QuickCommerce
PRO

Integration Guide

Revision 4
January 20, 2000

E-Commerce Exchange
1 Mauchly
Irvine, California 92618
www.ecx.com
support@ecx.com
(800) 757-6318

Contents

FIGURES

What's In the Guide

The QuickCommerce PRO Integration Guide is specialized to meet the needs of your operational, technical, and accounting staff. This guide will help your users to quickly familiarize themselves with the application to make full use of its many powerful tools and to maximize the profitability of your e-commerce operation.

Other Manuals available to view for QuickCommerce PRO:

Getting Started with QuickCommerce PRO	Provides a basic introduction to the merchant interface and use of the QuickCommerce PRO system
The QuickCommerce PRO User Guide	Contains a detailed description of the features and functions of the QuickCommerce PRO system

Guide Conventions

Submit	**Command buttons to click are bold**
<u>Profile Editor</u>	Links within the QuickCommerce PRO are underlined
www.quickcommerce.com	Hyperlinks display in Word hyperlink style
Account Updated	Computer-generated text messages display in bold italic
`declineurl=http://trans.quickcommerce.com/~ats/declined.html`	Computer code, such as HTML, C++, or Java is displayed in Courier
📝	Notes call your attention to special features or workarounds

QuickCommerce PRO

QuickCommerce PRO is a merchant transaction service that allows you to receive secure credit card and check payments for goods and services sold over the Internet. In addition to the application's secure payment features, it also provides a merchant with a comprehensive family of online and downloadable transaction management and accounting reports.

Setting Up a QuickCommerce PRO Account

The first step in setting up your QuickCommerce PRO account is to contact E-Commerce Exchange at 800-748-6318 to complete your registration by telephone. You can also request registration materials by e-mail at internetsales@ecx.com.

When your account is confirmed and set up, you will receive your five-digit QUICKCOMMERCE PRO Merchant ID number (ECX ID), a **very** important number, that identifies your account in the QuickCommerce PRO system and allows the system server to authenticate transactions originating from you. Be sure to include your QUICKCOMMERCE PRO ECX ID number in all correspondence with E-Commerce Exchange.

See the companion Manual *Getting Started with QuickCommerce PRO* for more information on setting up your account with E-Commerce Exchange.

Linking Your System to QuickCommerce PRO

Depending upon the transaction processing method, linking your system to the QuickCommerce PRO system may require a sophisticated understanding of HTML coding and network protocols.

This guide is intended for your management and technical support staff responsible for adopting your existing e-commerce systems to function smoothly with QuickCommerce PRO. This guide discusses the requirements for each transaction processing alternative. We recommend that you carefully review these materials before making your configuration choice.

System Overview

The QuickCommerce PRO system processes secure on-line payment transactions using many 3rd party payment authorization networks. The system can process both credit cards and electronic check payments. The authorization network secures credit card payment authorizations while the automated clearinghouse (ACH) network processes electronic checks.

QuickCommerce PRO automatically selects the appropriate payment network based upon the requested transaction type (credit card or ACH) and the authorizing network or Federal Reserve Bank region supported by the bank where you have a merchant account. Figure 1 is a conceptual schematic of the QuickCommerce PRO system.

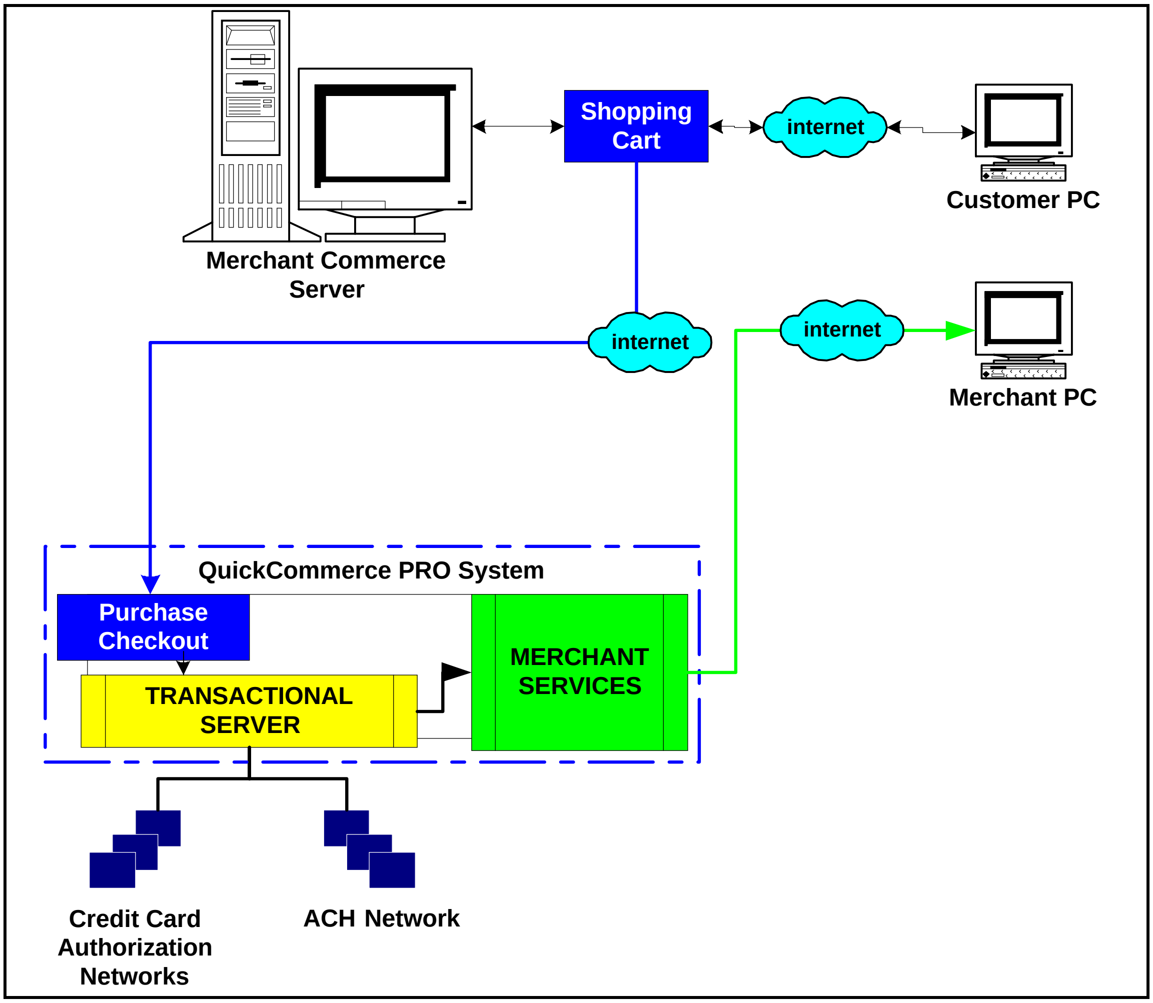


Figure 1 – QuickCommerce PRO system

Consumer PC

The on-line customer accesses your Merchant Commerce Server through the Internet to initiate the shopping experience. Typically, items are selected for purchase and placed in a virtual Shopping Cart. When the customer's purchase decisions are completed and confirmed, the Purchase Checkout module is activated. Purchase Checkout uses preloaded parameters like price, appropriate tax rate, item weight, and available shipping options to calculate the total amount due for the transaction. This calculated charge is passed back to the on-line customer as an invoice for their approval. The approved invoice activates the Transaction Engine to secure payment authorization through the appropriate authorization network.

Merchant Commerce Server

The Merchant Commerce Server supports the customer's shopping and purchasing experiences. It may host its own Shopping Cart and link to the Purchase Checkout and Transaction Engine modules for payment processing.

Shopping Cart

Shopping Cart programs are used to collect customer purchase and Merchant pricing information. The Shopping Cart program resides on the Merchant's server and is typically provided by a third party, however, E-Commerce Exchange can provide a simple shopping cart module as an example for Merchant's who wish to design and develop their own shopping carts. The Transaction Engine uses the information collected by the Shopping Cart to determine the amount for account debit authorizations.

Merchant PC

The Merchant connects to the QUICKCOMMERCE PRO Merchant Services module through the Internet. Using this interface, the Merchant may collect transaction, accounting, and customer information reports.

QuickCommerce PRO System Modules

QuickCommerce PRO modules accomplish the following functions:

Purchase Checkout completes the customer purchase experience by collecting payment information from the shopping cart or membership enrollment activities, calculating the amount to be charged, and passing this information to the Transaction Engine for processing.

Transaction Engine processes online, batch and recurring subscription payment authorization requests. Transactions are processed immediately, in real time. Within seconds, consumers receive an acceptance or decline notification. Funds from accepted credit card transactions are deposited into your checking account, typically within 24 hours. Funds from accepted electronic check (ACH) transactions are deposited into your merchant bank account within six business days.

Merchant Services provides Merchant account management functions and reports.

