows you to designate the background color of the final checkout page.

    ```
    <INPUT type="hidden" name="bgcolor" value="green">
    ```

  - fontcolor - This tag allows you to designate the color of the text on the final checkout page.

    ```
    <INPUT type="hidden" name="fontcolor" value="blue">
    ```

  - alink - This tag allows you to designate the color of an active link.

    ```
    <INPUT type="hidden" name="alink" value="red">
    ```

  - link - This tag allows you to designate the color of a link.

    ```
    <INPUT type="hidden" name="link" value="blue">
    ```

  - vlink - This tag allows you to designate the color of a visited link.

    ```
    <INPUT type="hidden" name="link" value="blue">
    ```

- **Recurring Transaction Commands**
  - recur_recipe - The value for this is the Recurring Recipe [61] that you have built in your Control Panel. This must always be paired with the recur_reps tag.
  - recur_reps - This numeric value is the number of times you would like a transaction to follow the recurring recipe which programs the gateway to bill a card on an ongoing basis at the merchant's request. This must always be paired with the recur_recipe tag.
  - recur_total - This can be used if you are using the recurring billing feature to bill a card for a different amount than the initiating transaction. This feature can only be used in conjunction with recipes set for split amount recurring billing [63]. If you pass this, you must pass recur_desc.
  - recur_desc - This tag allows a merchant to change the billing description which appears on the confirmation emails for recurring transactions. This feature can only be used in conjunction with recipes set for split amount recurring billing [63]. If you pass this, you must pass recur_total.

```
<input type="hidden" name="recur_recipe" value="monthly13">
<input type="hidden" name="recur_reps" value="6">
<input type="hidden" name="recur_total" value="50.00">
<input type="hidden" name="recur_desc" value="Enter a description here.">
```

- **Discount Commands**

- Negative Value Item - The format is the same as a normal item. A `*_qty`, `*_desc`, and `*_cost` must be passed for the negative item. The command is only to be used in conjunction with at least one non-negative value item with a positive total. This feature will subtract the amount of negative value item from the transaction total. The gateway will not allow a transaction to process with a negative total. A merchant should use the Post A Credit [57]function if money needs to be credited to a customer's account. * indicates the item number; `1_desc`, `2_desc`, `3_desc`, etc. A field separator of underscore [`1_desc`] or dash [`1-desc`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_desc`, `item_2_desc`, etc. Only one naming convention per order form may be used.

  ```
  <input type = "hidden" name="1_cost" value="100.00">
  <input type = "hidden" name="1_desc" value="Item description">
  <input type = "hidden" name="1_qty" value="1">
  <input type = "hidden" name="2_cost" value="-25.00">
  <input type = "hidden" name="2_desc" value="Discount description">
  <input type = "hidden" name="2_qty" value="1">
  ```

  The total to be charged to a customer's account in the above example is $75.00.

- *_discount - The value of this field is the amount in dollars and cents which is amount of the discount given to the customer. This amount will be subtracted from the `*_cost` of the item it is being passed with. * indicates the item number; `1_discount`, `2_discount`, `3_discount`, etc. A field separator of underscore [`1_discount`] or dash [`1-discount`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_discount`, `item_2_discount`, etc. Only one naming convention per order form may be used.

  ```
  <input type = "hidden" name="1_cost" value="100.00">
  <input type = "hidden" name="1_desc" value="Item Desc">
  <input type = "hidden" name="1_qty" value="1">
  <input type = "hidden" name="1_discount" value="10.00">
  ```

  The total to be charged to a customer's account in the above example is $90.00.

- *_discount_percent - The value of this field is the percentage of the item amount being given to the customer. This percentage will be subtracted from the `*_cost` of the item it is being passed with. * indicates the item number; `1_discount_percent`, `2_discount_percent`, `3_discount_percent`, etc. A field separator of underscore [`1_discount_percent`] or dash [`1-discount_percent`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_discount_percent`, `item_2_discount_percent`, etc. Only one naming convention per order form may be used.

  ```
  <input type = "hidden" name="1_cost" value="200.00">
  <input type = "hidden" name="1_desc" value="Item Desc">
  <input type = "hidden" name="1_qty" value="1">
  <input type = "hidden" name="1_discount_percent" value="10.00">
  ```

  The total to be charged to a customer's account in the above example is $180.00.

3.3. BuyNow Button

3.3.1. BuyNow Button - Introduction

The BuyNow commands can be used to generate a button for each item that you list on your website. Each button generates it's own secure payment page. This method is ideal if you do not have your own secure server and you only sell one type of item. The BuyNow commands can be used to generate a button for each item that you list on your website. Each button generates it's own secure payment page. This method is ideal if you do not have your own secure server and you only sell one type of item in any given order. A BuyNow button allows you to meet all of the necessary CISP/PCI Security requirements laid out by the credit card issuing banks for accepting credit cards through a gateway, without enduring the hassle and expense of meeting those requirements on your own server. Clicking the button generates a form post from your page that submits case sensitive fields to the secure payment page that resides on our secure server. Please remember that this feature will only allow for the purchase of any quantity of a single item. It does not operate like a shopping cart system which allows for the purchase of any quantity of multiple products.

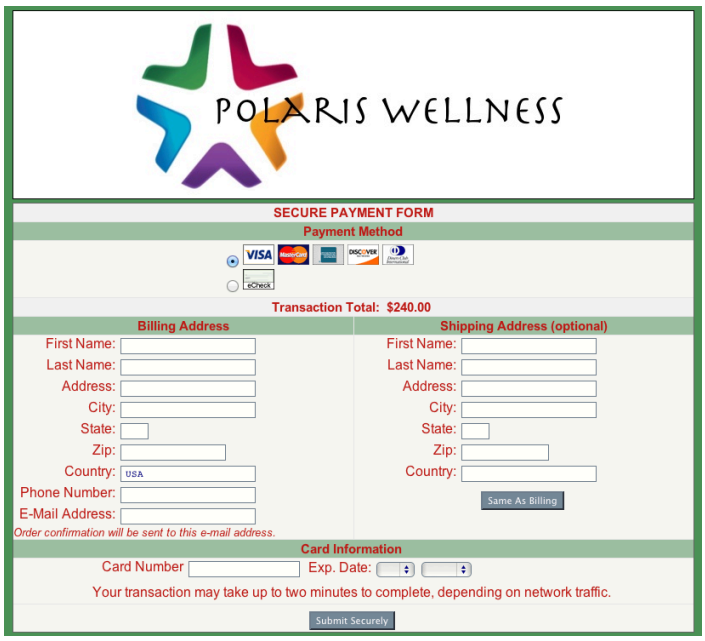
Posts must be submitted to

https://secure.itransact.com/cgi-bin/mas/buynow.cgi

Your post will generate the secure payment page. You can use the optional style settings or form tag commands to include images and colors to make the secure page look and feel more like your own site.

Example

Figure 3.34. BuyNow - Payment Page Example



3.3.2. BuyNow Button - Required Fields

Please pass these fields for all BuyNow transactions.

- **mername** - The value is the name of your company.
- **vendor_id** - This value is your Order Form UID value or Gateway ID number. For security purposes, we suggest using the randomly generated UID value. This value can be up to 20 characters.
- **ret_addr** - This value is the HTML page or dynamic script that a customer and/or data is passed to after an order is submitted. Please remember that any images or information on this page must have the absolute addresses. In most cases, this page will be emulated in our secure server environment.
- **show_addr** - The value of this field must be "1".
- **Item Cost Methods** - There are two methods for setting the amount to be billed to the customer. One of these must be used with each BuyNow transaction. If neither of these methods are included, the transaction will not be allowed.
  - Pre-Set Amount Method - Used when you have pre-set the amount of the transaction
    - *_desc - This value is the description of the order item. The * indicates the item number; `1_desc`, `2_desc`, `3_desc`, etc. A field separator of underscore [`1_desc`] or dash [`1-desc`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_desc`, `item_2_desc`, etc. Only one naming convention per order form may be used.
    - *_cost - This value is the numeric amount of the price of the item including dollars, decimal, and cents [`XXX.xx`]. The * indicates the item number; `1_cost`, `2_cost`, `3_cost`, etc. A field separator of underscore [`1_cost`] or dash [`1-cost`] may be used. In addition, if your system doesn't allow field names beginning

with a number, you may use `item_1_cost`, `item_2_cost`, etc. Only one naming convention per order form may be used.

- *_qty - This value is the number of the item desired. * indicates the item number; `1_qty`, `2_qty`, `3_qty`, etc. A field separator of underscore [`1_qty`] or dash [`1-qty`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_qty`, `item_2_qty`, etc. Only one naming convention per order form may be used.

- Amount Entry Method - Used when you allow your customers to enter the amount that they are paying or donating.
  - total_input - This value should be "1" if you want to allow the customer to enter their own payment amount.
  - total_input_desc - This value is the description of the order item. The localized version of transaction total will be used if not passed.

This example shows two BuyNow buttons.

```
<HTML> <HEAD><title>Secure Order Form</title></HEAD><BODY BGCOLOR="#FFFFFF"> <center> <BR>
<P> <table width=590> <TR> <TD> <P>
<font size=+2>S</font>elect the item below that you would like to order.
<P>You will then be taken to a SECURE server to enter your payment information.</P>
</table> <!center> <table border=0 cellpadding=7 width=500>
<td align=left><B>ORDER ITEM   &#036;1.00</B></FONT></TD></tr>
<tr> <td><font size=2>ORDER ITEM</font></td> </tr>
<tr> <td> <FORM METHOD="POST"
ACTION="https://secure.itransact.com/cgi-bin/mas/buynow.cgi">
<input type=submit value="Buy Now!"></td></FONT></TD>
<INPUT type=hidden name="vendor_id" value="XXXXX">
<INPUT type=hidden name="home_page" value="http://www.YourHomePage.com">
<INPUT type=hidden name="showaddr" value="1">
<input type=hidden name="1-desc" value="ORDER ITEM 1">
<input type=hidden name="1-cost" value="1.00">
<b>Quantity:</b><input type=text name="1-qty" size="3">
<INPUT type=hidden name="ret_addr" value="http://YourReturnAddress.com">
<INPUT type=hidden name="mername" value="Your Company.">
</form> </table> <!center>
<table border=0 cellpadding=7 width=500> <td align=left>
<B>ORDER ITEM2   &#036;2.22</B></FONT></TD></tr>
<tr> <td><font size=2>ORDER ITEM2</font></td> </tr>
<tr> <td> <FORM METHOD="POST"
ACTION="https://secure.itransact.com/cgi-bin/mas/buynow.cgi">
<input type=submit value="Buy Now!"></td></FONT></TD>
<INPUT type=hidden name="vendor_id" value="XXXXX">
<INPUT type=hidden name="home_page" value="http://www.merchant.com">
<INPUT type=hidden name="showaddr" value="1">
<input type=hidden name="1-desc" value="ORDER ITEM 2">
<input type=hidden name="1-cost" value="2.22">
<b>Quantity:</b><input type=text name="1-qty" size=3>
<INPUT type=hidden name="ret_addr"value="http://YourReturnAddress.com">
<INPUT type=hidden name="mername" value="Your Company">
</form> </table> </TD></table></BODY> </HTML>
```

3.3.3. BuyNow Button - Optional Fields

These fields can be used to enhance your forms, support additional data, and allow you to receive data back to your server after a transaction.

- ***_VARIABLE** - This value is an attribute of the order item. The * indicates the item number; `1_VARIABLE`, `2_VARIABLE`, `3_VARIABLE`, etc. ‘`VARIABLE`‘ can be anything. A field separator of underscore [`1_VARIABLE`] or dash [`1-VARIABLE`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_VARIABLE`, `item_2_VARIABLE`, etc. Only one naming convention per order form may be used.

```
<input name="1_size" value="Large">
<input name="1_color" value="White">
```

- **preauth** - Pass this command if you only want to pre-authorize a credit card to be charged at a later time.

```
<input name="preauth"/>
```

- **ret_mode** - The value for this command is either `post` or `redirect`.

```
<input type="hidden" name="ret_mode" value="redirect">
```

or

```
<input type="hidden" name="ret_mode" value="post">
```

Please review all of the features of the Return Mode [142] *[142]*function [142].

- **lookup** - The value(s) of this command is/are the desired predetermined values of some of the required/non-required fields.

```
<input type="hidden" name="lookup" value="first_name">
```

Please review all of the details of the Lookup function [143].

- **post_back_on_error** - This command is used to request the decline and error responses be sent to the *ret_addr* on failed transactions. The value for this field should be "1".

```
<input type="hidden" name="post_back_on_error" value="1">
```

Please review all of the details of the error postback system.

- **save** - The value(s) for this command is/are the name of any of the fields (see the section detailing this [126]) that you would like tracked through our meta-data system.

```
<input type="hidden" name="save" value="ordernum">
```

- **email_text...email_text10** - This/these command(s) can be used to add up to ten additional messages (up to 255 characters each) on the confirmation email generated by the gateway. This information will appear on both the customer and merchant emails.

```
<input type="hidden" name="email_text" value="Thanks for shopping with us today.">
<input type="hidden" name="email_text2" value="Please shop with us again!">
```

- **cust_id** - The value for this custom field can be a unique alpha-numeric identifier up to 40 characters assigned on the merchant's end for tracking purposes. It is a searchable field.

```
<input type="hidden" name="cust_id" value="ABC123">
```

- **Formatting Commands** - These optional fields (including the form tags) will allow you to change the way the secure payment page will display. Many of these tags have corresponding settings [30] that can be controlled in the Account Settings. Use of any of the formatting commands will circumvent the corresponding settings on your gateway.

  - showaddr - The value of this field must be "1".

    ```
    <input type="hidden" name="showaddr" value="1">
    ```

  - altaddr - The value for this is "1" if you would like the customer to enter a shipping address if different from the billing address.

    ```
    <input type="hidden" name="altaddr" value="1">
    ```

  - show_items - This displays order items on secure page instead of just the transaction total if you pass a value of "1".

    ```
    <input type="hidden" name="show_items" value="1">
    ```

  - showcvv - The value of this field must be "1" if you would like to allow the customer to enter the CVV number printed on their credit card for verification purposes.

```
<input type="hidden" name="showcvv" value="1">
```

- nonum - Use a value of “1” to remove the check number field from the payment page for a check transaction.

```
<input type="hidden" name="nonum" value="1">
```

- mertext - This allows you to add a line of text at the bottom of the secure payment page.

```
<input type="hidden" name="mertext" value="Thank you. Please come again!">
```

- Card Image Commands - Do not use for any card types you are not authorized to receive.
  - visaimage - If used, this value should be "1" if you are authorized to accept Visa if you want it to display, and "0" if you are not authorized for Visa acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="visaimage" value="0">
    ```

  - mcimage - This value should be "1" if you are authorized to accept MasterCard if you want it to display, and "0" if you are not authorized for MasterCard acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="mcimage" value="0">
    ```

  - ameximage - This value should be "1" if you are authorized to accept AMEX if you want it to display, and "0" if you are not authorized for AMEX acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="ameximage" value="0">
    ```

  - discimage - This value should be "1" if you are authorized to accept Discover if you want it to display, and "0" if you are not authorized for Discover acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="discimage" value="0">
    ```

  - dinerimage - This value should be "1" if you are authorized to accept Diners Card if you want it to display, and "0" if you are not authorized for Diners Card acceptance or you do not want it to display. This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="dinerimage" value="0">
    ```

- Payment Type Commands - Do not use for any card types you are not authorized to receive. A payment type will only work correctly if you have been activated with the correct processor. If you would like to accept additional payment types, please contact our *support team.*
  - acceptcards - This value should be "1" if you want card acceptance fields to display.

    ```
    <input type="hidden" name="acceptcards" value="0">
    ```

  - acceptchecks - This value should be "1" if you want check acceptance fields to display.

    ```
    <input type="hidden" name="acceptchecks" value="0">
    ```

  - accepteft - This value should be "1" if you want EFT acceptance fields to display.

    ```
    <input type="hidden" name="accepteft" value="0">
    ```

  - disable_cards - This tag allows you to remove the card acceptance fields from the final check out page if you pass the value of "1". This command circumvents the setting in the gateway.