Transaction Security

Payment processing requires mechanisms using encryption to scramble data to protect customer payment information. Customer's private information, especially credit card numbers and bank account numbers, are securely encrypted as they are transmitted over the Internet. The two most common data encryption methods are:

1) SSL (Secure Sockets Layer)
2) DES3 (Data Encryption Standard).

SSL Encryption

Secure Socket Layer (SSL) uses a public key to provide encryption between the host server and client browser and is the most secure encryption method. Many browsers including Microsoft Internet Explorer and Netscape Navigator support SSL encryption, but you will need to host your site on a secure server running SSL. When Internet transmissions are made via SSL, the protocol for the Uniform Resource Locator (URL) address must include HTTPS, rather than HTTP to direct the transmission to the secure SSL port.

The SSL public key encryption system works this way. The receiving computer discloses its public key and any other computer can use that public key to encrypt data that it sends to the receiving computer. While the public key empowers anyone to encrypt a message, decryption is **not** possible on the basis of the public key. Only the receiving computer has the ability to decrypt, therefore, there is no need to distribute or store private keys, which may fall into the wrong hands.

With SSL, you must obtain a SSL library to write a direct socket program to interface with QuickCommerce PRO.

DES3 Encryption

Unlike SSL, DES3 or Triple DES utilizes a private key and Internet transmissions are done using HTTP protocol, not HTTPS.

The DES3 private key encryption system works as follows. Both the transmitting computer and the receiving computer must possess the same private key to encrypt and then decrypt the message. The reason this system is less secure than SSL is that the private key must be delivered from one computer to the other and there is the risk of it falling into the wrong hands during this process. The ideal method for private key transmission is by surface mail, but this involves a delay of several days. Furthermore, once delivered, the private key could still fall into the wrong hands.

If you elect to use DES3, E-Commerce Exchange can provide sample C++ source code for the recommended implementation of the DES3 algorithm. There is currently no PERL implementation of DES3.

Online Transaction Processing

The QuickCommerce PRO system allows you to use one of the following transaction processing methods, which include:

- Purchase Checkout (Preview Method)
- Payment Submission (Quicksale Method) with a Web form
- Payment Submission (Quicksale Method) with a Socket connection
- Batch Payment Submission with FTP
- Batch Payment Submission with E-mail

Each method is a distinct way to connect your system and your customer's Web browser to the QuickCommerce PRO system and the banking system. You must choose a method when you sign up. We recommend you read the entire guide and examine Table 1 before you decide which method to use.

Table 1 – Product gateway comparison

Processing Method	Type of Sales	Ability to Customize the Interface	QCPRO URL Visibility	Security	Difficulty	Interface
Purchase Checkout	Any Goods	Limited	Visible	Provided	Moderate	HTML
Payment Submission						
- via Web form	Any Goods	Yes, unlimited	Visible	Provided*	Moderate	HTML
- via Socket connection	Any Goods	Yes, unlimited	Hidden	You Must Provide	Difficult	HTTP
Batch Submission						
- via File Transfer	Any Goods	Yes, unlimited	Hidden	Low	Moderate	FTP
- via Electronic mail	Any Goods	Yes, unlimited	Hidden	None	Moderate	E-mail

*If your Web site is not hosted on a secure server, transactions are secure only if your programmers follow recommended procedures.

Purchase Checkout (or Preview Method)

Purchase Checkout (also called the Preview Method) provides an easy way for customers to purchase your products online. It guarantees secure transactions, even if your Web site is not hosted on a secure server.

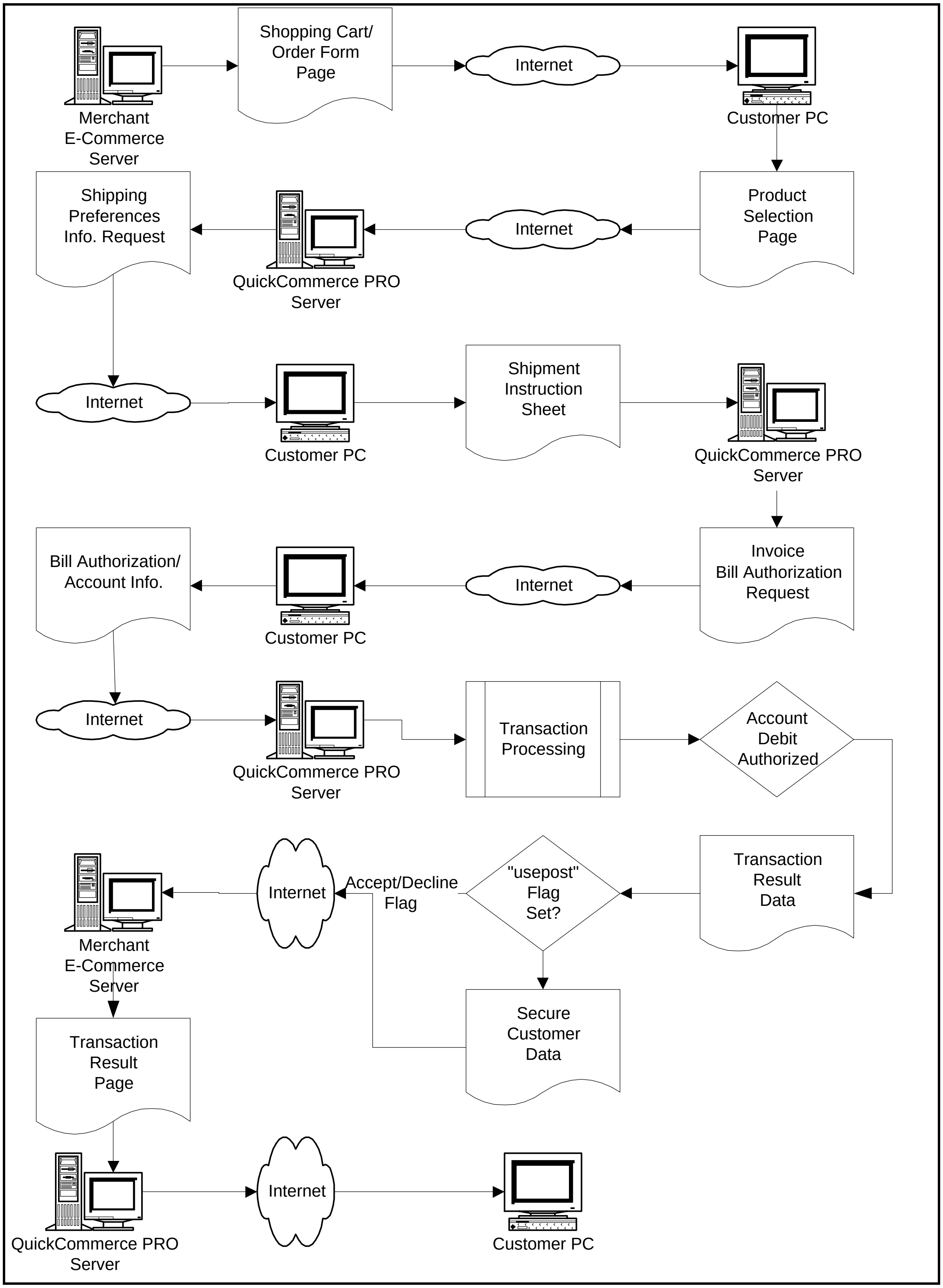
When the customer makes a purchase, QuickCommerce PRO initiates a secure dialog to obtain customer information and the preferred shipping method. The system displays the finalized order with the total billing amount. Your customer then provides a credit card number or bank account number and QuickCommerce PRO processes the transaction.

Purchase Checkout uses standard QUICKCOMMERCE PRO screens. To include your logo and company name on the standard pages, contact Customer Service.

How It Works

1. Your system displays the page with the Shopping Cart checkout or the Order Form to your customer. The page contains product information and choices to facilitate item selection. Data, in HTML hidden fields, with item price and shipping weight is also included in this page. With this information, QUICKCOMMERCE PRO calculates the transaction invoice.
2. The customer finalizes the decision on items and quantities and clicks **Submit**. This action initiates the secure transfer of the completed page, including the number of items ordered, weight, and unit price to QUICKCOMMERCE PRO.
3. QuickCommerce PRO transfers a page to your customer requesting preferred shipping method such as UPS Ground and shipping instructions.
4. After the information is provided, the page is securely transferred to QUICKCOMMERCE PRO and used to calculate the shipping charge, sales tax and total amount due.
5. QUICKCOMMERCE PRO transfers a transaction request page to your customer showing the amount due calculated for this transaction and requests a credit card number or bank account number to debit for the total amount.
6. The customer provides the requested account information and clicks **Submit.** The transaction request data is securely transferred to QUICKCOMMERCE PRO, which then processes the transaction request.
7. If the transaction is accepted, QUICKCOMMERCE PRO generates a receipt for the transaction and e-mails it to you and to your customer. You can configure the system to skip the e-mail notification for you, your customer, or both.
8. The transaction result data is sent securely from QuickCommerce PRO to your system. At minimum, the data contains the accept/decline flag. Your programmers can extract additional secure transaction data by setting the **usepost** flag.
9. Your system sends a transaction result page from your accepturl or declineurl to QUICKCOMMERCE PRO. Your programmers can use CGI, ASP or other methods to dynamically create this page.
10. QUICKCOMMERCE PRO passes the transaction result page to your customer’s browser.

Figure 2 illustrates the Purchase Checkout transaction process.


Merchant
E-Commerce
Server
Shopping Cart/
Order Form
Page
Internet
Customer PC
Shipping
Preferences
Info. Request
QuickCommerce PRO
Server
Internet
Product
Selection
Page
Internet
Customer PC
Shipment
Instruction
Sheet
QuickCommerce PRO
Server
Bill Authorization/
Account Info.
Customer PC
Internet
Invoice
Bill Authorization
Request
Internet
QuickCommerce PRO
Server
Transaction
Processing
Account
Debit
Authorized
Merchant
E-Commerce
Server
Internet
Accept/Decline
Flag
"usepost"
Flag
Set?
Transaction
Result
Data
Secure
Customer
Data
Transaction
Result
Page
QuickCommerce PRO
Server
Internet
Customer PC

Security

With Purchase Checkout processing, QUICKCOMMERCE PRO secures all customer transactions using SSL. Your e-commerce Web server does not have to be a secure server to take advantage of the SSL security provided by QuickCommerce PRO.

Getting Information About Your Customers

You may get information about your customers and their transactions in three different ways:

1. E-mail automatically generated and sent at the time of the transaction. Credit card and bank account numbers are omitted from e-mail receipts.
2. Data sent at the time of the transaction, which your system can use to update your database. Transaction data is only sent if your programmers have set the **usepost** flag.
3. Logging into your QuickCommerce PRO account some time after the transaction, and using the Transaction Menu to generate reports.

What Your Programmers Do

To create both the final order form and the transaction results pages, your programmers can use Web technologies, including CGI scripts, Active Server Pages, Cold Fusion applications, and so on. During the transaction sequence, several Web pages must appear on your customer's browser that are standard Purchase Checkout processing screens.

You may have your logo and company name included in these standard screens; however, they must have the same basic layout as those used by other merchants using Purchase Checkout transaction processing. Your customers will see the QuickCommerce PRO URL displayed on their browsers during execution of the secure transaction steps. Your programmers must be proficient in HTML.

Integrating Your Web Site

Your Web site must integrate with QuickCommerce PRO at Step 1 and Step 7 of the transaction process. When your system transfers the Shopping Cart checkout or Order form page (Step 1), the HTML for your Shopping Cart checkout page or Order form must contain an HTML form statement meeting the following requirements:

- Posts to the QUICKCOMMERCE PRO server at http://trans.quickcommerce.com/cgi-bin/ats.cgi
- Contains certain **required** inputs, to which QuickCommerce PRO will respond
- Contains **optional** inputs including Merchant-defined inputs, to which QuickCommerce PRO will not respond. The QuickCommerce PRO system passes the contents of these inputs directly to you if you have set the **usepost** flag within your site optional inputs

Some form inputs are of `type hidden`, which means they pass information through, without receiving any customer response. Other inputs, mainly of `type text` gather data provided by your customer.

If the transaction is accepted (Step 7), the QuickCommerce PRO server requests a formatted transaction result page from the accepturl or declineurl that you provided among the required (hidden) inputs. This request is sent to your merchant server via an HTTP GET statement, the usual way to request a Web page.

The HTTP GET statement may include additional data, depending on if you included a hidden input with the **usepost** field with value=1 in the Step 1 HTML.

If you do **not** include a hidden input with the **usepost** field with value=1 in the Step 1 HTML, the only information your Merchant Commerce Server receives from the QuickCommerce PRO server in Step 7 is either the accepturl or the declineurl, requesting the appropriate page sent via an HTTP GET. You will **not** get the customer's name or the items ordered or anything else.

If you **do** include a hidden input with the **usepost** field with value=1, in the Step 1 HTML, **all** the inputs including required inputs, optional inputs and the merchant-defined inputs are included in the HTTP GET that requests either your accepturl or your declineurl. Your Merchant Commerce Server can then place this data into your database, and if you are using a dynamic transaction result page via a CGI script or other way, it can fold some of this data into that transaction result page.

Examples

This section defines the required and optional input fields. There is an example of HTML code for Step 1 and an example of Step 7 HTTP GET code sent with **usepost**.

Required Input Fields Used in Step 1

```
usegc = 1 (required)

ecxid = TEST0 (replace TEST0 with your QUICKCOMMERCE PRO Merchant
ID)

action = Preview
```

The itemid must precede all other item fields; it is the delimiter between items in the stream.

```
itemid = M1234 (model #, SKU, etc)
itemname = Purple Polkadot Widget
itemdesc = long text stream
itemprice = 12.34 (price in US dollars)
itemweight = 5 (weight in pounds)
itemquant = 1
itemtax = 0 or 1 (does sales tax apply to this item? 1 = yes)
itemcomments = text comments
```

Customer may use itemcomments field to specify size, color, etc.

Optional Input Fields Used in Step 1

```
accepturl = http://trans.quickcommerce.com/~ats/accepted.html
(location of the page to display if transaction is accepted –
above URL is a default you may use)
declineurl = http://trans.quickcommerce.com/~ats/declined.html
(location of the page to display if transaction is declined –
above URL is a default you may use)
```

```
usepost = 0 or 1 (1 = a POST is generated to the accept or
decline URL)
body = BGCOLOR=#00000 BACKGROUND="some URL"
title = This is the HTML <title></title>
header= HTML code appearing at the top of the page
footer= HTML code appearing at the bottom of the page
getshippinginfo=0 or 1 (1 = show shipping address fields)
```

Example Step 1 HTML

```
<form method=post action="https://trans.quickcommerce.com/cgi-
bin/ats.cgi">
<input type=hidden name=usegc value=1>
<input type=hidden name=action value=preview>
<input type=hidden name=accepturl
value="http://trans.quickcommerce.com/~ats/accepted.html">
<input type=hidden name=declineurl
value="http://trans.quickcommerce.com/~ats/declined.html">
<input type=hidden name=ecxid value=TEST0><br>
<input type=hidden name=body value="bgcolor=#FFFFFF">
<input type=hidden name=header value="<FONT COLOR=maroon SIZE=+2
FACE=Arial>Lotto Card Order Form</FONT>">
<FONT COLOR="maroon" SIZE="+2" FACE="Arial">
<HR width=85%>
<blockquote>
<FONT FACE="Arial">Order Lotto Cards For $4.99 Each (Plus Tax &
Shipping if applicable) </FONT>
</blockquote>
<center>
<table width=80% cellpadding=4>
<tr>
<td>
<center>
&nbsp;<i>Quantity:</i>&nbsp;</font>
<input type=hidden name=itemid value="Lotto1">
<input type=hidden name=itemname value="Lotto Card ">
<input type=hidden name=itemdesc value="New York State Lotto
cards">
<input type=hidden name=itemprice value="4.99">
<input type=text name=itemquant value="0" maxlength=2 size=2>
<input type=hidden name=itemtax value="1">
<input type=hidden name=itemweight value="0.00">
</td>
</tr>
</table>
</center>
<p>
<input type=submit value="Proceed to Checkout">
```