    ```
    <input type="hidden" name="disable_cards" value="1">
    ```

  - disable_checks - This tag allows you to remove the check acceptance fields from the final check out page if you pass the value of "1".This command circumvents the setting in the gateway.

```
<input type="hidden" name="disable_checks" value="1">
```

- Image Commands - You may submit images to us to host on our server.
  - background - This command allows your submission to call an image that you have uploaded to our server to display as the background of the final checkout page. The background image should be named like `12345bg.gif` (or .jpg), where '`12345`' is your Gateway ID. This command circumvents the setting in the gateway.

```
<INPUT type="hidden" name="background"
value="https://secure.itransact.com/images/background/12345bg.gif">
```

  - Header Graphic Commands - These lines allow you to call an order form that displays your company's logo across the top of the final check out page. This same image will be used on Customer Billing Update [87] page for recurring transactions if you use that feature. These commands circumvent the setting in the gateway.
    - mastimage - The value for this is "1". This must always be paired with the mast tag.
    - mast - This command allows your script to call an image that you have uploaded to our server to display as the header graphic of the final checkout page. Maximum mast width is 800 pixels. This must always be paired with the mastimage tag. The header image should be named `12345mst.gif` (or .jpg), where '`12345`' is your Gateway ID. This command circumvents the setting in the gateway.

```
<INPUT type="hidden" name="mastimage" value="1">
<INPUT type="hidden" name="mast" value=
 "https:/secure.itransact.com/images/mast/12345mst.gif">
```

  - Color Commands - Change the coloring used in the template of your secure payment page. These commands circumvent the settings in the gateway.
    - bgcolor - This tag allows you to designate the background color of the final checkout page.

```
<INPUT type="hidden" name="bgcolor" value="green">
```

    - fontcolor - This tag allows you to designate the color of the text on the final checkout page.

```
<INPUT type="hidden" name="fontcolor" value="blue">
```

    - alink - This tag allows you to designate the color of an active link.

```
<INPUT type="hidden" name="alink" value="red">
```

    - link - This tag allows you to designate the color of a link.

```
<INPUT type="hidden" name="link" value="blue">
```

    - vlink - This tag allows you to designate the color of a visited link.

```
<INPUT type="hidden" name="link" value="blue">
```

- **Recurring Transaction Commands**
  - recur_recipe - The value for this is the Recurring Recipe [61] that you have built in your Control Panel. This must always be paired with the recur_reps tag.
  - recur_reps - This numeric value is the number of times you would like a transaction to follow the recipe which programs the gateway to bill a card on an ongoing basis at the merchant's request. This must always be paired with the recur_recipe tag.
  - recur_total - This can be used if you are using the recurring billing feature to bill a card for a different amount than the initiating transaction. This feature can only be used in conjunction with recipes set for split amount recurring billing [63]. If you pass this, you must pass recur_desc.

- recur_desc - This tag allows a merchant to change the billing description which appears on the confirmation emails for recurring transactions. This feature can only be used in conjunction with recipes set for split amount recurring billing [63]. If you pass this, you must pass recur_total.

```
<input type="hidden" name="recur_recipe" value="monthly13">
<input type="hidden" name="recur_reps" value="6">
<input type="hidden" name="recur_total" value="50.00">
<input type="hidden" name="recur_desc" value="Enter a description here.">
```

- **Discount Commands**

  - Negative Value Item - The format is the same as a normal item. A `*_qty`, `*_desc`, and `*_cost` must be passed for the negative item. The command is only to be used in conjunction with at least one non-negative value item with a positive total. This feature will subtract the amount of negative value item from the transaction total. The gateway will not allow a transaction to process with a negative total. A merchant should use the Post A Credit [57] function if money needs to be credited to a customer's account. * indicates the item number; `1_desc`, `2_desc`, `3_desc`, etc. A field separator of underscore [`1_desc`] or dash [`1-desc`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_desc`, `item_2_desc`, etc. Only one naming convention per order form may be used.

    ```
    <input type = "hidden" name="1_cost" value="100.00">
    <input type = "hidden" name="1_desc" value="Item description">
    <input type = "hidden" name="1_qty" value="1">
    <input type = "hidden" name="2_cost" value="-25.00">
    <input type = "hidden" name="2_desc" value="Discount description">
    <input type = "hidden" name="2_qty" value="1">
    ```

    The total to be charged to a customer's account in the above example is $75.00.

  - *_discount - The value of this field is the amount in dollars and cents which is amount of the discount given to the customer. This amount will be subtracted from the *_*cost* of the item it is being passed with. * indicates the item number; `1_discount`, `2_discount`, `3_discount`, etc. A field separator of underscore [1_discount] or dash [`1-discount`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_discount`, `item_2_discount`, etc. Only one naming convention per order form may be used.

    ```
    <input type = "hidden" name="1_cost" value="100.00">
    <input type = "hidden" name="1_desc" value="Item Desc">
    <input type = "hidden" name="1_qty" value="1">
    <input type = "hidden" name="1_discount" value="10.00">
    ```

    The total to be charged to a customer's account in the above example is $90.00.

  - *_discount_percent - The value of this field is the percentage of the item amount being given to the customer. This percentage will be subtracted from the *_*cost* of the item it is being passed with. * indicates the item number; `1_discount_percent`, `2_discount_percent`, `3_discount_percent`, etc. A field separator of underscore [`1_discount_percent`] or dash [`1-discount_percent`] may be used. In addition, if your system doesn't allow field names beginning with a number, you may use `item_1_discount_percent`, `item_2_discount_percent`, etc. Only one naming convention per order form may be used.

    ```
    <input type = "hidden" name="1_cost" value="200.00">
    <input type = "hidden" name="1_desc" value="Item Desc">
    <input type = "hidden" name="1_qty" value="1">
    <input type = "hidden" name="1_discount_percent" value="10.00">
    ```

    The total to be charged to a customer's account in the above example is $180.00.

3.4. Getting Data from Gateway

The gateway has methods for returning data to you after an HTML transaction. Those tools are the ret_mode [142], lookup [143], and passback [143] functions. You may use any or all of the methods for each transaction. The requested data will come back in a name/value pair string. The strings are always signed

with a PGP Signature [146] when either the lookup or passback function is used, so you can verify the response came from our server.

3.4.1. What is ret_mode?

The Return Mode function (ret_mode) enables you to specify how your customers are returned to your return address (ret_addr) after a successful transaction, as well as how transaction information is relayed to your script.

By default, customers are shown a “Thank You” page on the secure server after a successful transaction. When the customer clicks the `Continue` button, all name/value pairs are returned to your script using GET. However, the Return Mode function allows the customer to bypass the “Thank You” page and be directed to your ret_addr. This enables you to use the gateway services more seamlessly.

This function also has a feature enabling you to receive error messages and failures (declines) returned to a specified script on your server.

There are two values that can be used with the Return Mode Function. They are post and redirect. Each has a specific function. You can only use one method at a time.

3.4.1.1. Return Mode of Redirect

The redirect value of ret_mode value can be used if your ret_addr is a static HTML document. After a successful transaction, your customer is automatically redirected to your ret_addr, bypassing the “Thank You” page. Since this is a simple redirect, no passback or lookup values can be returned.

```
<input type="hidden" name="ret_mode" value="redirect">
```

3.4.1.2. Return Mode of POST

This method can only be used if the ret_addr page is a dynamic script/page. After a successful transaction, the information you have requested from the processing server will be posted to your ret_addr. All lookup and passback name/value pairs will be returned via the POST.

```
<input type="hidden" name="ret_mode" value="post">
```

3.4.1.3. POST of Error Messages

If you would like decline and error messages [178] also sent to your ret_addr, you may use this feature. This feature is very powerful, giving you complete control of the entire transaction process, because you can take the error message and display it in your own error screen.

```
<input type="hidden" name="post_back_on_error" value="1">
```

Responses will appear as follows:

- If a transaction is declined, you will receive a name/value pair of “err=Text message of the error” along with the name/value pairs for each passback and lookup variable requested.
- If an internal error is encountered, you will receive a name/value pair of “die=1”.
- If the transaction is successful, neither of these will appear.
- The responses of *err* and *die* will never be sent on the same transaction. A transaction will either be successful, an error, or a die.

3.4.1.4. Considerations and Restrictions

- When using ret_mode with the redirect option, your ret_addr page should reside on a secure server. If not, your customers will receive a browser security warning.
- When using ret_mode with the post option, your ret_addr URL should be on a secure (HTTPS) server. The page will be displayed securely as an emulated page in the secure server environment.

- When using ret_mode, your ret_addr must be an absolute URL, not a relative URL. (i.e. *http://www.yoursite.com/cgi-bin/return.cgi* is absolute. *.../cgi-bin/return.cgi* is relative.) In addition, all links located on your ret_addr page must also be absolute URLs.

- The ret_mode function uses a standard HTTP 1.1 POST.

- If your ASP or Cold Fusion application cannot "see" the incoming name/value pairs, you will need to consult your documentation/system admin.

- For security reasons, you must use a standard port for post-back data. (port 80 for HTTP or port 443 for HTTPS) In other words, a ret_addr of *http://www.yoursite.com:9876* will not work correctly.

- The Return Mode function is much more dynamic than the standard GET method used to return customers to your site. Please be aware that if your ret_addr is not available, unreadable, incomplete, times out, etc., your customer's account will have already been billed and they will receive an error message. Only use the Return Mode function if you are very familiar with CGI scripting. Always test your applications before making them available to your users.

3.4.2. How do I use passback?

This feature gives a merchant the ability to request and receive information that was submitted as a part of the order form post submission. This feature can be used to access any of the data that can not be retrieved via the lookup function. Fields reserved for the lookup function can not be requested using the passback function. The passback and lookup functions can both be used on the same form. The passback function enables you to include input variables in your order form that are passed back to your return address after the transaction is completed. The value for the ret_addr must be a dynamic page or script that can accept, parse, and interpret name/value pairs. In this example, the ret_addr field in the order form was set to *http://www.yoursite.com/cgi-bin/return.cgi*. This example uses two variables. They are named "fieldname1" and "ordernum". This example shows variables returned as part of the query string.

```
<INPUT type="hidden" name="passback" value="fieldname1">
<INPUT type="hidden" name="fieldname1" value="Ord">
<INPUT type="hidden" name="passback" value="ordernum">
<INPUT type="hidden" name="ordernum" value="99">
```

After a successful order has been completed, the customer is returned to the URL of the ret_addr. Any requested data will be sent as part of the query_string or as name/value pairs from an HTML POST. Orders are always signed with a PGP Signature [146] when either the lookup or passback function is used.

Example

```
http://www.site.com/cgi-bin/return.cgi?fieldname1=Ord&ordernum=99&signature=PGPSIGNATUREHERE
```

Considerations

- Field names used for the lookup function are reserved and may not be used as passback field names. However, the passback function may be used to request the contents of each description, cost, and quantity.

- Your ret_addr (return address) must be a script (such as Perl, PHP, ASP, CFM) that is capable of accepting and parsing variables posted from an HTML form.

- Each passback field must contain a value. If the field contains no value or an invalid value, you will receive a “nonexistent passback parameter” error.

3.4.3. How do I use lookup?

Don't be confused by the name of this function. This feature provides the same functionality of the passback function for 30 separate name/value pairs. The passback and lookup functions can both be used on the same form. The lookup function allows you to request specific customer data from the processing server at the time a transaction is processed. All data requested using the lookup function is sent directly to the return address (ret_addr)

specified in the order form. The value for the ret_addr must be a dynamic page or script that can accept, parse, and interpret name/value pairs. A merchant can request as many of the name/value pairs as needed.

The following fields can be received in the POST via the lookup function:

- **first_name** - Returns the value entered as the first_name.
- **last_name** - Returns the value entered as the last_name.
- **address** - Returns the value entered as the address.
- **city** - Returns the value entered as the city.
- **state** - Returns the value entered as the state.
- **zip** - Returns the value entered as the postal code.
- **country** - Returns the value entered as the country.
- **phone** - Returns the value entered as the phone.
- **email** - Returns the value entered as the email.
- **sfname** - Returns the value entered as the shipping first name.
- **slname** - Returns the value entered as the shipping last name.
- **saddr** - Returns the value entered as the shipping address.
- **scity** - Returns the value entered as the shipping city.
- **sstate** - Returns the value entered as the shipping state.
- **szip** - Returns the value entered as the shipping postal code.
- **sctry** - Returns the value entered as the shipping country.
- **authcode** - Returns the credit card authorization code.
- **cc_last_four** - Returns the last four digits of the card number.
- **ck_last_four** - Returns the last four digits of the checking account number.
- **cc_name** - Returns the name of the card type used. "Visa", for example.
- **total** - Returns the transaction total.
- **test_mode** - Returns "1" if your account is in test mode or "0" if it is not in test mode.
- **when** - Time/date stamp in format of "20010509134443" - meaning 05/09/2001 at 13:44:43.
- **xid** - Returns the transaction ID.
- **batch_number** - Returns the batch number for all transactions except EFT, since EFT transactions aren’t currently assigned batch numbers.
- **avs_response** - Returns the address verification response.
- **cvv2_response** - Returns the CVV response.

- **confemail** - Returns "1" if your account is set to send email confirmations to your customers or "0" if confirmations are not sent.
- **entry_method** - Returns either swipe or keyed.
- **email_text (email_text2 - email_text9)** - Returns the value(s) entered as the email_text field(s).
- **balance** - Returns the available credit line amount of the gift card/debit/prepaid if passed back from the processor.
- **authorized_amount** - Returns the partial amount that was approved for a retail swiped/keyed transaction.
- **billing_update_token** - If a transaction is initiating a recurring transaction tied to a recipe with Allow Customer Update [63] enabled, this is the value of the token to be used in a link to the secure billing update page. Should be 60-80 characters.

To use the *lookup* function, simply add the the commands to HTML code to your order form with the appropriate field name. In this example, the ret_addr field in the order form is *http://www.yourdomain.com/cgi-bin/return.cgi*. The request would look like this:

```
<INPUT type="hidden" name="lookup" value="first_name">
<INPUT type="hidden" name="lookup" value="authcode">
<INPUT type="hidden" name="lookup" value="total">
<INPUT type="hidden" name="lookup" value="xid">
<INPUT type="hidden" name="lookup" value="phone">
```

Assuming the value for the first_name was "Bob", the response would look like this:

```
http://www.yourdomain.com/cgi-bin/return.cgi?first_name=Bob&authcode=852346
&total=29.95&xid=987654&phone=8015551212&signature=PGPSIGNATUREHERE
```

Considerations

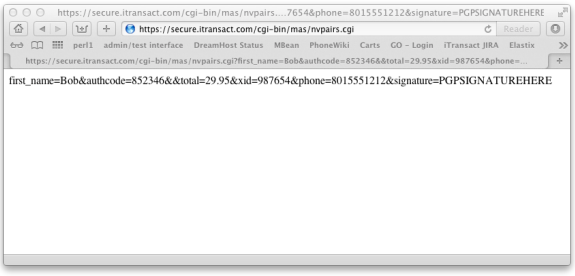
- As with any other dynamic web page, your ret_addr (return address) must be a CGI script or some other application, such as CFM or ASP that is capable of parsing the name/value pairs that are passed.
- Requested lookup fields that do not have a value will not be included in the information passed back.
- While in test mode, an authcode lookup will return "000000" and an XID lookup will return "9999999999".

3.5. What is nvpairs.cgi?

This script is hosted on the gateway's servers. When used as the value for ret_addr, this script displays all name/value pairs that were requested via lookup or passback as a URL-encoded string. This allows your script to parse a standard URL-encoded string to obtain returned values.

The address is *https://secure.itransact.com/cgi-bin/mas/nvpairs.cgi*

Figure 3.35. NVPairs Example



3.6. What is a PGP Signature?

One of the security features built into the transaction system is the use of a PGP signature. Each transaction confirmation is sent to the merchant's return address (ret_addr), signed with a PGP signature. This is a bullet-proof security feature that gives a merchant the knowledge that the transaction confirmation was sent by the processing server.

You will need a PGP application installed on your server that supports dynamic verification of an RSA signature. Find information about using PGP with Windows, Unix, Macintosh, Perl, Java, C++, and others at the *International PGP home page*.

You may obtain iTransact's Public Key and examples in the *toolkit*.

Considerations

- Transactions are only signed when either the passback [143] or lookup [143] function is used.
- As with any other dynamic web page, your ret_addr (return address) must be a CGI script or some other application, such as CFM or ASP that is capable of parsing the name/value pairs that are passed, including the signature.
- For security reasons, you should always pass a unique variable to the system using the passback function. This will cause the signature to have a unique value for each transaction.

3.7. Preventing Double HTML Submissions

The gateway has one method for preventing duplicated submissions of completed HTML transactions. However, the duplicate prevention account setting [25]only prevents transactions that have already processed successfully. Fortunately, there is a simple javascript that you can add to your forms to prevent double submissions at the point of the end user accidentally double clicking the submit button. To access that code, please see the *toolkit*.

4. Recurring Billing Integration

4.1. Overview

iTransact gateway offers one of the most robust recurring billing features available on the Internet. This is an ideal tool for merchants who bill according to a subscription or according to a set schedule. The recurring billing tool automates ongoing billings in a simple manner which gives a merchant as much control as if you were manually entering each of the transactions - without the hassle of manually entering the transactions. This is available for credit card and check transactions. If you do not need to initiate a charge at the time of entry, use a zero amount AVSOnly transaction type (if your processor has set your account to support that). Both the HTML [122] and the XML API [90] support the passing of recurring data in the transaction submissions. There are also methods for updating recurring transactions or turning them off. Additionally, the Control Panel interface provides methods for manually activating/deactivating recurring billings.