Example Step 7 HTTP GET with Usepost

```
<html>
<body>
<center>
<form method=post action=https://trans.quickcommerce.com/cgi-
bin/ats.cgi>
```

```
<h1>Joe's Discount Aircraft Shop</h1>
<input type=hidden name=usepost value=1>

<input type=hidden name=action value=preview>
<input type=hidden name=currency value=USD>

<input type=hidden name=accepturl
value="http://trans.quickcommerce.com/cgi-bin/test.cgi">
<input type=hidden name=declineurl
value="http://trans.quickcommerce.com/cgi-bin/test.cgi">
<input type=hidden name=ecxid value="TEST0">

<input type=hidden name=subid value="">

<input type=hidden name=header value="<h1>Joe's Discount
Aircraft Shop</h1>">
<input type=hidden name=getshippinginfo value=1>

<table width=80% cellpadding=4>
<tr>
<td>
      The unique design of the single-seat F-117A provides
exceptional combat capabilities. About the size of an F-15
Eagle, the twin-engine aircraft is powered by two General
Electric F404 turbofan engines and has quadruple redundant fly-
by-wire flight controls. Air refuelable, it supports worldwide
commitments and adds to the deterrent strength of the U.S.
military forces.
<p>
      The F-117A can employ a variety of weapons and is equipped
with sophisticated navigation and attack systems integrated into
a state-of-the-art digital avionics suite that increases mission
effectiveness and reduces pilot workload. Detailed planning for
missions into highly defended target areas is accomplished by an
automated mission planning system developed, specifically, to
take  advantage of the unique capabilities of the F-117A.
<p>
<i>Your discount price: $225.49</i>
&nbsp;&nbsp;Quantity:&nbsp;&nbsp;
<input type=hidden name=itemid value="F117">
<input type=hidden name=itemname value="Lockheed F-117A
ightHawk">
<input type=hidden name=itemdesc value="Single-Seat Attack and
Defense Suppression Aircraft">
<input type=hidden name=itemprice value="225.49">

<input type=text name=itemquant value="0" maxlength=2 size=2>
<input type=hidden name=itemtaxable value="1">
<input type=hidden name=itemweight value="1000.00">
</td>
</tr>
<tr>
<td>
      The B-1B's electronic jamming equipment, infrared
countermeasures, radar  location and warning systems complement
its low-radar cross-section and form an integrated defense
system for the aircraft.
```

```
<p>
      The swing-wing design and turbofan engines not only
provide greater range and high speed at low levels but they also
enhance the bomber's survivability. Wing sweep at the full-
forward position allows a short takeoff roll and a fast base-
escape profile for airfields under attack. Once airborne, the
wings are positioned for maximum cruise distance or high-speed
penetration.
<p>
      The B-1B uses radar and inertial navigation equipment
enabling aircrews to globally navigate, update mission profiles
and target coordinates in-flight, and precision bomb without the
need for ground based navigation aids.  Included in the B-1B
offensive avionics are modular electronics that allow
maintenance personnel to precisely identify technical
difficulties and replace avionics components in a fast,
efficient manner on the ground.
<p>
<i>Your discount price: $549.95</i>
&nbsp;&nbsp;Quantity:&nbsp;&nbsp;
<input type=hidden name=itemid value="B1B">
<input type=hidden name=itemname value="Rockwell B-1B Lancer">
<input type=hidden name=itemdesc value="Strategic bomber">
<input type=hidden name=itemprice value="549.95">
<input type=text name=itemquant value="0" maxlength=2 size=2>
<input type=hidden name=itemtax value="1">
<input type=hidden name=itemweight value="3500.00">
</td>
</tr>
<tr>
<td>
      Along with the B-52 and B-1B, the B-2 provides the
penetrating flexibility and effectiveness inherent in manned
bombers.  Its low-observable, or "stealth," characteristics give
it the unique ability to penetrate an enemy's most sophisticated
defenses and threaten its most-valued, and heavily defended,
targets. Its capability to penetrate air defenses and threaten
effective retaliation provide a strong, effective deterrent and
combat force well into the 21st century.
<p>
      The revolutionary blending of low-observable technologies
with high aerodynamic efficiency and large payload gives the B-2
important advantages over existing bombers.  Its low-
observability provides it greater freedom of action at high
altitudes, thus increasing its range and a better field of view
for the aircraft's sensors.  Its unrefueled range is
approximately 6,000 nautical miles (9,600 kilometers).  The B-
2's low observability is derived from a combination of reduced
infrared, acoustic, electromagnetic, visual and radar
signatures.  These signatures make it difficult for the
sophisticated defensive systems to detect, track and engage the
B-2. Many aspects of the low-observability process remain
classified; however, the B-2's composite materials, special
coatings and flying-wing design all contribute to its
"stealthiness."
```

```
<p>
<p>
<i>Your discount price: $849.95</i>
&nbsp;&nbsp;Quantity:&nbsp;&nbsp;
<input type=hidden name=itemid value="B2">
<input type=hidden name=itemname value="Northrop B-2
&quot;Spirit&quot;">
<input type=hidden name=itemdesc value="Strategic bomber">
<input type=hidden name=itemprice value="849.95">
<input type=text name=itemquant value="0" maxlength=2 size=2>
<input type=hidden name=itemtax value="1">
<input type=hidden name=itemweight value="5500.00">
</tr>
</table>
<p>
<input type=submit value="Proceed to Checkout">
<p>
<b><font size=-1>Note: Some aircraft may be subject to export
restrictions.</font></b>
</form>
</center>
</body>
</html>
```

Single Payment Submission (or Quicksale Method) with a Web form

Single Payment Submission (also known as the Quicksale Method) with a Web form gives you almost complete control over what your customers see on their browsers. Your programmers can customize your transaction pages with CGI scripts, Active Server Pages, Cold Fusion applications, and so on. This method can provide secure transactions, even if your Web page is not hosted on a secure server. However, your programmers must handle things correctly to ensure transaction security.

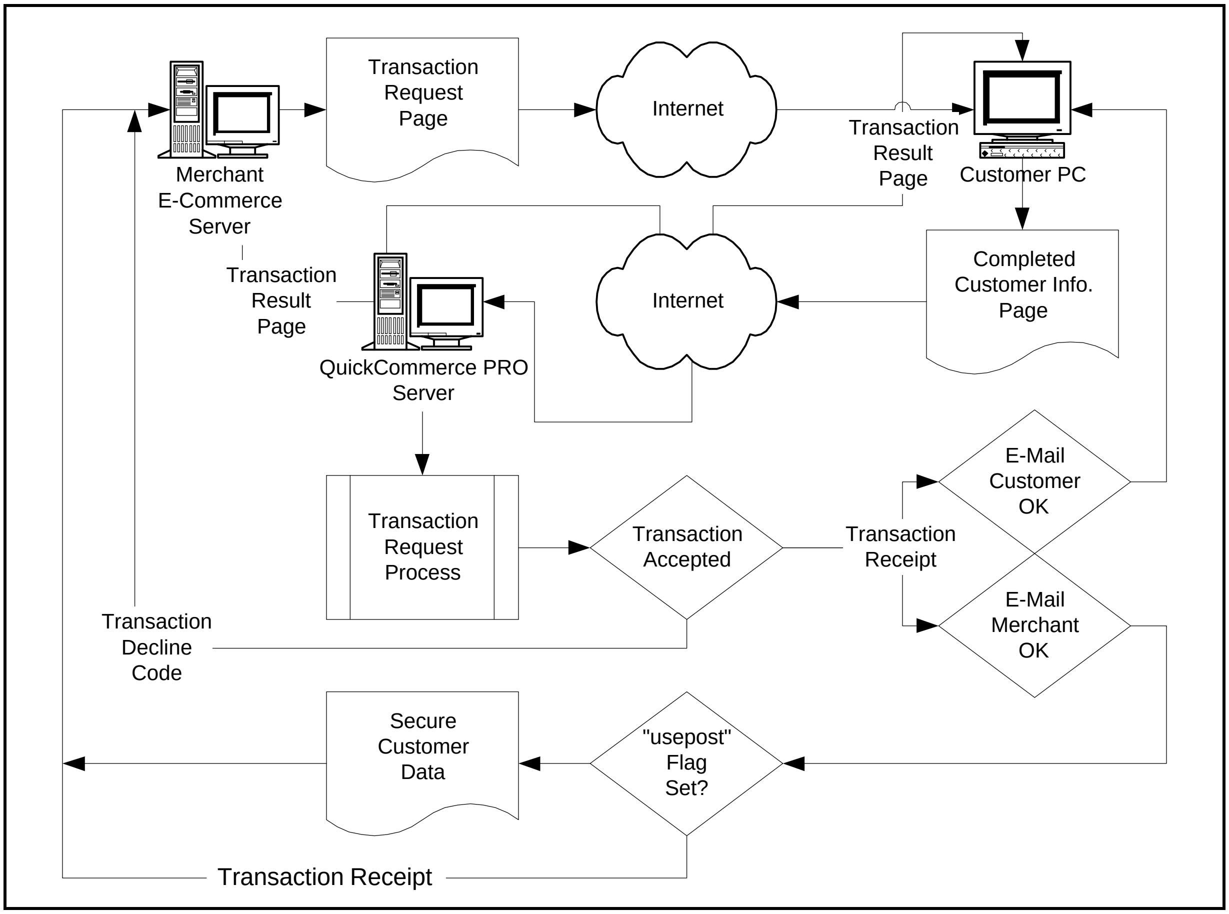
Your system gathers customer information and calculates shipping and tax in a manner controlled by your programmers. QuickCommerce PRO does the final transaction--obtaining the credit card number or bank account number from your customer and processing the transaction in a one-step secure dialog with your customer.

Your programmers must be proficient in HTML to integrate your e-commerce Web site.

How It Works

1. Your system displays the transaction request page, which asks your customer to supply either a credit card number or bank account number, to your customer's browser.
2. After your customer submits the account information, the data is securely transferred to QuickCommerce PRO, which processes the transaction request.
3. If the transaction is accepted, then QuickCommerce PRO generates a receipt for the transaction and e-mails it to you and to your customer. You can configure the system to **not** e-mail either you or your customer, or both.
4. QuickCommerce PRO securely sends the transaction result data to your system. At minimum, the data contains the accept/decline flag, the authorization code if accepted and the reason if declined.
5. Your programmers can take advantage of additional secure transaction data that QuickCommerce PRO can provide in this step, by setting the **usepost** flag.
6. Your system sends a transaction result page from either your accepturl or your declineurl to QuickCommerce PRO. Your programmers can use CGI or other methods to dynamically create this page.
7. QuickCommerce PRO passes the transaction result page to your customer's browser.

Figure 3 illustrates the Single payment processing with Web form process.



Security

The most sensitive transaction data like credit card number or bank account number is transmitted directly from the customer’s browser to the QuickCommerce PRO server via SSL. Security is provided, whether or not your Web site is hosted on a secure server. However, if you collect additional sensitive data, your server should be secure.

Customer data received by your system prior to your system asking the customer for a credit card number or bank account number (Step 1), which might be used to create the transaction request page, is transmitted securely only if you have a secure system.

If your system is not secure, it is best to **not** obtain sensitive data such as customer name and address prior to Step 1. Instead, obtain the information via the **usepost** flag in Step 5, which is secure because it is obtained via the secure QuickCommerce PRO system.

Getting Information About Your Customers

There are four ways to receive information about your customer's order:

1. Data your system obtains from your customer prior to your system asking the customer for a credit card number or bank account number (Step 1). Remember, this data can only be securely obtained if you have a secure server.
2. E-mail automatically generated and sent to you at the time of the transaction. Credit card and bank account numbers are omitted from e-mail receipts.
3. Data included at the time of the transaction, which your system can use to update your database. This data is only sent if your programmers set the **usepost** flag. This data is securely sent, even if your server is not secure.
4. Logging into your QuickCommerce PRO account some time after the transaction, and using the Transaction menu to generate reports.

What Your Programmers Do

When you use the Single Payment Submission (Quicksale Method) with a Web form, your programmers create all the pages leading up to and including the final transaction request page, updating your local database in the process with customer information. This transaction request page is transferred from your system to your customer's browser, where your customer supplies the credit card or bank account number.

The transaction request is then submitted from your custom designed transaction request page in your customer's browser directly to QuickCommerce PRO. Results are reported to your system, and your programmers provide a page for the final transaction results to display.

To create both the transaction request page and the transaction results pages, your programmers can use Web technologies, including CGI scripts, Active Server Pages, Cold Fusion applications, and so on.

Your programmers must be proficient in HTML if you select this transaction processing method.

Data Elements

Tables 2–5 describe required and optional data elements that can be **sent** to E-Commerce Exchange to complete a transaction with Single payment submission with a Web form processing.

Table 2 – Required data elements for Credit Card processing

Action=ns_quicksale_cc	This action instructs the QuickCommerce PRO Server to process a credit card order.
Ecxid=	Use TEST0 for testing. Change this to your correct Merchant ID for live transaction processing.
Amount=	The dollar amount of the transaction in the form of 0.00. All transactions are in US dollars.
Ccname=	The name of the consumer as it appears on the credit card.
Ccnum=	The credit card number of the consumer. Do not include spaces.
Expmon=	The expiration month on the consumer credit card. Use a numeral from 1-12 corresponding to the correct month.
Expyear=	The expiration year of the consumer's credit card in mmyy format

Table 3 – Optional data elements for Credit Card processing

Subid=	The E-Commerce Exchange merchant Sub ID. If unsure whether you have one, leave blank.

Authonly=	Set this value to equal 1 if you wish to pre-authorize the credit card only. A pre-authorization will "reserve" the amount specified in the amount field, it will not actually bill the customer's credit card.
Postonly=	Set this value to equal 1 if you wish to post authorize a credit card.
Accepturl=	The customer is transferred to this URL after the transaction is processed and approved. You may point to a CGI script. Example: http://www.myserver.com/accepted.html The script specified in Accepturl and Declineurl will always be called by a connection originating at trans.quickcommerce.com This field is **required** when using the HTML-based method. Do **not** set the return URL to https:
Declineurl=	The consumer is transferred to this URL after the transaction is processed and declined. You may point to a CGI script. Example: http://www.myserver.com/declined.html This field is required when using the HTML-based method. Do **not** set the return URL to https:
Usepost=	If your application requires consumer information to be returned in order to log it to a database, enabling the **usepost** option will fulfill this requirement. It is activated by specifying "usepost=1"
ci_companyname	The name of your company
ci_billaddr1	The billing address of the consumer
ci_billaddr2	The second line for the billing address of the consumer
ci_billcity	The consumer's city
ci_billstate	The consumer's state or province
ci_billzip	The consumer's Zip code or Postal code
ci_billcountry	The consumer's country
ci_shipaddr1	The consumer's shipping address
ci_shipaddr2	The consumer's second shipping address
ci_shipcity	The consumer's shipping city
ci_shipstate	The consumer's shipping state or province
ci_shipzip	The consumer's shipping Zip Code or Postal Code
ci_shipcountry	The consumer's shipping country
ci_phone	The consumer's phone number
ci_email	The consumer's email address
ci_memo	Miscellaneous information field
ci_dlnum	The consumer's driver's license number
ci_ssnum	The consumer's Social Security Number

emailto	The e-mail address to send the consumer e-mail receipt. If this tag is omitted, QuickCommerce PRO automatically uses the ci_email address as the default.
emailfrom	The return address on the consumer's e-mail receipt. If this tag is omitted, QuickCommerce PRO automatically uses mailto:null@quickcommerce.com as the default.
emailsubject	The subject line on the consumer's receipt email. If this tag is omitted, QuickCommerce PRO automatically uses *Payment Receipt #xzy* as the default message.
emailtext	The text in the body of the consumer's e-mail receipt. If this tag is omitted, QuickCommerce PRO automatically generates a generic receipt message as the default.

Table 4 – Required data elements for ACH processing

Action=ns_quicksale_check	Instructs the E-Commerce Exchange server to process a check order.
Ecxid=	Use TEST0 for testing. Change this to your correct Merchant ID for live transaction processing.
Amount=	Dollar amount of the transaction in the form of 0.00. All transactions are in U.S. dollars.
Ckname=	The name of the consumer as it appears on the checking account.
Ckaba=	Checking account routing or ABA number. This is a nine-digit numeric value. Do not include spaces.
Ckacct=	Checking account number. This is a numeric value of variable length. Do not include spaces.

Table 5 – Optional data elements for ACH processing

Subid=	The E-Commerce Exchange merchant sub ID. If unsure whether you have one, leave blank.
Accepturl=	URL that the consumer is transferred to after the transaction is processed and approved. You may point to a CGI script. Example: http://www.myserver.com/accepted.html The script specified in accepturl and declineurl is always called by a connection originating at trans.quickcommerce.com. This field is required when using the HTML method.
Declineurl=	The consumer is transferred to this URL after the transaction is processed and declined. You may point to a CGI script. Example: http://www.myserver.com/declined.html This field is required when using the HTML method.

ci_companyname	Name of your company.
ci_billaddr1	Billing address of the consumer.
ci_billaddr2	The second line for the billing address of the consumer.
ci_billcity	The consumer's city.
ci_billstate	The consumer's state or province.
ci_billzip	The consumer's zip code or postal code.
ci_billcountry	The consumer's country.
ci_shipaddr1	The consumer's shipping address.
ci_shipaddr2	The consumer's second shipping address.
ci_shipcity	The consumer's shipping city.
ci_shipstate	The consumer's shipping state or province.
ci_shipzip	The consumer's shipping zip or postal code.
ci_shipcountry	The consumer's shipping country.
ci_phone	The consumer's phone number.
ci_email	The consumer's e-mail address.
ci_memo	Miscellaneous information field.
ci_dlnum	The consumer's driver license number.
ci_ssnum	The consumer's Social Security number.
Emailto	The e-mail address where the consumer's e-mail receipt is sent. If this tag is omitted, QUICKCOMMERCE PRO uses the ci_email address as the default.
Emailfrom	The return address on the consumer's e-mail receipt. If this tag is omitted, QuickCommerce PRO uses null@quickcommerce.com as the default.
Emailsubject	The subject line on the consumer's receipt e-mail. If this tag is omitted, QuickCommerce PRO automatically uses Payment Receipt #xzy as the default message.
Emailtext	The text to display in the body of the consumer's e-mail receipt. If omitted, QUICKCOMMERCE PRO automatically generates a generic receipt message as the default.