4.2. Considerations

There are some basic ideas that you need to understand when using our recurring billing system. Please keep these items in mind:

- A Recurring Recipe is the schedule which contains the instructions as to when a recurring transaction is billed. The Recurring Repetitions/Remaining Repetitions is the number of times that a transaction follows the recipe. Once a transaction is set as a recurring transaction, it will continue to follow the recipe until the number of repetitions cycles down to or is manually set to zero. The transaction on-hold setting will prevent future billings without changing the number of repetitions.
- Separate recipes do not need to be built for transactions following the same schedule, even if the transactions are initiated at different times, have different amounts, and different necessary repetitions. There is no limit to the number of transactions that can use the same recipe.

- There is no limit to the number of recipes that you can build.
- The recurring cycle begins each night at 12 Midnight, Mountain time. Any necessary modifications to a recurring transaction, or its recurring recipe, must be completed prior to 11:59 PM for it to be reflected as a part of the next day's recurring cycle. For instance, if it is January 31st and a recurring transaction is scheduled to process on February 1st, that transaction can be modified up to 11:59 PM on January 31st. In further explanation, if a merchant needed to set the remaining repetitions to zero to prevent future transactions, but missed the 11:59 PM deadline, the merchant would have access to change the number of remaining repetitions to zero. However, since the cycle had already begun, the transaction would still be billed. Future transactions would be prevented, but a refund or void would now be necessary because the transaction which the merchant had intended to stop was billed. The remaining repetitions in such a case would display as “-1”.
- When a transaction is initially submitted for processing, recurring details may be passed as part of the form that will automatically create future recurring charges, based on the details that you provide. In addition, you may also modify previously submitted transactions and mark them as recurring. This is done via the *Transaction Details* window.
- If you do not need to initiate a credit card charge at the time of entry (and your processor supports this type of transaction) use a zero amount AVSOnly transaction type. If your processor does not support AVSOnly transactions, you can use a pre-auth for a minimal amount.
- The calculations used to determine when a transaction will recur are based on the initial transaction date and when the transaction was set to recur. Please contact the *support team* for clarification.
- The largest allowed value for the Recurring Repetitions is “99999”.
- Recurring transaction details can be updated by you or by the card holders.

4.3. The Recipe Builder

The Recurring Transaction [60] interface is used to access the Recipe Builder where you can setup and edit Recurring Recipes. Follow the setup instructions here.

4.4. Setting Transactions to Recur

A transaction can be set to recur automatically at the time of the transaction or manually anytime there after. There are several methods to trigger recurring for a transaction.

4.4.1. Automatic Recurring Activation Using HTML

For any of the HTML methods, recurring transactions may be initiated at the time the original transaction is processed. To initially set a transaction as recurring, simply add the following input fields to your HTML order form. In this example we'll use the “monthly13” recipe and have the transaction recur six times, with a recurring total of $100.00 and a recurring description of “test2”

```
<input type="hidden" name="recur_recipe" value="monthly13">
<input type="hidden" name="recur_reps" value="6">
<!-- Below are Optional used with Split Recurring to set a different amount -->
<input type="hidden" name="recur_total" value="100.00">
<input type="hidden" name="recur_desc" value="test2">
```

4.4.2. Automatic Recurring Activation Using the API

For API transactions submitted with the AuthTransaction request, recurring transactions may be initiated at the time the original transaction is processed. To initially set a transaction as recurring, simply include the following fields to your XML query. In this example we'll use the “monthly13” recipe and have the transaction recur six times:

```
<RecurringData>
```

```
    <Recipe>monthly13</Recipe>
    <RemReps>6</RemReps>
        <!-- Can either supply OrderItems or Total and Description (but not both)
        to have recurring transactions run for a separate amount (Split Recurring) -->
          <OrderItems>
              <Item>
                <Description>test</Description>
                <Cost>10.00</Cost>
                <Qty>1</Qty>
               </Item>
          </OrderItems>
          <Total>10.00</Total>
          <Description>desc</Description>
</RecurringData>
```

4.4.3. Manual Recurring Activation for Existing Transactions

This manual activation method can be used for transactions that were submitted via HTML or XML. Once a sale transaction has been processed successfully, it can be set as a recurring transaction by following these simple steps:

1. Log into the Control Panel and open the Transaction Listing for the day when the original transaction was processed. Locate the transaction that needs to be set to recur and click on the XID number to open the Transaction Detail screen.

2. In the Transaction Detail screen, click on the `Go` button in the `Recur` column.

3. In the Recurring Details window, enter the number of repetitions (number of times a transaction needs to rebill) and choose the Recurring Recipe name from the drop down menu. If you need to change the amount to be billed or the items, uncheck the `Use Original Order Items` box and the editing and additional line tools will display. Once the necessary fields are completed, click the `Setup Recurring` button.

4. The Recurring Setup information page will display if all of the fields have been filled out correctly and the detail has been updated.

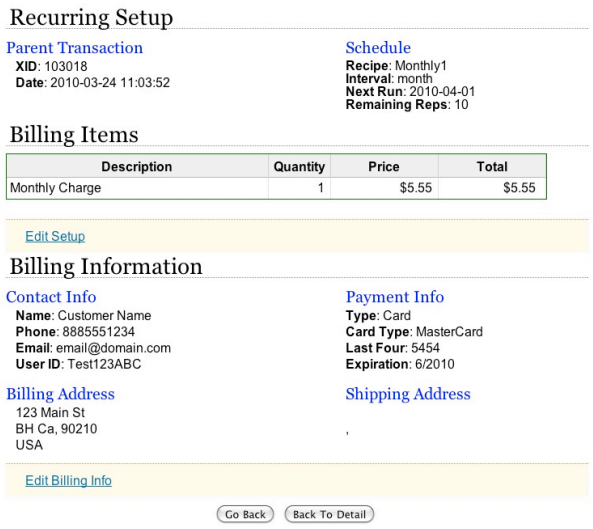
Figure 3.36. Recurring Setup Window

4.4.4. Real-Time Recurring Activation for Virtual Terminal Transactions

The recurring transactions can be submitted through the Standard Virtual Terminal [8]interface. Choosing this interface allows for the entry of multiple Order Items, as well as separate shipping and tax charges. Access the recurring billing information entry fields by clicking the `Recurring` toggle. Using this interface will charge a transaction in real-time, but will also schedule future transactions on that payment account. Enter the necessary values into the fields and the transaction will be billed and triggered to recur.

4.5. Understanding the Recurring Setup Window

Figure 3.37. Recurring Setup Window



This tools is used for manual setup and modification of recurring transactions. It is accessed by clicking the `Go` button for recurring in the Transaction Listing or the Transaction Detail.

- **Parent Transaction Data**
  - XID - This value is the numeric identifier assigned automatically to a transaction when it is submitted through the gateway system. This is the originating transaction and will be the parent transaction for all future recurring attempts.
  - Date - The time stamp of the original transaction attempt.
- **Schedule Data**
  - Recipe - This value is the name of the recurring recipe that is set to the transaction at that time.
  - Interval - This identifies the recipe type being used.
  - Next Run - A calculated value for the next recurring attempt.
  - Remaining Reps - The number of repetitions left that the transaction will continue to rebill until it cycles to zero or is set to zero.
- **Billing Items Data**
  - Description - The recurring item description.
  - Quantity - This number multiplied by the price value equals the total item amount.
  - Price - This is the cost of a single item. This number multiplied by the quantity value equals the total item amount.
  - Edit Setup - Click this link to be able to edit, modify, or add to the any of the billing items data.
- **Billing Information Data**
  - Contact Info - Information being used for recurring transactions.
  - Payment Info - Account information being used for recurring transactions.
  - Billing Address - The information submitted to the processor for AVS verification on recurring transactions.

- Shipping Address - The shipping address on file (not verified by the processor).
- Edit Billing Info - Click this link to be able to edit, modify, or add to the any of the billing information data.

4.6. Modifying, Suspending, or Canceling Recurring Transactions

The gateway provides several methods for modifying or canceling recurring transactions. Recurring information can be modified so that future recurring attempts will be made using updated information. Changes must be submitted prior to 11:59 PM Mountain time the day before the recurring is scheduled to process. Updating the data for a recurring transaction does not change information applied to the originating transaction itself, only future transactions. Resubmits using the originating transaction will use the original transaction information.

- **Modifying via the Control Panel**

  1. Log into the Control Panel and access the Transaction Detail window or open the Transaction Listing for the day when the original transaction was processed. Locate the transaction that needs to be modified and click on the XID number to open the Transaction Detail request screen.
  2. In the Transaction Detail screen, click on the `Go` button in the `Recur` column to open the Recurring Setup window.
  3. To edit the recipe, the remaining repetitions, or item amounts, click `Edit Setup`. Modify the information as necessary. Click into the `Repetitions` field, and edit or delete the value, and enter the new number of repetitions (set it to “0” to stop future recurrings). If you need to change the amount to be billed or the items, uncheck the `Use Current Items` box and editing and additional line tools will display. Once the necessary fields are completed, click the `Update Recurring Setup` button.
  4. If you need to edit contact information or account information, click the `Edit Billing Info`. Edit any necessary information and click the `Submit` button.

- **Modifying via the API**

  The RecurUpdate [107] request is used to update any of the data for a recurring transaction. A RecurUpdateResponse [113] will be generated.

- **Suspending a Transaction Using the Manual Hold Setting**

  Placing a transaction on a temporary hold to prevent billings is simple.

  1. Log into the Control Panel and access the Transaction Detail window or open the Transaction Listing for the day when the original transaction was processed. Locate the transaction that needs to be modified and click on the XID number to open the Transaction Detail request screen.
  2. In the Transaction Detail screen, click on the `No` link in the On Hold row of the Recurring Information area.
  3. A window will pop up to verify that you want to place the transaction on hold. Click `Ok`. The setting will change from `No` to `Yes`. This will temporarily prevent future recurrings (if done anytime prior to 11:59 PM Mountain time on the day before the next scheduled transaction) for as long as you deem necessary.
  4. The transaction can be taken off hold following the same steps as above (but by clicking on the `Yes` and toggling it to `No`).

- **Customer Recurring Update**

  If you have setup your recipe to support it, your customers can update their own payment information without you needing to do any manual entry yourself. The Customer Recurring Update tool will send out an email with a link that the customer can follow to a secure page where they can modify their card information. Additionally, you can choose to display a user specific link on an area of your site which the customer can follow to the same secure update page.

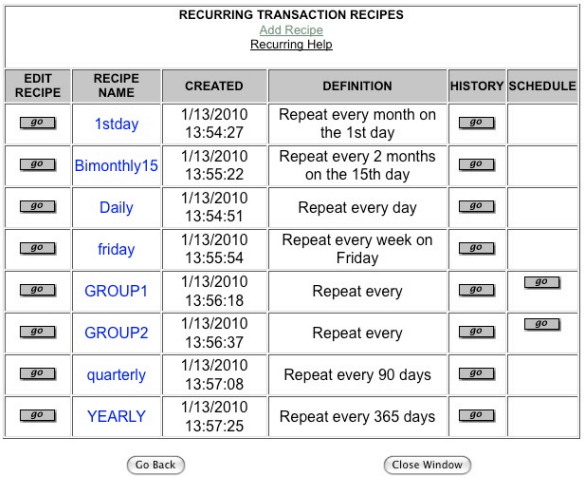
- **Canceling a Recurring Transaction**

  To cancel a recurring transaction, set the value for the repetitions to zero (“0”). You can submit that request through the Transaction Details, or through an API call.

4.7. Setting Scheduled-Type Transactions

Using a scheduled recipe allows you to run all of the transactions linked to a recipe at a date that can be controlled and scheduled manually using the scheduling tool. Merchants can access the scheduling tool in the Recurring Recipe list. In this example, the recipes named “Group 1” and “Group 2” are Scheduled-Type recurring recipes. To open the scheduling tool, click on the `Go` button in the Schedule column of the list.

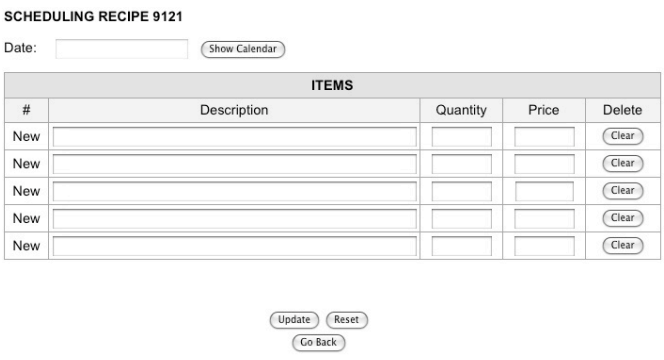
Figure 3.38. Recurring Recipe List



Follow these steps to set a group of transactions to run:

1. The schedule tool will open.

Figure 3.39. Schedule Tool



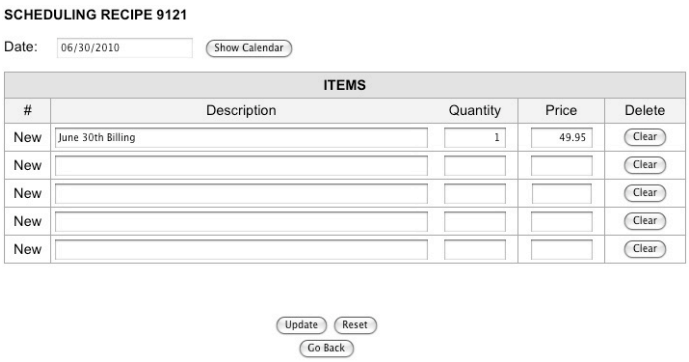
2. Click on the `Show Calendar` button to open the scheduling calendar.

Figure 3.40. Scheduling Calendar

3. Scroll to the correct month and year and then click on the appropriate day of the month and the `Date` field in the Scheduling Tool will be populated.

4. Enter the description, quantity and price into the appropriate fields.

Figure 3.41. Using the Scheduling Tool

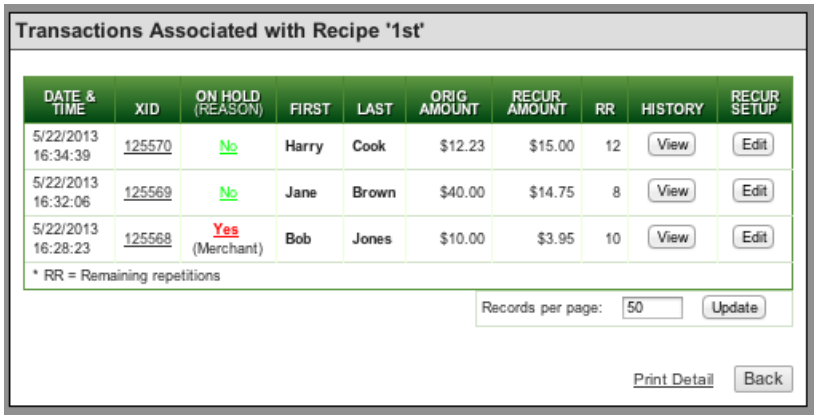


5. If any modifications need to be made, either click the `Clear` button in the Delete column to clear the row, or click the `Reset` button to clear the entire page.

4.8. Viewing Recurring Histories

To view all of the customers using a specific recipe, view the Recurring Recipe list, choose the recipe in question, and click the `Go` button in the History column.

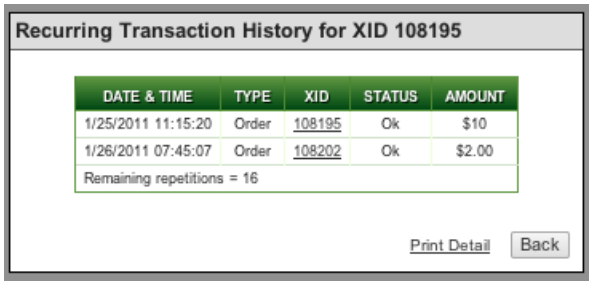
Figure 3.42. Recurring Recipe History



The `Yes` or `No` value in the On Hold column is a toggle switch that can be use to place a recurring billing on hold or vice versa. The red `Yes` indicates that a transaction is on hold and will not attempt. A `Yes` setting will also display "Merchant" or "Recur Fail" to specify the reason for the transaction being on hold. If the merchant manually sets a transaction on hold, it will list "Merchant". If "Recur Fail" displays, it means the transaction failed a recurring attempt, and the gateway is set to put a transaction on hold if the recurring attempt fails. The green `No` indicates that a transaction is not on hold and will attempt to bill according to the recipe.

To view all of the XIDs for a specific customer, click on the `View` button in the History column and a page listing the history for a specific customer will display. If you need to edit any of the information for that transaction, click the `Edit` button in the Recur Setup column.

Figure 3.43. Recurring Transaction History



4.9. Recurring PostBack Feature

Merchants using the recurring transaction features of the gateway may specify a URL to receive transaction post-back information after a recurring attempt. This can be enabled via the Account Settings in the Control Panel. To use this feature, enter the URL to be used for postback information. Then select the checkbox for recurring. Each time a recurring transaction is processed through the gateway system, the transaction server will post the transaction results to the designated postback URL. The fields listed below will be available when the postback function is used. Please note that all fields will be posted even if they do not have a value.

- **xid** - This is the value for the transaction ID assigned by the gateway.
- **authcode** - This is the authorization code issued by the credit card processor.
- **avs_response** - The value of this is the response received from the address verification system.
- **cc_last_four** - These are the last four digits of the account number.
- **cc_name** - This identifies the card type used.
- **cvv2_response** - The value of this is the response received from the card verification value system.
- **trans_type** - This should be listed as “order”.

- **when** - This is a time stamp in format of “20010509134443” (meaning 05/09/2001 at 13:44:43).
- **status** - This indicates the validity of a transaction. The following responses are possible: “ok”, “error”, “fail”, “begun”.
- **error_message** - If a transaction fails, this value will be the error response.
- **recipe_name** - This is the name of the recurring recipe being used for the transaction.
- **recipe_every** - This indicates the how often during a recurring period a transaction is to recur.
- **recipe_period** - The value for this is “day”, “week”, “month” or “scheduled”.
- **orig_xid** - This value is the transaction ID for the originating/operating/Parent transaction.
- **rem_reps** - This shows the number of times that a recurring transaction is set to cycle.
- **start_date** - This is the date of when the transaction was set as a recurring transaction.
- ***-desc** - This value is the description of the order item. * indicates the item number.
- ***-cost** - This value is the cost of the order item. * indicates the item number.
- ***-qty** - This value is the quantity of the order item. * indicates the item number.
- ***-X** - This value is an attribute of the order item. * indicates the item number. X is a user specified item attribute.
- **recur_desc** - This will have a value if the merchant has passed a recur_desc field or if the field was modified in the recurring system.
- **recur_total** - This will have a value if the merchant has passed a recur_total field or if the field was modified in the recurring system.
- **first_name** - This is the customer's first name.
- **last_name** - This is the customer's last name.
- **address** - This is the customer's billing address.
- **city** - This is the customer's billing city.
- **state** - This is the customer's billing state.
- **zip** - This is the customer's billing postal code.
- **ctry** - This is the customer's country.
- **email** - This is the email address entered by the customer.
- **phone** - This is the phone number submitted with the order.
- **phone** - This is the phone number submitted with the order.
- **saddr** - This is the shipping address.
- **scity** - This is the shipping city.
- **sctry** - This is the shipping country.
- **sfname** - This is the first name of the person which will receive the shipment.
- **slname** - This is the last name of the person which will receive the shipment.
- **sstate** - This is the shipping state.
- **szip** - This is the shipping postal code.
- **cust_id** - This is the merchant's optional custom id field passed with the transaction.

- **billing_update_token** - This parameter is included if the recipe has Allow Customer Update [63] enabled. It is the value of the token to be used in a link to the secure billing update page. Should be 60-80 characters.
- **on_hold** - This parameter is included if the recipe has Hold on Failure enabled. This will show a “1” for yes (it's been put on hold) or a “0” for no (it is no longer on hold).

4.10. Recurring On-Hold

The gateway allows merchants to put specific transactions “on hold” to temporarily prevent recurring billing attempts. This can be used automatically [63] or manually [153]. The Recurring On-Hold feature allows a merchant to toggle recurring on and off for specific transactions, without changing the number of remaining repetitions for that account. To place a transaction on hold, access the specific transaction (either in the Recurring History [152] interface [152] or in the Transaction Detail interface) and click on the On Hold toggle (which is a green `No`). A dialogue box will display verifying that you want to activate the hold for this transaction. Click `OK`. The setting will change from `No` to `Yes`. To take the transaction off hold, follow the same procedure (but the toggle will display a red `Yes`).

5. Form-Based Recurring Transaction Update

5.1. Introduction

This module provides a seamless way for a merchant's website to allow a customer to update their recurring billing information. Depending on the "on hold" status of the recurring transaction it will also catch a customer up by running a sale transaction. This reduces the amount of work that is required for a merchant to maintain their recurring transactions.

A form is hosted on the merchant's website which allows the customer to enter new billing information. This would most likely be done within a customer interface that requires a login. This form is posted directly to iTransact's site so that the merchant's servers never touch card information reducing PCI compliance issues. For added security, the credentials for this request includes the use of a payload signature.

This process appears seamless to customers because of the postback mechanism. All authenticated requests to the module result in a post to the merchant's website to render HTML to be displayed in the browser. This post is done inside of the request thread, and the results of the post are written to the active output stream. In addition to the required fields, passback [143] input fields can be provided so that the merchant's scripts can pass through session identifiers that will be returned in the postback.

5.2. Process Overview

5.2.1. Request Authentication

Requests are authenticated using a payload signature. For this module, the only request parameter used for the payload is xid. The reason for this is this module is intended to be used in an browser environment. Requiring any of the other data elements to be used in the payload signature would require the signature to be generated using Javascript or a background http call. This would add unnecessary complexity to the module.

The standard use scenario would be for a merchant to generate the payload signature while rendering the customer update form.

Authentication failures will not result in a postback attempt. This includes requests that have no authentication fields, have invalid postback URLs or that have invalid payload signatures. If authentication fails, a page will be rendered which indicates that there was an authentication failure. This page is non-branded and will simply notify the customer that their attempt did no go through.

5.2.2. Request Validation

Once the request is authenticated, the input fields are validated. If there are any field validation errors, a postback is generated to the error postback URL which includes field specific error information. This allows the postback script to render a form with specific field errors.

card_number, account_number, and cvv values should be validated client-side using JavaScript if possible since we will not postback these values. If invalid values result in an error postback, then the customer will have to enter these values again. There are several free JavaScript implementations of the Luhn 10 card validation algorithm. We have included some options below although we do not endorse or provide technical support for their use.

- *http://blog.planzero.org/2009/08/javascript-luhn-modulus-implementation/*
- *http://github.com/madrobby/creditcard_js*
- *http://javascript.internet.com/forms/val-credit-card.html*

5.2.3. Billing Info Validation/Recurring Authorization

If a request passes all validations, the module will then look at the "on hold" state of the recurring transaction. If the transaction is on hold, a Sale transaction will be attempted with the provided billing information. If the transaction is not on hold and card information is being provided, a $1.00 PreAuth or an AVSOnly transaction will be attempted which will validate the card information provided. If check information is being provided the module will not try any account authentication. The type of transaction used for cards will be dependent on whether AVSOnly is supported for the merchant's processing network. AVSOnly is the preferred method for this type of transaction and is the default. If the request results in a successful Sale transaction, the recurring transaction's "on hold" state is cleared and the number of remaining recurring repetitions is reduced by one.

If this transaction is not successful a postback is generated to the error postback URL. The presence of either internal_err or proc_err designates what type of error occurred. A processor error is normally displayed to the customer so that they have feedback about the problem with the card they attempted to use.

5.2.4. Email Notification

Depending on the merchant's recurring customer email setting, a confirmation email is sent to the customer for any update that is successful. The email clearly indicates whether or not a sale transaction took place. A confirmation email is always sent to the merchant's order email address on success.

5.2.5. Interface Postback

All authenticated requests to the module result in a post to the merchant's website to render HTML to be displayed in the browser. This post is done inside of the request thread, and the results of the post are written to the active output stream.

Currently, only one postback attempt is made. If this request fails, a page will be rendered which has basic information useful to the customer. This page is non-branded and will simply notify the customer that their attempt did/didn't go through and indicates that there was a system error. It is the responsibility of the merchant to ensure that their postback pages are up and functional. We recommend using a tool like *Nagios* or *Zenoss* to monitor your applications.

5.2.6. Background Postback

If an update is successful, an additional postback call can be made. This request is sent to the recurring postback URL specified in the merchant's control panel and is run in the background. It is only sent if the merchant has also enabled the recurring postback feature. It will be attempted every 10 minutes for an hour in case the postback address is unresponsive.

5.3. Module Interface

Request content should conform to the "application/x-www-form-urlencoded" content type. This means that the module only accepts POST requests. If a GET request is initiated against the model a 405 error will be returned with an "Allow" header with the value of "POST". The "content-type" header should also be specified in the request.

The Module URL is: *https://secure.itransact.com/merchant_api/billing_update*

5.4. Payload Signature Generation

5.4.1. Overview

To generate the payload_signature input parameter you are going to generate a string that looks like a HTTP parameter set and sign this string using HMAC-SHA1. You need to sign the xid. The HMAC-SHA1 algorithm generates a result that is not web friendly so you have to then Base64 encode it so it can be sent through the request.

5.4.2. Signature Process

For this section we are going to assume that we are going to issue a request against XID "99999999". We have been issued API credentials with a key of "12345678901234567890".

5.4.2.1. Generate The Payload

Note that although we are generating something that looks like part of a HTTP request URL we don't need to URI encode it. The request should be a string without any carriage return or newline characters.

```
xid=99999999
```

5.4.2.2. Sign The Payload

After Base64 encoding the output of our HMAC-SHA1 library we end up with a signature of "*QAnVUdQjREN2hbcsKAok7Ma7/SM=*"

5.5. Input Fields For Form-Based Recurring Billing Update Request

The address to POST these request to is *https://secure.itransact.com/merchant_api/billing_update*

These are the parameters to use in the POST to the iTransact gateway. All fields are required unless indicated as optional.

Table 3.34. Recurring Billing Update Input Fields

Value	Description	Billing Type	Required
aba	Nine digit numeric ABA routing number	Check	Yes (checks)
account_number	Bank account number	Check	Yes (checks)
account_source	Either "checking" or "savings". Default value is checking. This is not available for redicheck processing.	Check	No
account_type	Either "personal" or "business". This is valid only for CheckGateway processing	Check	No
addr	Customer address. Up to 100 characters	Card, Check	Yes
api_username	As displayed in the Gateway Control Panel's Account Settings	Card, Check	Yes
card_number	Credit card account. 14-16 digit numeric value must validate to Luhn algorithm	Card	Yes (cards)
city	Customer city. Up to 25 characters	Card, Check	Yes

Value	Description	Billing Type	Required
ctry	Customer country. Up to 45 characters	Card, Check	Yes
cvv	Cardholder verification value. 3-4 characters. OPTIONAL	Card	No
email	Must be a valid email address in standard email format. Up to 255 characters	Card, Check	Yes
error_ret_addr	Must be absolute https address used for Postback URL location for field validation, authentication errors or declined attempts.	Card, Check	Yes
exp_month	Credit card expiration month. Two digit numeric month value [can be 01...12].	Card	Yes (cards)
exp_year	Credit card expiration year. Four digit numeric year value.	Card	Yes (cards)
first_name	Customer first name. Up to 50 characters.	Card, Check	Yes
last_name	Customer last name. Up to 50 characters.	Card, Check	Yes
passback[]	This field allows you to pass additional custom fields that will come through in the return post. Include the open and closed brackets [] characters with each passback request. There should be one input field named "passback[]" for every input field that you want to have returned to your application. See the passback example below.	Card, Check	No
payload_signature	HMAC-SHA1 signature of the XID parameter	Check	Yes
state	Cardholder state. Up to 25 characters	Card, Check	Yes
sec_code	Standard Entry Category. 3 characters	Check	No
success_ret_addr	Must be absolute https address used for Postback URL location for successful updates.	Card, Check	Yes
xid	The transaction ID (originating or Parent XID) tied to the recurring transaction.	Card, Check	Yes

Value	Description	Billing Type	Required
zip	Customer postal code. Up to 20 characters	Card, Check	Yes

Passback[] Examples

```
<input type="hidden" name="myfield1" value="data1">
<input type="hidden" name="myfield2" value="data2">
<input type="hidden" name="passback[]" value="myfield1">
<input type="hidden" name="passback[]" value="myfield2">
```

The following is an example of a request being generated from an HTML page.

Form-Based Recurring Billing Update Example - Card

```
<form method=post action="https://secure.itransact.com/merchant_api/billing_update">
<!-- API Key for example is: 9pMx5z2G246W5vwSZ9Et -->
<P>
API Username <input type=text name=api_username value="testaccount_XXXXXXXX">
<P>
Payload Sig <input type=text name=payload_signature value="PUly2kg117rM1+fsEH1mFoZxOe8=">
<P>
XID <input type=text name=xid value="101166">
<P>
Error Return Address
<input type=text name=error_ret_addr value="https://www.domain.com/errscript.cgi">
<P>
Success Return Address
<input type=text name=success_ret_addr value="https://www.domain.com/script.cgi">
<p>
First Name <input type=text name=first_name value="John">
<p>
Last Name <input type=text name=last_name value="Smith">
<p>
Email <input type=text name=email value="email@domain.com">
<p>
Address <input type=text name=addr value="123 Main ST">
<p>
City <input type=text name=city value="Bountiful">
<p>
State <input type=text name=state value="UT">
<p>
Zip <input type=text name=zip value="84010">
<p>
Country <input type=text name=ctry value="USA">
<p>
Card Number <input type=text name=card_number value="5454545454545454">
<p>
Exp Month <input type=text name=exp_month value="11">
<p>
Exp Year <input type=text name=exp_year value="2020">
<p>
CVV <input type=text name=cvv value="123">
</P>
<br>
<input type=submit name=go value="Process">
</center>
</form>
```

Form-Based Recurring Billing Update Example - Check

```
<form method=post action="https://secure.itransact.com/merchant_api/billing_update">
<!-- API Key for example is: 9pMx5z2G246W5vwSZ9Et -->
<P>
API Username <input type=text name=api_username value="testaccount_XXXXXXXX">
<P>
Payload Sig <input type=text name=payload_signature value="PUly2kg117rM1+fsEH1mFoZxOe8=">
<P>
```

```
XID <input type=text name=xid value="101166">
<P>
Error Return Address
<input type=text name=error_ret_addr value="https://www.domain.com/errscript.cgi">
<P>
Success Return Address
<input type=text name=success_ret_addr value="https://www.domain.com/script.cgi">
<p>
First Name <input type=text name=first_name value="John">
<p>
Last Name <input type=text name=last_name value="Smith">
<p>
Email <input type=text name=email value="email@domain.com">
<p>
Address <input type=text name=addr value="123 Main ST">
<p>
City <input type=text name=city value="Bountiful">
<p>
State <input type=text name=state value="UT">
<p>
Zip <input type=text name=zip value="84010">
<p>
Country <input type=text name=ctry value="USA">
<p>
ABA <input type=text name=aba value="124000054">
<p>
Account Number <input type=text name=account_number value="12345">
<br>
<input type=submit name=go value="Process">
</center>
</form>
```

5.6. Recurring Billing Update Responses

5.6.1. Recurring Billing Update Response Parameters

These are the parameters that can be included in a recurring billing update response.

Table 3.35. Standard Recurring Billing Update Response Fields

Value	Description	Billing Type	Responses
aba	Nine digit numeric ABA routing number	Check	All
account_source	Either "checking" or "savings". Default value is checking. This is not available for redicheck processing.	Check	All
account_type	Either "personal" or "business". This is valid only for CheckGateway processing	Check	All
addr	Customer address. Up to 100 characters	Card, Check	All
authcode	The approval number issued by card processor/issuer. Up to 8 alpha-numeric characters	Card	Success
avs_category	Address verification response category. See table below for possible return values.	Card	Success

Value	Description	Billing Type	Responses
avs_response	The raw address verification response code from processor.	Card	Success
card_name	The credit card type submitted.	Card	All (If determinable)
city	Customer city. Up to 25 characters	Card, Check	All
ctry	Customer country. Up to 45 characters	Card, Check	All
cvv_response	The CVV verification response from processor.	Card	Success
email	Account holder email address in standard email format. Up to 255 characters	Card, Check	All
err_message	This will be included in any post to the error_ret_addr for any validation, processor, or auth type errors.	Card, Check	Error
exp_month	Credit card expiration month. Two digit numeric month value [can be 01...12].	Card	All
exp_year	Credit card expiration year. Four digit numeric year value.	Card	All
first_name	Customer first name. Up to 50 characters.	Card, Check	All
internal_err	If an internal error occurred while processing the transaction, this value will be "1". It will be blank otherwise.	Card, Check	All
last_four	The last four digits of the credit card account submitted.	Card	All
last_name	Customer last name. Up to 50 characters.	Card, Check	All
Passback[] Fields	Any values passed as passback[] variables will be included	Card, Check	All
payload_signature	HMAC-SHA1 signature of the all included parameters signed using merchant's API Key.	Card,Check	All
proc_err	If a processor type error takes place, this value will be "1". It will be blank otherwise.	Card,Check	All
sec_code	Standard Entry Category. 3 characters	Check	All

Value	Description	Billing Type	Responses
state	Cardholder state. Up to 25 characters	Card, Check	All
total	The dollar amount of the sale transaction or validation. If a preauth was run then this would be "1.00". If an AVS Only was run then the value will be "0.00".	Card, Check	All
trans_xid	The Transaction ID of the transaction that was run during the update.	Card, Check	All (If determinable)
validation_err	If validation type error takes place, this value will be "1". It will be blank otherwise. If equal to "1", there will be a Field Validation Error Indicator included in the response.	Card, Check	All
xid	The transaction ID (originating or Parent XID) tied to the recurring transaction.	Card, Check	All
zip	Account holder postal code. Up to 20 characters	Card, Check	All

These are the potential values for avs_category that may be included in a Recurring Billing Update Response for a credit card update.