Table 6 – Optional Recurring Billing Information

Recur_create	Creates a recurring billing record for a consumer. Set to recur_create=1 to create a recurring billing record.
Recur_billingcycle	This tag allows you to provide exact information on the billing cycle(s). For the recurring cycle you are implementing, set the value according to the values listed below: 0=No Recurring Billing Cycle 1=Weekly Recurring Cycle 2=Monthly Recurring Cycle 3=Quarterly Recurring Cycle 4= Semi-Annual Recurring Cycle 5= Annual Recurring Cycle 6=Bi-Weekly Recurring Cycle 7=Bi-Annual Recurring Cycle 8=Quad Weekly (28 day) Recurring Cycle

Recur_billingmax	Maximum number of times a consumer's account is re-debited through recurring billing. For example setting recur_billingmax =6 bills the consumer 6 times. -1= Unlimited number of times 0= No Recurring Billing
Recur_start	This value equals the number of days after an initial payment where the consumer is re-debited on a recurring cycle.
Recur_amount	Amount the consumer is to be re-debited on the recurring cycle. Do **not** use dollar signs.

Web Form Submission Data Formats

To send the transaction to the E-Commerce Exchange server via HTTP or HTTPS, post the fields to the following URL:

Using HTTP with SSL (Secure Socket Layer):

https://trans.quickcommerce.com/cgi-bin/ats.cgi

Using HTTP without SSL (no encryption):

http://trans.quickcommerce.com/cgi-bin/ats.cgi

> The **accepturl** and **declineurl** are required when using the HTML method**.** They can reference either static HTML pages or CGI scripts

Request Format

The following two examples show how to use HTML to transmit a transaction using HTTP and HTTPS. The QuickCommerce PRO distribution package contains several additional example HTML files, attached as zip files.

Sample HTML to Submit a Check Transaction Using HTTP(S)

```
<form method=post action=https://trans.quickcommerce.com/cgi-
bin/ats.cgi>

<input type=hidden name=action value="ns_quicksale_check">

<input type=hidden name=ecxid value="TEST0">

<input type=hidden name=amount value="1.00">

<input type=hidden name=ckname value="Joe Customer">

<input type=hidden name=ckaba value="123456789">

<input type=hidden name=ckacct value="123456789012345">

<input type=hidden name=accepturl
value="http://trans.quickcommerce.com/cgi-bin/showresult.cgi">

<input type=hidden name=declineurl
value="http://trans.quickcommerce.com/cgi-bin/showresult.cgi">

<input type=submit>

</form>
```

Sample HTML to Submit a Canadian Check Transaction Using HTTP(S)

```
<form method=post action=https://trans.quickcommerce.com/cgi-
bin/ats.cgi>

<input type=hidden name=action value="ns_quicksale_check">

<input type=hidden name=ecxid value="TEST0">

<input type=hidden name=amount value="1.00">

<input type=hidden name=ckname value="Joe Customer">

<input type=hidden name=ckaba value=" 12345-123">

<input type=hidden name=ckacct value="123456789012345">

<input type=hidden name=accepturl
value="http://trans.quickcommerce.com/cgi-bin/showresult.cgi">

<input type=hidden name=declineurl
value="http://trans.quickcommerce.com/cgi-bin/showresult.cgi">

<input type=submit>

</form>
```

Sample HTML to Submit a Credit Card Transaction:

```
<form method=post action=https://trans.quickcommerce.com/cgi-
bin/ats.cgi>

<input type=hidden name=action value="ns_quicksale_cc">

<input type=hidden name=ecxid value="TEST0">

<input type=hidden name=amount value="1.00">

<input type=hidden name=ccname value="Joe Customer">

<input type=hidden name=ccnum value="5454545454545454">

<input type=hidden name=expmon value="01">

<input type=hidden name=expyear value="1999">

<input type=hidden name=accepturl
value="http://trans.quickcommerce.com/cgi-bin/showresult.cgi">

<input type=hidden name=declineurl
value="http://trans.quickcommerce.com/cgi-bin/showresult.cgi">

<input type=submit>

</form>
```

Response Format

When a transaction is processed, QUICKCOMMERCE PRO retrieves the accepted or declined URL directly from the Merchant Commerce Server, then relays it to the customer's browser. The customer's browser is connected to the QUICKCOMMERCE PRO secure system during the entire transaction process event. The mechanics of this process are:

1. After a consumer's order is processed, QUICKCOMMERCE PRO initiates a TCP/IP connection between the QUICKCOMMERCE PRO server and the Merchant system.
2. An HTTP GET command retrieves the accepted or declined URL.
3. If the Merchant's e-commerce application includes a query string as part of the accepted or declined URL, QUICKCOMMERCE PRO appends its own responses to the end of the query string. This allows the Merchant to pass as much additional information in the query string as desired.
4. HTTP GET command output is sent to the customer's browser. If the accepted or declined URL references a static HTML page, it is displayed. If the accepted or declined URL references a CGI script, the script output is displayed.

   QUICKCOMMERCE PRO uses the following formats to return the query string response to the consumer's browser:

Transaction is Accepted

If the transaction is accepted, the following format is used:
http://www.myserver.com/cgi-bin/mycgi?Status=Accepted&AuthNo=AUTHCODE

The authorization code is the alphanumeric value returned by the credit card processing network, representing a receipt of each individual transaction.

Transaction is Accepted and Consumer Info is Returned

If the **usepost** flag is enabled, the POST sent to the **accepturl** is transmitted in the following format:

```
Version=1
Status=Accepted
AuthNo=######
PostedVars=BEGIN
action=ns_quicksale_check
ecxid=TEST0
amount=1.00
ckname=Joe+Customer
usepost=1
accepturl=http%3A//trans.quickcommerce.com/cgi-bin/ret-cgi
declineurl=http%3A//trans.quickcommerce.com/cgi-bin/ret-
cgi
PostedVars=END
```

The POST body is encoded as above. Notice that the entire original POST information sent to SPX is listed between `PostedVars=BEGIN` and `PostedVars=END`. All consumer financial information (account numbers) is **not** sent.

Transaction is Declined

If the transaction is declined, the following format is used:

http://www.myserver.com/cgi-in/mycgi?Status=Declined&Reason=Reason

The reasons for a declined transaction are numerous and depend on the credit card or check processing network through which your account is routed. QuickCommerce PRO does generate a number of internal declined reasons. They are:

"Invalid first name"
"Invalid last name"
"Invalid checking account number"
"Invalid checking account number, enter only digits"
"Invalid ABA/Routing Number"
"Invalid ABA/Routing Number, must be nine characters in length
"Negative Entry"
"Database unavailable"
"Transaction processing sub-system failed"
"Address Verification Failed"

Transaction is Declined and Consumer Info is Returned

If the usepost flag is enabled, the POST sent to the declineurl is transmitted in the following format:

```
Version=1
Status=Decline
Reason= (decline reason goes here)
PostedVars=BEGIN
action=ns_quicksale_check
ecxid=TEST0
amount=1.00
ckname=Joe+Customer
usepost=1
accepturl=http%3A//trans.quickcommerce.com/cgi-bin/ret-cgi
declineurl=http%3A//trans.quickcommerce.com/cgi-bin/ret-
cgi
PostedVars=END
```

The POST body is encoded as above. Notice that all of the original POST information sent to SPX is listed between `PostedVars=BEGIN` and `PostedVars=END`. All consumer financial information (account numbers) is **not** sent.

Single Payment Submission (or Quicksale Method) with a Socket Connection

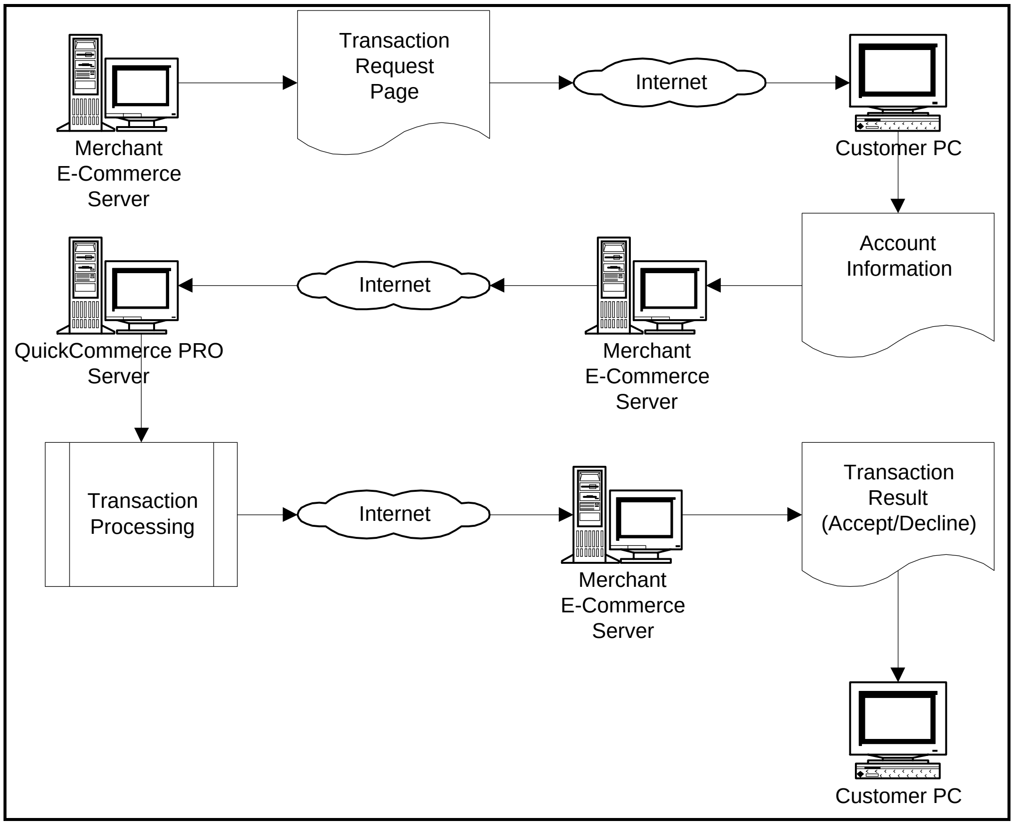
This method gives you 100% control over what your customers see. All QuickCommerce PRO server interactions are through your system. The QuickCommerce PRO URL never appears on your customer’s browser. This method requires advanced programming skills since your programmers create *all* the transaction pages.