Table 3.36. Potential AVS_Category Values - Recurring Billing Update Response Values

Value	Description
address	Street address matched
address_postal	Street address and postal code matched
address_zip5	Street address and five digit postal code matched
address_zip9	Street address and nine digit postal code matched
address_ok_postal_format_error	Street address matched, postal code format error
global_non_participant	International with no AVS support
international_address_not_verified	International with no AVS support
no_match	No street address or postal code match
no_response	No response
not_allowed	Not allowed
postal	Postal code matched
postal_ok_address_format_error	Postal code matched, street address format error
service_not_supported	AVS service not supported for card
unavailable	AVS service not supported for card
zip5	Five digit postal code matched
zip9	Nine digit postal code matched

These are the field validation response parameters than may be included in a Recurring Billing Update Response.

Table 3.37. Field Validation Error Indicators - Recurring Billing Update Response Fields

Field	Description	Description
aba_err	Indicates a validation error with the aba field.	Check
account_number_v_err	Indicates a validation error with the account_number field.	Check
account_source_v_err	Indicates a validation error with the account source field.	Check
account_type_v_err	Indicates a validation error with the account type field.	Check
addr_v_err	Indicates a validation error with the address field.	Card, Check
card_number_v_err	Indicates a validation error with the credit card account field.	Card
city_v_err	Indicates a validation error with the city field.	Card, Check
ctry_v_err	Indicates a validation error with the country field.	Card, Check
cvv_v_err	Indicates a validation error with the CVV field.	Card
email_v_err	Indicates a validation error with the email field.	Card, Check
exp_month_v_err	Indicates a validation error with the expiration month field.	Card
exp_year_v_err	Indicates a validation error with the expiration year field.	Card
first_name_v_err	Indicates a validation error with the first name field.	Card, Check
last_name_v_err	Indicates a validation error with the last name field.	Card, Check
sec_code_v_err	Indicates a validation error with the SEC code field.	Check
state_v_err	Indicates a validation error with the state field.	Card, Check
success_ret_addr_v_err	Indicates a validation error with the success_ret_addr field.	Card, Check
xid_v_err	Indicates a validation error with the transaction ID field.	Card, Check
zip_v_err	Indicates a validation error with the postal code field.	Card, Check

6. Reporting API

6.1. Introduction

The same tool that generates the merchant reports available from the merchant control panel may also be accessed externally, through our Reporting API. This allows you to automate the process of downloading either CSV or XML transaction reports.

6.2. Authentication

Authentication is handled using the combination of a username and a HMAC digest. A HMAC digest verifies both the authenticity and data integrity for a request payload. Our implementation is based on HMAC-SHA-1 using a 160 bit key. You can read more about HMAC at Wikipedia [*http://en.wikipedia.org/wiki/HMAC*] or for the adventurous the RFC [*http://www.faqs.org/rfcs/rfc2104.html*] is available. By using a signature based authentication scheme we can validate who generated an API request and ensure that the request was not altered using a man-in-the-middle attack [*http://en.wikipedia.org/wiki/Man-in-the-middle_attack*]. To access the reporting tool you will sign a string containing all the request parameters you are posting to the report and then pass that in along with your API username.

NOTE:*The Reseller API Credentials can not be used with these requests.*

6.3. Request Parameters

The integration with the reporting application is done through a form post. There are seven required parameters that must be included with every request and there are also optional parameters that let you narrow down your result set. The URL of the report application is

https://secure.itransact.com/jfe/translist/list.do

NOTE: *The Reseller API Credentials can not be used with these requests.*

6.3.1. Required Parameters

The following request parameters must be submitted with every request.

- **apiUsername** - This is the API username that has been assigned to you.
- **outputFormat** - Report Format. Valid input values are:

  list-csv

  list-xml
- **payloadSignature** - This is the payloadSignature that you have generated using any other request parameters. Review complete documentation to see the specifics of how this is generated.
- **transDateFirst** - This is the start date for the set of transactions you are trying to access. The format is *yyyy/MM/dd*.
- **transDateLast** - This is the end date for the set of transactions you are trying to access. The format is *yyyy/MM/dd*.
- **transTimeFirst** - This is the start time for the set of transactions you are trying to access. The format is *HH:mm.*
- **transTimeLast** - This is the end time for the set of transactions you are trying to access. The format is *HH:mm.*

6.3.2. Optional Parameters

The following parameters are optional search parameters. These search parameters do not result in full text searches, so they will only match records that have identical field values.

- **action** - Transaction Type. The valid action inputs are:

  order - Order which includes Sale, PreAuth and Force transactions. Use the subaction input to narrow down the result to a specific type.

  credit - Credit

  void - Void

  postauth - PostAuth

- **auth** - Authorization Code
- **avsCategory** - AVS Response Category
- **avsResponse** - AVS Response Code
- **batchNumber** - Batch Number
- **card** - Card Type. This input field can contain a single or comma separated list of card types to include in the results. The valid card types are:

  1 - American Express

  3 - Discover

  4 - Diners/Carte Blanche

  7 - Mastercard

  8 - Visa
- **cvv2Response** - CVV Response Code
- **email** - Customer email
- **extUserid** - Merchant supplied user ID (the *cust_id* field)
- **haddr** - Billing Address
- **hcity** - Billing City
- **hctry** - Billing Country
- **hfname** - Billing First Name
- **hlname** - Billing Last Name
- **hstate** - Billing State
- **hzip** - Billing Postal Code
- **instr** - Payment Instrument. Valid *instr* inputs are:

  cc - Credit Card

  ck - Check

  eft - EFT
- **ip** - Transaction IP Address Source
- **lastFour** - Last four digits of the account number
- **parentXid** - Parent transaction ID
- **phone** - Customer phone
- **recurRecipe** - Recurring recipe name
- **recurStatus** - Include only recurring transactions with this status. Status values are the same as the status input parameter.
- **recurXid** - Include only recurring transactions that have recurXid as their parent XID.
- **saddr** - Shipping Address
- **scity** - Shipping City

- **sfname** - Shipping First Name
- **slname** - Shipping Last Name
- **sstate** - Shipping State
- **szip** - Shipping Postal Code
- **status** - Transaction Status. Valid *status* values are:

  ok - Transaction Completed Successfully

  error - Transaction was refused by the processing network

  fail - The transaction failed for an unexpected reason

  avs_fail - The AVS response did not satisfy the merchant's AVS requirements at the time the transaction was run

  unknown - An unknown error occurred

  begun - Transaction did not complete which usually indicates an internal error
- **subAction** - Used to narrow down the result set to one type of action. Valid input values are:

  force - Force Transaction

  preauth - Preauth Transaction

  sale - Sale Transaction
- **total** - Transaction total in the format DDDD.cc
- **transSource** - Transaction source. Valid input values are:

  av - AutoVoid

  html - Web Form

  phone - Phone

  recur - Recurring

  session - Virtual Terminal

  xml - XML
- **xidFirst** - This is the opening parameter that allows you to pull a range of specified transaction IDs.
- **xidLast** - This is the closing parameter that allows you to pull a range of specified transaction IDs.

6.4. Generating the payloadSignature Value

The same tool that generates the merchant reports available from the merchant control panel may also be accessed externally though our reporting tool. This allows you to automate the process of downloading either CSV or XML transaction reports.

6.4.1. Overview

To generate the payloadSignature input parameter you are going to generate a string that looks like a HTTP parameter set and sign this string using HMAC-SHA1. You include all the parameters you are going to pass through except for payloadSignature. This parameter set string also needs to be generated in alphabetical order. The HMAC-SHA1 algorithm generates a result that is not web friendly so you have to then Base64 encode it so it can be sent through the request.

6.4.2. Signature Process

For this section, assume that we are going to issue a request to search for all credit card transactions between "10/1/2007" and "10/31/2007" and we have been issued API credentials with a username of test with a key of "12345678901234567890".

6.4.3. Assemble the Request Parameters

To emphasize the point that we are going to sort input parameters alphabetically in the signature, the *instr* field is listed out of order:

apiUsername - test

outputFormat - list-xml

transDateFirst - 2007/10/01

transDateLast - 2007/10/31

transTimeFirst - 00:00

transTimeLast - 23:59

instr - cc

6.4.4. Generate The Payload

Note that although we are generating something that looks like part of a HTTP request URL we don't need to URI encode it. The example below shows up in two lines in this document for formatting purposes but would actually be a string without any carriage return or newline characters.

```
apiUsername=test&instr=cc&outputFormat=list-xml&transDateFirst=2007/10/01
&transDateLast=2007/10/31&transTimeFirst=00:00&transTimeLast=23:59
```

6.4.5. Sign the Payload

After Base64 encoding the output of our HMAC-SHA1 library we end up with a signature of

cyeL36oyyManvmmJWBHZJv3Z1bE=

Remember, each transaction will have a unique payload. These examples are simply for testing. Do not try to use this payload or the generated payloadSignature for your real transactions. It will not work.

6.4.5.1. Perl HMAC Signature Example

```
my $EXPECTED_SIGNATURE = 'cyeL36oyyManvmmJWBHZJv3Z1bE=';
my $payload = "apiUsername=test&instr=cc&outputFormat=list-xml&"
  . "transDateFirst=2007/10/01&transDateLast=2007/10/31&"
  . "transTimeFirst=00:00&transTimeLast=23:59";
use Digest::HMAC_SHA1;
my $hmac = Digest::HMAC_SHA1->new("12345678901234567890");
$hmac->add($payload);
# The Perl Digest lib doesn't provide the trailing '=' character
my $actual_signature = $hmac->b64digest . '=';
if ($EXPECTED_SIGNATURE eq $actual_signature) {
  print "Success!\n";
  print "Signature: '$actual_signature'\n";
} else {
  print "Failure!\n";
  print "Expected Signature: '$EXPECTED_SIGNATURE'\n";
  print "  Actual Signature: '$actual_signature'\n";
}
```

6.4.5.2. Java HMAC Signature Example

```
import java.util.*;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
```

```
import org.apache.commons.codec.binary.Base64;
public class HMACSignature {
  private static final String EXPECTED_SIGNATURE = "cyeL36oyyManvmmJWBHZJv3Z1bE=";
  public static void main(String[] args) throws Exception {
    java.security.Security.addProvider(new com.sun.crypto.provider.SunJCE());
    SecretKeySpec hmac = new SecretKeySpec( "12345678901234567890".getBytes(),
     "HmacSHA1" );
    Mac mac = Mac.getInstance( hmac.getAlgorithm() );
    mac.init( hmac );
    String sigLoad = "apiUsername=test&instr=cc&outputFormat=list-xml&" +
      "transDateFirst=2007/10/01&transDateLast=2007/10/31&transTimeFirst=00:00&" +
      "transTimeLast=23:59";
    byte[] digest = mac.doFinal(sigLoad.getBytes());
    String actualSignature = new String(Base64.encodeBase64(digest));
    if (EXPECTED_SIGNATURE.equals(actualSignature)) {
      System.out.println("Success!");
      System.out.println("Signature: " + actualSignature);
    }
    else {
      System.out.println("Failure!");
      System.out.println("Expected Signature: " + EXPECTED_SIGNATURE);
      System.out.println("  Actual Signature: " + actualSignature);
    }
  }
}
```

6.4.5.3. Ruby HMAC Signature Example

```
require 'openssl'
require 'base64'
EXPECTED_SIGNATURE = 'cyeL36oyyManvmmJWBHZJv3Z1bE='
payload = "apiUsername=test&instr=cc&outputFormat=list-xml&" +
  "transDateFirst=2007/10/01&transDateLast=2007/10/31&" +
  "transTimeFirst=00:00&transTimeLast=23:59"
key = "12345678901234567890"
digest=OpenSSL::HMAC.digest(OpenSSL::Digest::SHA1.new(key), key, payload)
actual_signature = Base64.b64encode(digest)
# For some reason, we end up with a new line character in the actual_signature...
actual_signature.chomp!
if EXPECTED_SIGNATURE.eql?(actual_signature)
  puts 'Success!'
  puts 'Signature: ' + actual_signature
else
  puts 'Failure!'
  puts 'Expected Signature: ' + EXPECTED_SIGNATURE
  puts '  Actual Signature: ' + actual_signature
end
```

6.5. API Responses

6.5.1. XML Response Structure

If list-xml was specified for the outputFormat parameter a XML response structure is returned. The structure includes both information about the values of the inputs and the individual transaction records. The example provided below contains all the response elements that might be returned in the response.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<transactionSearchResult>
  <query>
    <action />
    <auth />
    <avsCategory />
    <avsResponse />
    <batchNumber />
    <card />
    <cvv2Response />
    <email />
    <custId />
    <haddr />
```

```
    <hcity />
    <hctry />
    <hfname />
    <hlname />
    <hstate />
    <hzip />
    <instr />
    <ip />
    <lastFour />
    <merName>Merchant Name</merName>
    <parentXid />
    <phone />
    <recurRecipe />
    <recurStatus />
    <recurXid />
    <status />
    <subAction />
    <total />
    <transDateFirst>2007/11/1</transDateFirst>
    <transDateLast>2007/11/2</transDateLast>
    <transTimeFirst />
    <transTimeLast />
    <xidFirst />
    <xidLast />
    </query>
  <transactions>
    <transaction>
      <action>order</action>
      <auth>1234567</auth>
      <avsCategory>zip5</avsCategory>
      <avsResponse>Z</avsResponse>
      <batchNumber>7</batchNumber>
      <card>American Express</card>
      <childXids>
        <childXid>123456789</childXid>
      </childXids>
      <cvv2Response>M</cvv2Response>
      <email>test@example.com</email>
      <custId>12345</custId>
      <haddr>123 Street Name</haddr>
      <hcity>City</hcity>
      <hctry>USA</hctry>
      <hfname>First</hfname>
      <hlname>Last</hlname>
      <hstate>UT</hstate>
      <hzip>84010</hzip>
      <instr>cc</instr>
      <ip>11.11.11.11</ip>
      <items>
        <item index="0">
          <itemDetail code="cost">1.99</itemDetail>
          <itemDetail code="desc">Item Description</itemDetail>
          <itemDetail code="qty">1</itemDetail>
        </item>
      </items>
      <merchantData>
        <value name="orderNumber">123456789</value>
      </merchantData>
      <lastFour>2000</lastFour>
      <merName>Merchant Name</merName>
      <parentXid>123456789</parentXid>
      <phone>801-555-1212</phone>
      <recurRecipe>recipe name</recurRecipe>
      <recurStatus>ok</recurStatus>
      <recurXid>123456789</recurXid>
      <saddr>123 Street Name</saddr>
      <scity>City</scity>
      <sctry>USA</sctry>
      <sfname>First</sfname>
      <slname>Last</slname>
```

```
      <sstate>UT</sstate>
      <szip>84010</szip>
      <status>ok</status>
      <subAction>sale</subAction>
      <total>1.99</total>
      <transDate>11/1/2007 12:00:00</transDate>
      <transSrc>session</transSrc>
      <xid>123456789</xid>
    </transaction>
    <transaction>
      ....
    </transaction>
  </transactions>
</transactionSearchResult>
```

6.5.2. CSV Response Structure

If list-csv was specified for the outputFormat parameter a comma delimited response structure is returned. This report has one line per transaction but the number of columns is variable depending on the transaction data in the report. There are two data elements that cause the number of columns to vary:

- If there are any child transactions each child transaction XID is added as a separate column. If there are no child transactions there are no columns added to the report.
- Since each line contains all the order items, the number of order item columns in a report is based on the transaction in the set with the largest number of order items.

Each line can be viewed as having five different sections.

1. Header Set One - This is a fixed set of header values.

   ```
   action,auth,avsCategory,avsResponse,batchNumber,card
   ```

2. Header Set Two - This is a variable set of header values based on whether any child XIDs exist and if so on how many. For each child XID associated with a transaction a column is added. For any given report the number of childXID columns is equal to the maximum number of child XIDs associated with any transaction in the report.

   ```
   childXid_0, childXid_1, childXid_2
   ```

3. Header Set Three - This is a fixed set of header values.

   ```
   cvv2Response,custId, email,haddr,hcity,hctry,hfname,hlname,hstate,hzip,instr,ip
   ```

4. Header Set Four - This set of headers consists of the order items for the request. Each order item has a minimum of three values associated with it (cost, qty, desc), although there can be more. Each value generates two columns, one for the value code and one for the actual value. The column headers in this section are zero based meaning your order item numbers are not preserved in the report. If you pass through items 1 and 2 you will still have column header names like item_0_code_0. Let's consider an example transaction with two order items:

   Table 3.38. Output Example

Item #	Cost	Quantity	Description
1	$1.00	1	Item 1
2	$2.00	2	Item2, with comma

   Since we have six order item attributes, this will result in twelve columns being generated. The output that would be generated for this example transaction is shown below.

   ```
   item_0_code_0,item_0_value_0,item_0_code_1,item_0_value_1,item_0_code_2,item_0_value_2,
   item_1_code_0,item_1_value_0,item_1_code_1,item_1_value_1,item_1_code_2,item_1_value_2
   cost,1.00,desc,Item 1,qty,1,cost,2.00,desc,Item2&#44; with comma,qty,2
   ```

5. Header Set Five - This is a fixed set of header values.

   ```
   lastFour,merName,mid,parentXid,phone,recurRecipe,recurStatus,recurXid,saddr,scity,
   ```