Your system must be a secure server. Communication between your system and the QuickCommerce PRO system is done in single transactions via a direct socket, using HTTP. A socket connection supports interactivity with your customer. Your system submits a single transaction request on behalf of your customer, and gets an immediate transaction result back.

How It Works

1. Your system transfers the transaction request page, which asks your customer to supply either a credit card number or bank account number.
2. After your customer fills in the account information and clicks **Submit**, the transaction request data is transferred back to your system.
3. Your system sends the transaction request data to the QuickCommerce PRO server via the direct socket connection. The QuickCommerce PRO server then processes the transaction request.
4. The QuickCommerce PRO server sends the transaction result data back to your system via the socket connection.
5. Your system creates a transaction result page and transfers it to your customer’s browser.

Figure 4 illustrates Single Payment Submission flow using a socket connection.



Security

All aspects of transaction security are dependent on the Merchant Commerce Server. If you choose the socket connection method, your server must be secure with SSL to protect your transmissions. You must then decide upon a method to secure transaction request data over the socket connection to the QuickCommerce PRO server.

To secure data transmitted between your system and your customer's browser, your system must be a secure server using SSL. However, the SSL provided by your system will **not** secure the transmissions between your system and the QuickCommerce PRO server. Even though these transmissions **are** done over the Internet, they are **not** done over the World Wide Web and do **not** pass through your secure system. They use HTTP independent of your system.

Getting Information About Your Customers

You can receive information about your customer's order in three ways:

1. Data your system obtains from your customer prior to asking for a credit card or bank account number (Step 1). This data can be securely obtained only if you have a secure server.
2. Data included when QUICKCOMMERCE PRO sends the transaction request data back to your system (Step 4), which your system can use to update your database.

3. Logging into your QuickCommerce PRO account some time after the transaction, and using the Transaction menu to generate reports.

What Your Programmers Do

Your programmers have more responsibility than with the other transaction processing methods. They create the entire interface to your customer's browsers and must follow specific protocols in communicating with the QuickCommerce PRO server. Your programmers must have advanced skills and proficient in HTTP.

Socket Connection Data Formats

To prepare to transmit a transaction via the direct socket post method, you must first initiate a socket connection to the following URL: trans.quickcommerce.com

Use **Port 80** for http. Use **Port 443** for https (Secure Socket Layer).

Request Format

When you establish a socket connection to the E-Commerce Exchange server, you are ready to transmit the transaction information.

The next section contains two sample streams, which illustrate how to transmit an electronic check and a credit card transaction using a direct socket post. To test:

1. Open a TCP/IP session to the host listed above using a Telnet client.
2. When connected, cut and paste the text into the Telnet window replacing checking account or credit card numbers with valid account information.
3. Set the Content-length field to the exact number of characters in the HTTP request body (the line that starts with "action="). For example, if the line contains exactly 102 alphanumeric characters, the entry appears as Content-length: 102.
4. Wait for a response from the server.

Terminate all lines by a carriage return/line feed (**CR/LF)**. Be careful not to split credit card or checking account information with line breaks or the server generates an error response.

Sample HTTP Request to Submit an Electronic Check (ACH) Transaction

```
POST /cgi-bin/ats.cgi HTTP/1.0

Content-type: application/x-www-form-urlencoded

Content-length: 101

action=ns_quicksale_check&ecxid=TEST0&amount=1.00&ckn
ame=John%20Doe&ckaba=99999999&ckacct=9999999999&
```

Sample HTTP Request to submit a credit card transaction

```
POST /cgi-bin/ats.cgi HTTP/1.0

Content-type: application/x-www-form-urlencoded

Content-length: 111

action=ns_quicksale_cc&ecxid=TEST0&amount=1.00&ccname
=John%20Doe&ccnum=5454545454545454&expmon=01&expyear=
2001&
```

To use the direct socket post method to transmit transactions, you must be fully proficient with the HTTP protocol.

If you are uncertain as to how to correctly use the direct socket post method, use the HTML (Web Form) Submission Method.

Response Format

After a transaction is submitted via the direct socket submission method, the QuickCommerce PRO server responds with the results of the transaction within a few seconds.

It is up to your program to interpret the results of the transaction and route the consumer accordingly. Here are sample responses for accepted and declined transactions.

Transaction is Accepted

```
HTTP/1.1 200 OK
Date: Thu, 16 Jul 1998 01:23:15 GMT
Server: Stronghold/2.3 Apache/1.2.6 C2NetUS/2007
Connection: close
Content-Type: text/html
<html><body><plaintext>Accepted=0000000875
```

Transaction is Declined

```
HTTP/1.1 200 OK
Date: Thu, 09 Jul 1998 20:38:16 GMT
Server: Stronghold/2.3 Apache/1.2.6 C2NetUS/2007
Connection: close
Content-Type: text/html
<html><body><plaintext>Declined=Negative Entry
```

Transaction Request is in Error:

```
HTTP/1.1 200 OK
Date: Thu, 09 Jul 1998 20:38:16 GMT
Server: Stronghold/2.3 Apache/1.2.6 C2NetUS/2007
Connection: close
Content-Type: text/html
<html><body><plaintext>Error=Invalid Merchant
```

Batch Payment Submission with FTP

Batch processing is oriented toward the non-interactive approach to data processing. Your system accumulates a number of transaction requests (a batch), submits them all for processing, and then gets a return batch of transaction results. This is not a good approach if your customer is waiting online, but it is an excellent way to process a large number of recurring billings at the end of the month.

Batch processing gives you complete control over what your customers see. The QuickCommerce PRO URL never appears on your customer’s browser. Communication between your system and QuickCommerce PRO is done in batches using FTP. Your system must be a secure server.

This method requires advanced programmer skills.