```
sctry,sfname,slname,sstate,szip,status,subAction,total,transDate,transSrc,xid
```

6.6. Testing the Process

If you would like to test your HMAC signature process without having to hit our servers, use the demo signature generated as a test base. If you use the same set of parameters your code should generate the same signature value that was shown in the example.

7. Deprecated XML Method

7.1. Overview

You should not use this method for integrating with the gateway. This is a deprecated method. Information is included here for the reference of users still utilizing this older version. Future versions of this documentation may not include information regarding this method. If you are using this system, please consider making the necessary modifications to your site to utilize the newer XML Connection Method [90] (using the API keys).
Merchants using this interface submit transactions directly with the transaction processing server may do so using XML. XML requests are posted directly to the processing server via HTTPS, ensuring the security of the submitted information. XML responses are generated during the same connection, reducing the number of connections required to process a transaction. The XML schema provides an easy alternative for developers wishing to process real-time transactions. Because XML is open-architecture, developers may create their own Windows COM objects, Java apps, PHP routines, Perl libraries, standardized Web Services, etc. If a merchant's application can generate XML, the merchant can process transactions using this method. Since the XML method is simply another method for processing transactions, all gateway features remain available. This includes the Virtual Terminal, Transaction Listing [32], Testing Interface, etc. By default, this feature is not activated on a gateway account. Please contact the support team to have this feature activated.

There are two separate interface modules available when using the XML connection method (xmltrans.cgi and xmltrans2.cgi). Each interface is built for specific types of transactions. Merchants have access to use both modules.

7.2. Considerations

- If using Java, download the Java Secure Socket Extension package available at *http://java.sun.com*.
- If using a Microsoft technology (such as ASP), a merchant may use the XMLHTTPRequest object of msxml.dll. This solution is built upon wininet.dll, which is not scalable. Consult Microsoft documentation for details.
- If using PHP, a merchant may use the CURL Library functions found at *http://www.php.net/manual/en/ref.curl.php*.
- The XML Connection method is not activated on a gateway account by default. If a merchant desires to use the XML Connection method, an XML activation request needs to be sent to the *support team*.

7.3. The xmltrans.cgi Module

The use of this module allows a merchant to submit force, preauth, and sale transactions. Transactions submitted using this module will be listed in the transaction listing and email confirmations can be sent. These transactions can also be set as recurring. SaleRequests should be sent to the following URL:

https://secure.itransact.com/cgi-bin/rc/xmltrans.cgi

7.3.1. MIME Type Information

The above cgi is accessed with an HTTP POST and requires a CONTENT_TYPE header to be specified. Either "application/x-www-form-urlencoded" or "text/xml" must be used. If "application/ x-www-form-urlencoded"is sent then the HTTP body must contain valid form markup. See WW3 Form Spec for details at w3.org. If "text/xml" is used then the HTTP body should only contain the XML request. If the incorrect MIME type is used, the following response will be sent back to your server:

```
<?xml version="1.0" standalone="yes"?>
```

```
<GatewayFailureResponse>
  <Status>FAILED</Status>
    <ErrorCategory>REQUEST_FORMAT</ErrorCategory>
      <ErrorMessage>Unexpected mime type: </ErrorMessage>
</GatewayFailureResponse>
```

7.3.2. The SaleRequest

To run a transaction, a SaleRequest query must be generated.

```
<?xml version="1.0" standalone="yes"?>
<SaleRequest>
  <CustomerData>
    <Email>name@domain.com</Email>
<!-- Optional CustId-->
    <CustId>12345</CustId>
    <BillingAddress>
      <Address1>100 This Street</Address1>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
      <City>Yourtown</City>
      <State>NY</State>
      <Zip>55555</Zip>
      <Country>USA</Country>
      <Phone>230-555-1212</Phone>
    </BillingAddress>
<!-- Optional Shipping Address -->
    <ShippingAddress>
      <Address1>100 This Street</Address1>
      <City>Thatcity</City>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
      <State>NY</State>
      <Zip>55555</Zip>
      <Country>USA</Country>
      <Phone>231-555-1212</Phone>
    </ShippingAddress>
    <AccountInfo>
<!-- Include either credit card or check info. Not both.-->
<!-- For Credit card transaction. -->
      <CardInfo>
        <CCNum>5454545454545454</CCNum>
        <CCMo>05</CCMo>
<!-- CCMo must be two digits. -->
        <CCYr>2002</CCYr>
<!-- CCYr must be four digits. -->
<!-- CVV2 is optional. Pass only if your Gateway account requires it. -->
        <CVV2Number>123</CVV2Number>
        <CVV2Illegible>1</CVV2Illegible>
<!-- Submit only if CVV number is illegible. -->
      </CardInfo>
<!-- For Check transactions. -->
      <CheckInfo>
        <ABA>324377516</ABA>
        <Account>12345</Account>
        <AccountType>business</AccountType>
<!-- business or personal -->
<!-- Optional -->
        <CheckNum>100</CheckNum>
<!-- Optional -->
        <CheckMemo>Memo text here</CheckMemo>
      </CheckInfo>
    </AccountInfo>
  </CustomerData>
  <TransactionData>
<!-- AuthCode optional. (INDICATES FORCE TRANSACTION) -->
    <AuthCode>12345</AuthCode>
<!-- Preauth optional. (INDICATES PREAUTH TRANSACTION) -->
    <Preauth/>
    <VendorId>XXXXX</VendorId>
```

```
    <VendorPassword>PASSWORD</VendorPassword>
<!-- Optional. This information will be saved on our
servers and is available in the XML transaction report. This is
useful if you want to save your own transaction meta-data with a transaction. -->
    <VendorData>
      <Element>
        <Name>repId</Name>
        <Value>1234567</Value>
      </Element>
    </VendorData>
    <HomePage>http://www.example.com</HomePage>
<!-- Optional -->
    <RecurringData>
      <RecurRecipe>test</RecurRecipe>
      <RecurReps>5</RecurReps>
<!-- Optional (For Split Recurring) -->
      <RecurTotal>100.00</RecurTotal>
<!-- Optional (For Split Recurring) -->
      <RecurDesc>test2</RecurDesc>
<!-- Optional (For Split Recurring) -->
    </RecurringData>
<!-- Optional. Text here will be included in transaction confirmation emails. -->
    <EmailText>
<!-- Up to 10 EmailTextItem elements allowed.-->
      <EmailTextItem>line1</EmailTextItem>
      <EmailTextItem>line2</EmailTextItem>
    </EmailText>
    <OrderItems>
      <Item>
        <Description>item1</Description>
        <Cost>5</Cost>
        <Qty>1</Qty>
      </Item>
      <Item>
        <Description>item2</Description>
        <Cost>3.50</Cost>
        <Qty>1</Qty>
      </Item>
      <Item>
        <Description>item3</Description>
        <Cost>-2.99</Cost>
        <Qty>1</Qty>
      </Item>
   </OrderItems>
  </TransactionData>
</SaleRequest>
```

7.3.3. The SaleResponse

The SaleResponse is essentially a duplicate of the information provided in the SaleRequest with additional approval and verification information included. This information is included in a response from the transaction servers. Note that ALL fields are provided in the response even if they were not included in the request. The schema is shown here with demo data.

```
<SaleResponse>
  <CustomerData>
    <Email>name@domain.com</Email>
    <BillingAddress>
      <Address1>100 This Street</Address1>
      <City>Yourtown</City>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
      <State>NY</State>
      <Zip>55555</Zip>
      <Country>USA</Country>
      <Phone>230-555-1212</Phone>
    </BillingAddress>
    <ShippingAddress>
      <Address1>100 This Street</Address1>
```

```
      <Address2>Building 3</Address2>
      <City>Thatcity</City>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
      <State>NY</State>
      <Zip>55555</Zip>
      <Country>USA</Country>
      <Phone>231-555-1212</Phone>
    </ShippingAddress>
    <AccountInfo>
<!-- For Credit card transactions. -->
      <CardInfo>
        <CCLastFour>5454</CCLastFour>
        <CCName>Mastercard</CCName>
      </CardInfo>
<!-- For Check transactions. -->
      <CheckInfo>
        <AccountType>Business</AccountType>
        <CheckNum>100</CheckNum>
        <CheckMemo>Memo text here.</CheckMemo>
      </CheckInfo>
    </AccountInfo>
  </CustomerData>
  <TransactionData>
    <Status>OK</Status>
<!-- Will be one of: ERROR, FAIL, FAILED, OK -->
<!-- ErrorCategory will be one of :
AVS_FAILURE - Transaction will be automatically voided.
CVV2_FAILURE - Transaction will be automatically voided.
INTERNAL_ERROR - Something unexpected happened.
PROCESSOR_ERROR - Something such as DECLINED, etc .
PROCESSOR_FAIL -
REQUEST_FORMAT - Request received has an invalid format.
REQUEST_VALIDATION - XML content is invalid. -->
    <ErrorCategory></ErrorCategory>
<!-- ErrorMessage could be anything. -->
    <ErrorMessage></ErrorMessage>
<!-- Authorization code received from processing network.-->
    <AuthCode>12345</AuthCode>
<!-- AVSResponse is actual AVS response received from the processor. -->
    <AVSResponse>A</AVSResponse>
<!-- AVSCategory will be one of :
address - Address Matched
address_postal - Address and postal patched
address_zip5 - Address and five digit ZIP matched
address_zip9 - Address and nine digit ZIP matched
address_ok_postal_format_error - Address matched, postal format error
global_non_participant - International with no AVS support
international_address_not_verified - International with no AVS support
no_match - No address or postal match
no_response - No response
not_allowed - Not allowed
postal - Postal match
postal_ok_address_format_error - Postal matched, address format error
service_not_supported - AVS service not supported for card
unavailable - AVS service unavailable.
zip5 - Five digit ZIP matched
zip9 - Nine digit ZIP matched -->
    <AVSCategory>Address</AVSCategory>
<!-- CVV2Response is actual response received from the processor. -->
    <CVV2Response>M</CVV2Response>
    <TimeStamp>20021003172303</TimeStamp>
<!-- TestMode indicates the test status of your gateway account.
0=off 1=on -->
    <TestMode>0</TestMode>
    <Total>100</Total>
    <XID>10000</XID>
    <RecurringData>
      <RecurRecipe>test</RecurRecipe>
      <RecurReps>5</RecurReps>
```

```
      <RecurTotal>100</RecurTotal>
      <RecurDesc>test2</RecurDesc>
    </RecurringData>
  </TransactionData>
</SaleResponse>
```

7.4. The xmltrans2.cgi Module

The use of this module allows a merchant to submit Refund, Force, Postauth, Retry and Void transactions based on previous transactions, and Credit transactions that were not generated through the gateway. Transactions submitted using this module will be listed in the transaction listing and email confirmations can be sent.

https://secure.itransact.com/cgi-bin/rc/xmltrans2.cgi

7.4.1. MIME Type Information

This cgi is accessed with an HTTP POST and requires a CONTENT_TYPE header to be specified. Either "application/x-www-form-urlencoded" or "text/xml" must be used. If "application/ x-www-form-urlencoded" is sent then the HTTP body must contain valid form markup. See WW3 Form Spec for details at w3.org. If "text/xml" is used then the HTTP body should only contain the XML request. If the incorrect MIME type is used, the following response will be sent back to your server:

```
<?xml version="1.0" standalone="yes"?>
  <GatewayFailureResponse>
    <Status>FAILED</Status>
      <ErrorCategory>REQUEST_FORMAT</ErrorCategory>
      <ErrorMessage>Unexpected mime type: </ErrorMessage>
  </GatewayFailureResponse>
```

7.4.2. Testing Tools for xmltrans2.cgi

You can run all requests through xmltrans2 in test mode in order to help with the integration process. There are three different ways to run a XML transaction in test mode:

1. Enable the Test Mode checkbox in your merchant Account Settings. Please remember this turns on Test Mode for all transactions.

2. Setup the Test User First Name [24] field in your merchant account settings. Any xmltrans2 request that contains a BillingAddress FirstName tag with this same value will be run as a test transaction. This is only applicable to a couple transactions such as CreditTransaction.

3. Send through a TestMode tag with the value "TRUE". The TestMode tag which is in the TransactionControl structure always overrides the other test settings. This means that sending through TestMode equal to "FALSE" will cause a transaction to be run as a live transaction regardless of what values are in the Account Settings.

7.4.3. Requests

The xmltrans2.cgi request schemas are the same as the schemas used by the API. The only difference is the credential set that is used. Instead of the user and signature, the deprecated method uses the VendorIdentification set. That is made up of the following:

- **VendorId** - Your five digit gateway ID.
- **VendorPassword** - Your gateway password.
- **HomePage** - Your website URL.

```
<VendorIdentification>
        <VendorId>XXXXX</VendorId>
        <VendorPassword>test</VendorPassword>
        <HomePage>http://example.com</HomePage>
</VendorIdentification>
```

7.4.4. Responses

7.4.4.1. TransactionResponse

Transaction requests will return this response structure.

```
<?xml version="1.0" standalone="yes"?>
<GatewayInterface>
  <TransactionResponse>
    <TransactionResult>
      <Status>text</Status>
<!-- Will be one of: ERROR, FAIL, FAILED, OK -->
      <ErrorCategory>text</ErrorCategory>
<!-- ErrorCategory will be one of :
AVS_FAILURE - Transaction will be automatically voided.
CVV2_FAILURE - Transaction will be automatically voided.
INTERNAL_ERROR - Something unexpected happened.
PROCESSOR_ERROR - Something such as DECLINED, etc .
PROCESSOR_FAIL -
REQUEST_FORMAT - Request received has an invalid format.
REQUEST_VALIDATION - XML content is invalid. -->
      <ErrorMessage>text;/ErrorMessage>
<!-- ErrorMessage could be anything. -->
      <AuthCode></AuthCode>
<!-- Authorization code received from processing network. -->
      <AVSCategory></AVSCategory>
<!-- AVSCategory will be one of :
address - Address Matched
address_postal - Address and postal patched
address_zip5 - Address and five digit zip matched
address_zip9 - Address and nine digit zip matched
address_ok_postal_format_error - Address matched, postal format error
global_non_participant - International with no AVS support
international_address_not_verified - International with no AVS support
no_match - No address or postal match
no_response - No response
not_allowed - Not allowed
postal - Postal match
postal_ok_address_format_error - Postal matched, address format error
service_not_supported - AVS service not supported for card
unavailable - AVS service unavailable.
zip5 - Five digit zip matched
zip9 - Nine digit zip matched -->
      <AVSResponse></AVSResponse>
<!-- AVSResponse is actual AVS response received from the processing network. -->
      <CVV2Response></CVV2Response>
<!-- CVV2Response is actual CVV response received from the processing network. -->
      <TimeStamp></TimeStamp>
      <TestMode>FALSE</TestMode>
<!-- TestMode indicates the test status of your gateway account. TRUE/FALSE -->
      <Total></Total>
      <XID></XID>
      <CustomerData>
        <BillingAddress>
          <Address1 />
          <City></City>
          <FirstName></FirstName>
          <LastName></LastName>
          <State></State>
          <Zip></Zip>
          <Country></Country>
          <Phone></Phone>
        </BillingAddress>
        <ShippingAddress>
          <Address1></Address1>
          <City></City>
          <FirstName></FirstName>
          <LastName></LastName>
          <State></State>
```

```
          <Zip></Zip>
          <Country></Country>
          <Phone></Phone>
        </ShippingAddress>
      </CustomerData>
    </TransactionResult>
  </TransactionResponse>
</GatewayInterface>
```

7.4.4.2. RecurUpdateResponse

This request allows you to modify the transaction information and recurring commands for a recurring transaction.

```
<?xml version="1.0" standalone="yes"?>
<GatewayInterface>
  <RecurUpdateResponse>
    <Status>ok</Status>
    <ErrorCategory></ErrorCategory>
    <ErrorMessage></ErrorMessage>
    <TimeStamp>20060621154341</TimeStamp>
    <TestMode>FALSE</TestMode> <!-- TRUE/FALSE -->
    <RecurDetails>
      <RemReps>10</RemReps>
      <RecipeName>daily</RecipeName>
      <RecurTotal>1.00</RecurTotal>
    </RecurDetails>
  </RecurUpdateResponse>
</GatewayInterface>
```

7.4.4.3. RecurDetailsResponse

RecurDetailsResponse - The RecurDetails request will generate a response like this:

```
<?xml version="1.0" standalone="yes"?>
<GatewayInterface>
  <RecurDetailsResponse>
    <Status>ok</Status>
    <ErrorCategory></ErrorCategory>
    <ErrorMessage></ErrorMessage>
    <TimeStamp>20060621154341</TimeStamp>
    <TestMode>FALSE</TestMode> <!-- TRUE/FALSE -->
    <RecurDetails>
      <CardExpired>FALSE</CardExpired> <!-- TRUE/FALSE -->
      <CardExpiresWithinThirty>FALSE</CardExpiresWithinThirty> <!-- TRUE/FALSE -->
      <CardLastFour>1234</CardLastFour>
      <CardName>Visa</CardName>
      <RemReps>10</RemReps>
      <RecipeName>daily</RecipeName>
      <RecurTotal>1.00</RecurTotal>
    </RecurDetails>
  </RecurDetailsResponse>
</GatewayInterface>
```

8. Testing Your Integration

8.1. Introduction

The gateway has multiple ways that will allow you to run test transactions against the gateway to verify that your integration is correct. When you have set your account into to Test Mode (automatically or manually), you can use this demo account information:

Demo Credit Card Information

- **Card Number** - 5454545454545454
- **Expiration Date** - Any date after the current date

Demo Checking Account Information

- **ABA Routing Number** - 324377516
- **Account Number** - 1234567890

8.2. Test User First Name

Read about the first name testing method here [24].

8.3. Test All Checkbox

Read about the the Test All mode here [24].

8.4. API Test TransactionContol

Read about the testing TransactionControl element for the API here.

8.5. Testing Other Features

If you need to test live functionality like recurring, voids, or transaction searches, you will need us to activate additional testing settings for you. Please send your request to ticket.liaison@itransact.com.

9. Errors and Failures

Transactions can potentially fail one of many reasons. Most failure responses are generated by the EFT processor, credit card processing networks, and the credit card issuing banks. The information below includes the response as it's received from the processor and how the gateway interprets and displays that message.

9.1. NBE Errors

Transactions can potentially fail one of many reasons. Most failure responses are generated by the credit card processing networks and the credit card issuing banks. The information below includes the response as it's received from the processor and how the gateway interprets and displays that message. This is a long list, but in not complete. Processors are always adding new support for errors. If you experience an error not listed here, please let us know so that we can add support in the gateway.

- **Response** - DECLINE

  Message - Code: NBE001 Your credit card was declined by the credit card processing network. Please use another card and resubmit your transaction.

- **Response** - DECLINED YM

  Message - Code: NBE001 Your credit card was declined by the credit card processing network. Please use another card and resubmit your transaction.

- **Response** - DECLINED R

  Message - Code: NBE001 Your credit card was declined by the credit card processing network. Please use another card and resubmit your transaction.

- **Response** - DECLINE RP

  Message - Code: NBE001 Your credit card was declined by the credit card processing network. Please use another card and resubmit your transaction.

- **Response** - AUTH DECLINED 200

Message - Code: NBE001 Your credit card was declined by the credit card processing network.
Please use another card and resubmit your transaction.

- **Response** - TRANS DENIED

  Message - Code: NBE001 Your credit card was declined by the credit card processing network. Please use another card and resubmit your transaction.

- **Response** - INVALID C

  Message - Code: NBE002 We received a response that this is an invalid card. Please use another card and resubmit your transaction.

- **Response** - EXPIRED CARD

  Message - Code: NBE003 The card appears to be expired. Please use another card and resubmit your transaction.

- **Response** - INCORRECT PIN

  Message - Code: NBE004 This system cannot be used to process ATM cards.

- **Response** - PICK UP CARD

  Message - Code: NBE005 The credit card processing network has recognized this card as lost or stolen. The transaction has been cancelled.

- **Response** - CALL

  Message - Code: NBE006 The processing network has responded with a CALL error. Please do not resubmit your transaction. There may be a problem processing your card. Please call your credit card company at the phone number listed on the back of your card.

- **Response** - AMOUNT ERROR

  Message - Code: NBE007 The dollar amount of this transaction is invalid. Please verify the information entered and resubmit.

- **Response** - INVLD TERM ID 1

  Message - Code: NBE008 The credit card processing network has responded with an Invalid Merchant Number error. Your transaction has been cancelled.

- **Response** - INVLD TERM ID 2

  Message - Code: NBE009 The credit card processing network has responded with an Invalid SE Number error. Your transaction has been cancelled.

- **Response** - RECORD NOT FOUND

  Message - Code: NBE010 We received a response that this record could not be found. The transaction has been cancelled.

- **Response** - MUST SETTLE

  Message - Code: NBE011 The credit card processing network has responded with a MUST SETTLE error. We apologize for the inconvenience.

- **Response** - REC NOT FOUND

  Message - Code: NBE013 We received a response that this record could not be found. The transaction has been cancelled.

- **Response** - PLEASE RETRY

  Message - Code: NBE014 We received a PLEASE RETRY message. Please attempt your transaction again or use a different card.

- **Response** - has already been settled

  Message - This transaction has already been settled. You will need to issue a credit instead.

- **Response** - Invalid Bank Number

  Message - Code: NBE016 The credit card processing network has responded with a Bank Number error. Your transaction has been cancelled.

- **Response** - javax.management.ReflectionException

  Message - Our servers are currently undergoing scheduled maintenance. Please try back later...

- **Response** - TRANSACTION NOT FOUND

  Message - Code: NBE017 The processing network no longer has an authorization for this transaction. The transaction cannot be voided.

- **Response** - INVALID M

  Message - Code: NBE018 The credit card processing network has responded with an "Invalid Merchant Number" error. Your transaction has been cancelled.

- **Response** - INV TRAN T

  Message - Code: NBE019 Invalid transaction type. Please verify and resubmit.

- **Response** - AP DUPE

  Message - Code: NBE020 The credit card processing network has recognized this as a duplicate transaction. We apologize for the inconvenience.

- **Response** - EDC UNAVAILABLE

  Message - Code: NBE021 The credit card processing network has responded with an EDC Unavailable error. Please use another card or try your transaction later. We apologize for the inconvenience.

- **Response** - TRANSMIT

  Message - Code: NBE022 This transaction cannot be processed. If you are attempting a VOID you must process a credit instead.

- **Response** - EXPIRED CA

  Message - Code: NBE023 The card appears to be expired. Please use another card and resubmit your transaction.

- **Response** - INV CVV2 M

  Message - Code: NBE024 The CVV2 information entered is invalid. Please verify the information and resubmit your transaction.

- **Response** - CVC2 MISMATCH

  Message - Code: NBE025 The CVC2 information entered is invalid. Please verify the information and resubmit your transaction.

- **Response** - CVV2 MISMATCH

Message - Code: NBE026 The CVV2 information entered is invalid. Please verify the information and resubmit your transaction.

- **Response** - CARD NO. ERROR

  Message - Code: NBE027 Card number error. Please use another card or resubmit your transaction.

- **Response** - PIC UP

  Message - Code: NBE028 The credit card processing network has recognized this card as lost or stolen. The transaction has been cancelled.

- **Response** - UNAUTH TRANS

  Message - Code: NBE029 Unauthorized transaction. Please use another card and resubmit your transaction.

- **Response** - INVLD EXP

  Message - Code: NBE030 You have entered an invalid expiration date. Please return to the form and verify the information entered.

- **Response** - ISSUER UNAVAIL

  Message - Code: NBE032 Your card issuer cannot validate your request. Please use another card and resubmit your transaction.

- **Response** - INVALID EX

  Message - Code: NBE033 You have entered an invalid expiration date. Please return to the form and verify the information entered.

- **Response** - INVLD CODE ACCT

  Message - Code: NBE034 The credit card processing network has responded with an invalid code error. Please check your input and try your transaction again. We apologize for the inconvenience.

- **Response** - SERV NOT ALLOWED

  Message - Code: NBE035 The credit card processing network does not allow this card type or service. Please use a different card and try your transaction again.

- **Response** - INVALID TRAN

  Message - Code: NBE036 This transaction cannot currently be processed. The credit card processing network has responded with an invalid transaction error. Please try back later. We apologize for the inconvenience.

- **Response** - AMEX NOT ALLOWED

  Message - Code: NBE037 This transaction cannot be processed. American Express cards are not accepted. Please use a different card and resubmit your transaction.

- **Response** - ERROR 06

  Message - Code: NBE038 We received an error response while processing your card. Please use another card and resubmit your transaction.

- **Response** - NO SUCH ISSUER

  Message - Code: NBE039 The credit card processing network has responded with a NO SUCH ISSUER ERROR. Please use a different card. We apologize for the inconvenience.

- **Response** - DATE ERROR

Message - Code: NBE040 We received a DATE ERROR while processing your card. Please use another card and resubmit your transaction.

- **Response** - RE ENTER

  Message - Code: NBE041 The credit card processing network has responded with a RE ENTER message. Please try your transaction again. We apologize for the inconvenience.

- **Response** - WRONG PIN

  Message - Code: NBE042 ATM cards cannot be used. The card used must display the Visa, Mastercard, American Express or Discover symbol.

- **Response** - LOST

  Message - Code: NBE043 The credit card processing network has recognized this card as lost or stolen. The transaction has been cancelled.

- **Response** - NO REPLY

  Message - Code: NBE044 We were unable to obtain a response from the credit card processing net- work. Please try your transaction again. We apologize for the inconvenience.

- **Response** - ALREADY REVERSED

  Message - Code: NBE045 This transaction has already been reversed.

- **Response** - VALID RECORD

  Message - Code: NBE046 This transaction cannot be voided. You will need to issue a credit instead.

- **Response** - INVALID AM

  Message - Code: NBE047 We received a response that this is an invalid amount. Please resubmit your transaction.

- **Response** - SYSTEM ERROR or Unknown Error

  Message - Code: NBE048 The credit card processing network experienced a system error during your transaction. Please resubmit your transaction

- **Response** - ACCT LENGTH

  Message - Code: NBE049 The credit card processing network has responded with an ACCT LENGTH ERROR. Please verify your entries and resubmit your transaction.

- **Response** - PIN EXCEED

  Message - Code: NBE050 ATM cards cannot be used. The card used must display the Visa, Mastercard, American Express or Discover symbol.

- **Response** - INVALID ACCT

  Message - Code: NBE051 We received a response that this is an invalid account. Please use another card and resubmit your transaction.

- **Response** - DUPLICATE

  Message - Code: NBE052 The credit card processing network has recognized this as a duplicate transaction. We apologize for the inconvenience.

- **Response** - OVER LIMIT

Message - Code: NBE053 The credit card processing network has responded with a LIMIT error. We apologize for the inconvenience.

- **Response** - INVALID PI

  Message - Code: NBE054 ATM cards cannot be used. The card used must display the Visa, Master- card, American Express or Discover symbol.

- **Response** - NO ACCOUNT

  Message - Code: NBE055 The credit card processing network has responded with a No Account message. Please check your input and try your transaction again. We apologize for the inconvenience.

- **Response** - INVLD AMOUNT

  Message - Code: NBE056 We received a response that this is an invalid amount. Please resubmit your transaction.

- **Response** - NO CHECK ACCOUNT

  Message - Code: NBE057 Your card was declined by the credit card processing network. Please contact your bank regarding the checking account associated with your debit card.

- **Response** - NOT PERMITTED

  Message - Code: NBE058 This transaction type is not permitted by the processing network.

- **Response** - INVLD PIN

  Message - Code: NBE059 ATM cards cannot be used. The card used must display the Visa, Mastercard, American Express or Discover symbol.

- **Response** - OVR LIMT AMT

  Message - Code: NBE060 Your transaction amount cannot be processed. Your transaction has been cancelled.

- **Response** - INVLD MERCH ID

  Message - Code: NBE061 The credit card processing network has responded with an INVALID MERCHANT ID message. Your transaction has been cancelled.

- **Response** - Unknown AccountType

  Message - Code: NBE062 The EFT provider has responded with an Unknown AccountType error. This transaction cannot be processed.

- **Response** - DECLINE-CV2 FAIL

  Message - Code: NBE063 Your credit card was declined because of an invalid CV2 entry. Please verify your information and resubmit your transaction.

- **Response** - CUSTOMEREMAIL

  Message - Code: NBE064 The EFT provider has responded with a CUSTOMEREMAIL error. Please verify that your email address was entered and is correct.

- **Response** - temporarily disabled

  Message - Code: NBE066 Transaction processing is currently disabled for maintenance. Please try back later.

- **Response** - valid 2 character state abbreviation

Message - Code: NBE067 The STATE entry must contain a valid two-character state abbreviation.

- **Response** - MAXIMUM ATTEMPTS

  Message - Code: NBE069 We were unable to obtain a response from the credit card processing net- work. Please try your transaction later. We apologize for the inconvenience.

- **Response** - DINERS NOT ALLOW

  Message - Code: NBE070 This transaction cannot be processed. Diners Club cards are not accepted. Please use a different card and resubmit your transaction.

- **Response** - maximum daily spending

  Message - Code: NBE071 You have exceeded your maximum daily spending limit. Please contact customer service for assistance.

- **Response** - INV TERM ID

  Message - Code: NBE072 The credit card processing network has responded with an INV TERM ID error. Your transaction has been cancelled.

- **Response** - AMNT TOO LRG

  Message - Code: NBE073 The credit card processing network has responded with a AMNT TOO LRG error. We apologize for the inconvenience.

- **Response** - INVALID STORE

  Message - Code: NBE074 The credit card processing network has responded with an INVALID STORE error. Your transaction has been cancelled.

- **Response** - TERM ID ERROR

  Message - Code: NBE075 The credit card processing network has responded with an TERM ID ERROR message. Your transaction has been cancelled.

- **Response** - FAILURE CV

  Message - Code: NBE076 The credit card processing network has responded with a FAILURE CV message. Please try your transaction again. We apologize for the inconvenience.

- **Response** - FAILURE HV

  Message - Code: NBE077 The credit card processing network has responded with a FAILURE HV message. Please try your transaction again. We apologize for the inconvenience.

- **Response** - Unknown Error

  Message - Code: NBE078 Your credit card was not approved. Please use another card and resubmit your transaction.

- **Response** - CARD NOT ALLOW

  Message - Code: NBE079 Your account does not accept this card type. Please use a different card.

- **Response** - SEC VIOLATION

  Message - We received a SEC VIOLATION response from Discover. Please try again using the CVV code.

- **Response** - SECURITY VIOLATION

Message - Code: NBE081 We received a SECURITY VIOLATION response from Discover. Please try again using the CVV code.

9.2. NAVS Errors

These errors are generated based on the auto-void settings chosen by the merchant in the Fraud Controls.

- **Response** - no_match

  Message - Code: NAVS001 The address and zip entered do not match the address and zip listed on your credit card account. Your transaction will be voided.

- **Response** - address

  Message - Code: NAVS002 The zip code entered does not match the zip code listed on your credit card account. Your transaction will be voided.

- **Response** - zip5

  Message - Code: NAVS003 The address entered does not match the address on your credit card account. Your transaction will be voided.

- **Response** - zip9

  Message - Code: NAVS004 The address entered does not match the address on your credit card account. Your transaction will be voided.

- **Response** - no_response

  Message - Code: NAVS005 The address and/or zip code listed on your account could not be verified. We received no AVS response. Your transaction will be voided.

- **Response** - avs_incompatible_card_type

  Message - Code: NAVS006 The address and zip code could not be verified because of an incompatible card type. Your transaction will be voided.

- **Response** - cardnumber_not_on_file

  Message - Code: NAVS007 The address and zip code could not be verified. Your card number is not on file in the processing network database. Your transaction will be voided.

- **Response** - domestic_address_not_verified

  Message - Code: NAVS008 The address entered could not be verified. Your transaction will be voided.

- **Response** - unavailable

  Message - Code: NAVS009 The address and zip code verification service is unavailable for your credit card account. Your transaction will be voided.

- **Response** - service_not_supported

  Message - Code: NAVS010 The address and zip code verification service is not supported for your credit card account. Your transaction will be voided.

- **Response** - address_verification_not_supported

  Message - Code: NAVS011 The address and zip code verification service is not supported for your credit card account. Your transaction will be voided.

- **Response** - global_non_participant

  Message - Code: NAVS012 The address and zip code verification service is not supported outside of the United States. Your transaction will be voided.

- **Response** - avs_error

  Message - Code: NAVS013 The address and/or zip code listed on your account could not be verified. We received an AVS error response. Your transaction will be voided.

- **Response** - postal

  Message - Code: NAVS014 The address entered does not match the address on your credit card account. Your transaction will be voided.

9.3. NBF Errors

These errors indicate that there was a communication error or a system error.

- **Response** - SERV NOT ALLOWED

  Message - Code: NBF001 The credit card processing network does not allow this card type or request. SERV NOT ALLOWED error.

- **Response** - INVALID TERM ID

  Message - Code: NBF002 The credit card processing network has responded with an Invalid Term Id error. Your transaction cannot be processed.

- **Response** - INVLD VOID DATA

  Message - Code: NBF003 This transaction cannot be voided. You will need to issue a credit instead.

- **Response** - APPL TYPE ERROR

  Message - Code: NBF004 The credit card processing network has responded with an Appl Type error. Your transaction cannot be processed.

- **Response** - REC NOT FOUND

  Message - Code: NBF005 This transaction cannot be voided. You will need to issue a credit instead.

- **Response** - INVLD TERM ID 1

  Message - Code: NBF006 The credit card processing network has responded with an Invalid Merchant Number error. Your transaction has been cancelled.

- **Response** - INVLD TERM ID 2

  Message - Code: NBF007 The credit card processing network has responded with an Invalid SE Number error. Your transaction has been cancelled.

- **Response** - INVLD DATA

  Message - Code: NBF008 The credit card processing network has responded with an invalid data error. Please verify the information and resubmit.

- **Response** - AMOUNT ERROR

  Message - Code: NBF009 The dollar amount of this transaction is invalid. Please verify the information entered and resubmit.

- **Response** - The transaction has already been settled

Message - Code: NBF010 This transaction cannot be voided. You will need to issue a credit instead.

- **Response** - CaughtIOException:Read timed out

  Message - Code: NBF011 The Elavon/Nova processing network is not responding to our request for authorization. Please try your transaction later. We apologize for the inconvenience.

- **Response** - invalid data

  Message - Code: NBF012 The credit card processing network has responded with an invalid data error. Please verify the information and resubmit.

- **Response** - Maximum number of attempts exceeded

  Message - Code: NBF013 The Elavon/Nova processing network is not responding to our request for authorization. Please try your transaction later. We apologize for the inconvenience.

- **Response** - Maximum Postal Code size is 12 characters

  Message - Code: NBF014 The maximum postal code size is 12 characters. Please verify this information and resubmit.

- **Response** - Response Timeout

  Message - Code: NBF015 The processing network is not responding to our request for authorization.
  Please try your transaction again.

- **Response** - Invalid Bank Number

  Message - Code: NBF016 The credit card processing network has responded with a Bank Number error. Your transaction has been cancelled.

- **Response** - CardCVV2Data/CVV2Indicator

  Message - Code: NBF017 The CVV number entered is invalid. It must be three or four digits in length.

- **Response** - nova.NovaMerchantLockFactory.getNovaHybr

  Message - Code: NBF018 This transaction could not be processed. The support team has been notified of the problem.

- **Response** - This request has already been settled

  Message - Code: NBF019 This transaction cannot be voided. You will need to issue a credit instead.

- **Response** - INVALID STOREKEY

  Message - Code: NBF020 This transaction cannot be processed. The processing network has responded with an INVALID STOREKEY error.

- **Response** - NOT PERMITTED

  Message - Code: NBF021 This transaction type is not permitted by the processing network

- **Response** - VALID RECORD TO VOID

  Message - Code: NBF022 This transaction cannot be voided.

- **Response** - suspended for maintenance

  Message - Code: NBF023 Our servers are currently undergoing scheduled maintenance. Please try back later.

- **Response** - temporarily disabled

Message - Code: NBF024 Transaction processing is currently disabled for maintenance. Please try back later

- **Response** - EDC UNAVAILABLE

  Message - Code: NBF025 The credit card processing network has responded with an EDC Unavailable error. This indicates that the processing network application is not available. Please use another card or try your transaction later. We apologize for the inconvenience

- **Response** - NO REPLY/TRANSMIT ERROR

  Message - Code: NBF027 We were unable to obtain a response from the credit card processing network. Please try your transaction again. We apologize for the inconvenience.

- **Response** - DUPLICATE

  Message - Code: NBF028 The credit card processing network has recognized this as a duplicate transaction. We apologize for the inconvenience.

- **Response** - valid 2 character state abbreviation

  Message - Code: NBF029 The STATE entry must contain a valid two-character state abbreviation.

- **Response** - DECLINE

  Message - Code: NBF030 Your credit card was declined by the credit card processing network. Please use another card and resubmit your transaction.

- **Response** - Consumer verification negative.

  Message - Consumer verification negative

9.4. NCVV Errors

These auto-void errors indicate that the cardholder's card security value verification did not meet your required level of validation.

- **Response** - " " or -

  Message - Code: NCVV001 No CVV2 Response Was Returned. Your Transaction Will Be Voided

- **Response** - P

  Message - Code: NCVV002 CVV2 Information Was Not Processed. Your Transaction Will Be Voided

- **Response** - N

  Message - Code: NCVV003 CVV2 Information Did Not Match. Your Transaction Will Be Voided

- **Response** - U

  Message - Code: NCVV004 Issuer Not Certified For CVV2. Your Transaction Will Be Voided

- **Response** - S

  Message - Code: NCVV005 You indicated that CVV2 was not available, however the issuer indicates it should be present. Your Transaction Will Be Voided

9.5. THR Errors

There is only one THR error. When THR001 is displayed, this indicates that your account has the Restrict Order Usage [25] feature activated and it has been engaged by a declined transaction attempt. The THR error will display for a specific IP address until the restriction time has expired. This is how the error message displays:

- Code: THR001 This transaction has been suspended and will not be processed. Please see the section for additional information.

9.6. VCC Errors

VCC errors are errors generated by the card validation script. All credit card account numbers follow a specific algorithm. If a bogus account number is attempted, a VCC error will display:

- Code: VCC001 The credit card number entered contains non-numeric characters. Please verify.
- Code: VCC002 The credit card entered matches no known card type. Please use a different card.
- Code: VCC003 The credit card number entered has the wrong number of digits.
- Code: VCC004 The credit card number entered is invalid. Please verify and resubmit.

9.7. VALSYS Errors

If you receive a response that mentions VALSYS, it indicates that your account has been taken out of test mode, but does not have an active merchant account enabled in the gateway. This error can be fixed either by reactivating test mode, or adding a merchant account to the gateway. Merchants may also experience this error by passing a "FALSE" value for TestMode through the the xml method if there gateway has no active merchant account. Never pass a "FALSE" value for TestMode on a gateway that is strictly a test account.

9.8. REQUEST_VALIDATION Errors

If you receive a response that mentions REQUEST_VALIDATION, it indicates that the credentials being used are incorrect, or the IP address submitting the request has not been set as an allowed address in the IP Filter [26]. The rest of the response will indicate the cause of the error.

9.9. REQUEST_FORMAT Errors

If you receive a response that mentions REQUEST_FORMAT, it indicates that your request is not following the required specifications. Please refer to the documentation.

9.10. VXN Errors

If you receive a response that mentions VXN, it indicates that your Datawire ID account information is invalid. Please contact the support team for help.

Chapter 4. Glossary

1. Glossary of Terms

- **Authorization** - Receiving an approval for a credit card transaction from a card issuing bank through a response from a credit card processing network.
- **Authorization Code** - Alpha-numeric response received from the processing network indicating a credit card transaction approval. Approvals “freeze” a specified amount on a customer's credit line, but will not actually charge the card until batch settlement.
- **AVS** - The Address Verification System is one of the credit card industry's methods to prevent fraud. It is used to verify the billing address for Internet-based transactions. Domestic US transactions can be verified using the AVS, but very few foreign credit card banks support AVS.
- **AVSOnly** - This transaction type can be used to verify the AVS and CVV data for a credit card without charging the card. A zero amount AVSOnly transaction can be setup to recur or be resubmitted for actual charges. This is not available for check transactions.
- **Batch** - A group of credit card transactions - normally batched together by day - which deposits together by card type to a merchant.
- **Batch Settlement** - When the gateway system closes the group of open authorizations and completes the transaction process. Batch settlement takes place each day that there is at least one transaction or when a manual batch settlement is generated.
- **Brick and Mortar** - Physical retail storefronts. Often used to indicate that a business is not Internet based.
- **BuyNow Format** - A simple method to add a button to a website to allow a customer to place an order by clicking on a button on the merchant's site which takes them to a secure server environment where a transaction can be submitted.
- **Child Transaction (CXID)** - A transaction that was processed from a previously entered, or Parent transaction.
- **CID** - American Express Card Identification number listed as a security code on the front of AMEX cards.
- **CISP** - Cardholder Information Security Program is Visa's standard requirements for safeguarding personal cardholder information. The gateway system meets and exceeds these standards.
- **Control Panel** - The administrative interface that allows a merchant to activate, learn about, and utilize the features of the gateway.
- **Crediting** - To generate a refund to a customer's account. Money is withdrawn from a merchant's account and deposited into the customer's account.
- **Credit Card Merchant Account** - An account which authorizes a merchant to accept a specified credit card type.
- **CVV** - Cardholder Verification Values are the three or four digit security codes listed on the back of most credit cards.
- **Deposit** - When the merchant account or EFT account makes direct payment into a merchant's bank account.
- **E-Commerce** - The buying and selling of goods and services over the Internet.
- **EFT** - Electronic Funds Transfer. This is a method for accepting check payments over the Internet. Also referred to as ACH/Automated Clearinghouse transactions.
- **Encryption** - The translation of data into a secure format. Encryption is the most effective way to achieve data security. To read an encrypted file, you must have access to a key or password that enables you to decrypt it. Encrypted information is unintelligible.

- **Form Wizard** - A simple tool used to create basic order form pages that are built to the specifications of the gateway.
- **Gateway System** - A tool used by businesses to accept payments by credit cards and check over the Internet. These systems securely submit transaction information to credit card processing networks and record and display the approval or decline responses.
- **HTML** - Hypertext Markup Language. A basic web language that can be used to create order forms which communicate with the gateway.
- **Merchant** - A business which accepts credit cards or checks as payment for their services or products.
- **Merchant/Developer Toolkit** - An interface which provides integration information and examples for the gateway account.
- **NACHA Processing** - The gateway can create direct NACHA formatted file. To use this system, a merchant must meet necessary requirements with their bank and iTransact must run a through a verification process before a merchant can utilize the system. To inquire, please submit a support ticket.
- **Parent Transaction (PXID)** - An original transaction from which subsequent, or Child, transactions have been processed.
- **Partial Authorization** - The industry calls this a "split-tender purchase transaction." This allows you as a merchant to approve a partial amount on a debit, prepaid, or gift card transaction, and then allow the remaining balance be paid with another card or payment method. This improves your customer's payment experience because instead of their card being totally declined, they will have the ability to make a partial payment on the card they want to use and then immediately complete the transaction with another form of payment. This will increase your customers' payment options and help you complete more sales.
- **PCI** - Payment Card Industry Data Security Standard. The credit card industry's standard requirements for safeguarding personal cardholder information. The gateway system meets and exceeds these standards.
- **Pre-authorization** - A transaction which only verifies the card account and “freezes” a set amount in the account, but it does not actually charge the card. A pre-authorized transaction can be converted to a full transaction by running a post-authorization on the transactions. If no post-authorization is run, the money is never paid to the merchant.
- **Processing Network/Platform** - The merchant service center which processes credit card transactions.
- **Real-Time Processing** - When a credit card is approved over the Internet within seconds of being submitted through a merchant's gateway account.
- **Recurring Recipe** - The schedule which dictates when a transaction rebills.
- **Recurring Transactions** - Transactions which occur on an ongoing basis according to schedule or recipe.
- **RediCheck** - iTransact’s proprietary check draft printing service.
- **Refund** - To generate a credit to a customer's account. Money is withdrawn from a merchant's account and deposited into the customer's account.
- **Resubmit** - A function which allows a merchant to process a subsequent credit card transaction through the gateway based on previous transactions without having to re-enter the credit card information.
- **Ret_addr** - The full URL to which customers are directed after order completion.
- **Ret_mode** - The Return Mode Function enables you to by-pass the intermediate "Continue" page that is displayed after a transaction completes. This is useful if you would like your online transaction system to be "transparent" to the end-user. This applies for both valid and declined transactions.
- **SEC Codes** - This is the standard entry class used for EFT processing.

- **Secure Server** - Technology that is required to be used for websites that want to accept payments. Merchants may use their own secure servers or they can use the gateway's secure servers if they utilize the Split Form method.
- **Shopping Cart** - This is a dynamic order form system which allows a merchant's website to calculate things like shipping and taxes. A shopping cart can be used to submit transaction information to the gateway system.
- **Split Form** - An HTML order format which allows a merchant to process transactions securely without their own secure server. A customer enters public information on a merchant's non-secure page and then is taken to a secure server page to enter private information.
- **SSL** - Secure Socket Layer technology is a protocol designed to enable secure transmission of information on the Internet. It provides encryption and integrity of communications along with strong authentication using digital certificates.
- **Standard Form** - An HTML order format used by merchants who have their own secure servers.
- **Test Mode** - A setting in an account which allows a merchant to test the functionality of their order forms. When an account is in TestMode, no transactions will be processed or charged. The gate- way system will generate emails and postbacks, but nothing will be recorded in the Transaction Listing.
- **Test Transaction Accounts** - In Test Mode, you may use the following test account information:

  For processing credit cards:

  - Credit Card Number: 5454545454545454
  - Exp. Date: anything not expired

  For processing echecks:

  - Checking ABA Number: 324377516
  - Account Number: 1234567890
- **Transaction Listing** - This interface allows a merchant to view a history of transactions based on a date range selected in the Control Panel interface.
- **URL** - Uniform Resource Locator. It is a string of characters conforming to a standardized format, which refers to a resource on the Internet (such as a document or an image) by its location. The web address of a page on the Internet.
- **URL (Absolute)** - The full file address of a webpage including "http://" or "https://".
- **Virtual Terminal** - An online interface that allows a merchant to manually process credit card transactions as if it was being entered and processed through a physical credit card terminal.
- **Void** - This is used to prevent an authorized transaction from closing in a batch settlement. This cancels a transaction. A transaction can be voided up until batch settlement begins. After that time a credit/refund must be run on a transaction. For all processors, a void will attempt to reverse the approval and release the frozen authorized funds back to a cardholders available credit line. If the reversal is unsuccessful, a void of an authorized transaction cancels the charge, but does not cancel an authorization. An authorization freezes funds in an account, so that a completed charge can withdraw those frozen funds. A voided authorization may "freeze" the funds in the customer's account for up to 10 days.
- **XML** - Extensible Markup Language. A very flexible text format that can be used to generate transaction queries through the gateway system.

2. SEC Code Information

The Standard Entry Class Codes

Due to NACHA requirements, it is mandatory for some merchants to pass a standard entry class code to as an identifier with EFT transactions. If you have a CheckGateway account that is listed as using multiple SEC codes, you will need to pass through an appropriate code here. If you are listed as using a single code, this field will be optional. We are not able to verify the status of your account with CheckGateway before sending through requests. This means that any errors due to using a SEC code not supported by your account will be generated by CheckGateway. If you do not know how your account is setup with CheckGateway, please contact their customer service department.

The gateway supports the three letter values for the SEC Codes. Here is an explanation of those.

- **WEB** - Internet (Initiated and authorized via the web through a secure system, can be a single entry or recurring debit, 60 day return timeframe may apply)
- **ARC** - Accounts Receivable (Used for check conversion of one-time payments received via US Mail or a drop box, 60 day return timeframe may apply)
- **BOC** - Back Office Conversion (Used for check conversion of one-time payments received at points-of-purchase or manned payment locations, 60 day return timeframe may apply)
- **CCD** - Corporate Credit Or Debit (B2B payments or disbursements, can be a single entry or recurring transaction, 24 hour return timeframe)
- **POP** - Point Of Purchase (Check provided at point-of-purchase, and the scanned, voided, and returned to customer, for a one-time payment less than $25,000.00, 60 day return timeframe may apply)
- **PPD** - Prearranged Payment (Requires written authorization from customer for debits, can be used for credits, 60 day return timeframe for unauthorized returns)
- **RCK** - Returned Check (Entry of a previously bounced check for less than $2500.00, NSF fee must be initiated as a separate transaction, 60 day return timeframe may apply)
- **TEL** - Telephone (One-time payment initiated and authorized via phone call, can receive authorization in writing or via taped phone conversation, 60 day return timeframe may apply)

A merchant can hard code these values into their Standard forms or XML requests. The Split and Buynow forms support entry of these values on the non-secure page. The Virtual Terminal provides a drop down menu to allow for entry of the SEC Code.