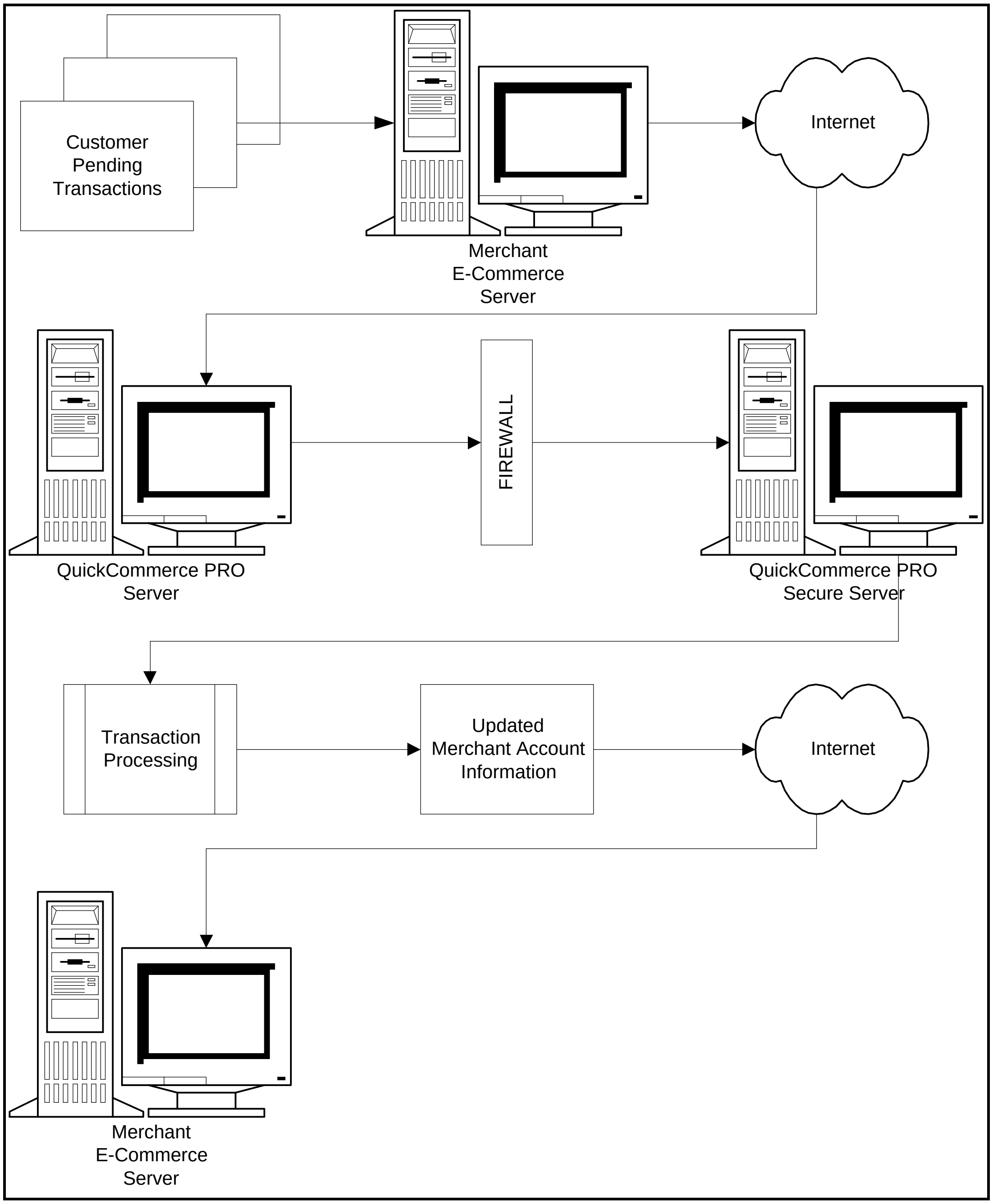
How It Works

Interactions between you and your customer browser are not shown in Figure 7, because this process assumes the interactions to have accumulated into the transaction requests comprising the batch file.

1. You prepare a batch file of transaction requests, consisting of the data elements.
2. Your system connects to the QuickCommerce PRO system server via the Internet using the following URL: ftp.quickcommerce.com
3. Select the \Uploads directory and transmit the file using FTP.
4. QuickCommerce PRO scans the \Uploads directory several times a minute and moves any files found behind the system firewall for processing.
5. Batch files begin processing within one minute of being submitted, but the time required to process the whole file will depend on the number of records in it. The results of the processing are available in the regular Transaction Report.
6. The QuickCommerce PRO server returns a batch file containing transaction result data for the transactions to your system. This return batch is currently only implemented for a limited number of customers.

Improperly formatted or incomplete records you inadvertently place in the batch file are discarded, and do not show up in any reports.

Figure 5 illustrates the Batch payment submission with FTP.


Customer Pending Transactions
Merchant E-Commerce Server
Internet
QuickCommerce PRO Server
FIREWALL
QuickCommerce PRO Secure Server
Transaction Processing
Updated Merchant Account Information
Internet
Merchant E-Commerce Server

File Format

The batch transaction request file is an ANSI-format text file with one transaction per line and a single TAB (ANSI 9) character separating each field. Fields may be left blank, but still included in the file. You must include the tab separators delimiting an empty field in the file.

Each transaction starts with a Transaction ID and a Record Type. The Transaction ID is supplied by you, the Merchant, and is used to link the batch file record to your database. For example, you could pass a Customer ID or Account Number from your own software system as the Transaction ID. The Record Type tells QuickCommerce PRO what to do with the remaining fields on the line; the meaning of each field depends on the Record Type. See the Data element table 2-5 for more detail.

Because the data for each transaction is prefixed with a Record Type, you may submit different types of transactions in a single file. For example, you could submit a file containing both credit card and check sales and update the recurring database with new members.

The easiest way to create a file in the required format is to place the fields to be written into separate columns in Microsoft Excel. Save the file using the *Save As* command; when prompted for the name of the file, change the type from Microsoft Excel Workbook to Text (Tab Delimited).

Supported Record Types

The second field on each line of the batch file specifies the record type.

Table 7 – Required Data Elements for Credit Card Sales, **Table 8 – Required Data Elements for Check Sales** and **Table 8 – Required Data Elements for Check Sales** display the currently supported record types, and the fields that must be passed for them:

Table 7 – Required Data Elements for Credit Card Sales

Column	Field Name	Data Type	Size	Required	Value
1	TransactionID	char	32		Merchant-supplied
2	RecordType	char			ns_quicksale_cc
3	ECXID	char	5		
4	Amount	numeric	10,2		
5	CCName	char	64		
6	CCNum	char	18		
7	CCExpMonth	numeric	2,0		
8	CCExpYear	numeric	4,0		
9	SubID	char	5		
10	CICompanyName	char	64		
11	CIBillAddress1	char	64		
12	CIBillAddress2	char	64		
13	CIBillCity	char	32		
14	CIBillState	char	32		
15	CIBillZip	char	16		
16	CIBillCountry	char	32		
17	CIShipAddress1	char	64		
18	CIShipAddress2	char	64		
19	CIShipCity	char	32		
20	CIShipState	char	32		
21	CIShipZip	char	16		
22	CIShipCountry	char	32		
23	CIPhone	char	16		
24	CIEmail	char	64		
25	CIMemo	char	1024		
26	CIDriversLicenseNo	char	32		
27	CISocialSecurityNo	char	32		
28	EMailTo	char	64		
29	EMailFrom	char	64		
30	EMailSubject	char	64		
31	EMailText	char	1024		
32	RecurBillingCycle	numeric	1,0		
33	RecurBillingMax	numeric	5,0		
34	RecurStart	numeric	2,0		
35	RecurrAmount	numeric	10,2		

Table 8 – Required Data Elements for Check Sales

Column	Field Name	Data Type	Size	Required	Value
1	TransactionID	char	32		*Merchant-supplied*
2	RecordType	char		✓	ns_quicksale_check
3	ECXID	char	5	✓	
4	Amount	numeric	10,2	✓	
5	CKName	char	64	✓	
6	CKABA	char	16	✓	
7	CKAccount	char	18	✓	
8	SubID	char	5		
9	CICompanyName	char	64		
10	CIBillAddress1	char	64		
11	CIBillAddress2	char	64		
12	CIBillCity	char	32		
13	CIBillState	char	32		
14	CIBillZip	char	16		
15	CIBillCountry	char	32		
16	CIShipAddress1	char	64		
17	CIShipAddress2	char	64		
18	CIShipCity	char	32		
19	CIShipState	char	32		
20	CIShipZip	char	16		
21	CIShipCountry	char	32		
22	CIPhone	char	16		
23	CIEmail	char	64		
24	CIMemo	char	1024		
25	CIDriversLicenseNo	char	32		
26	CISocialSecurityNo	char	32		
27	EMailTo	char	64		
28	EMailFrom	char	64		
29	EMailSubject	char	64		
30	EMailText	char	1024		
31	RecurBillingCycle	numeric	1,0		
32	RecurBillingMax	numeric	5,0		
33	RecurStart	numeric	2,0		
34	RecurrAmount	numeric	10,2		

Table 9 – Required Data Elements for Recurring Import

Column	Field Name	Data Type	Size	Required	Value
1	Transaction ID	char	32		*Merchant-supplied*
2	Record Type	char		✓	ns_recurring_import
3	ECXID	char	5	✓	
4	SubID	char	5	✓	
5	CICompanyName	char	64	✓	
6	CIFirstName	char	64	✓	
7	CILastName	char	32	✓	
8	Address	char	64		
9	City	char	32		
10	State	char	32		
11	Zip	char	16		
12	Phone	char	16		
13	Email	char	64		
14	Account Name	char	64	✓	
15	Amount	numeric	10,2	✓	
16	AcctType	char	2	✓	CC or CK
17	AcctData1	char	16	✓	Routing # or Exp. Date
18	AcctData2	char	18	✓	Account # or CC #
19	CKMemo	char	1024		
20	CKDriversLicenseNo	char	32		
21	CKSocialSecurityNo	char	32		
22	RecurBillingCycle	numeric	1,0	✓	
23	RecurNextBillingDate	char	16	✓	
24	RecurBillingMax	numeric	5,0	✓	
25	MemberUserName	char	64		
26	MemberPassword	char	64		
27	MemberExpirationDate	char	10		MM/DD/YYYY

Recurring Add

The Recurring Add record is almost identical to the Recurring Import record, except it requires that the Transaction ID field be a non-empty, unique value. Without this, you would not be able to send Recurring Update or Recurring Cancel records, all of which match records in the database against the supplied Transaction ID field.

Table 10 – Required Fields for Recurring Add

Column	Field Name	Data Type	Size	Required	Value
1	Transaction ID	char	32	✓	*Merchant-supplied*
2	Record Type	char		✓	ns_recurring_add
3	ECXID	char	5	✓	
4	SubID	char	5	✓	
5	CICompanyName	char	64	✓	
6	CIFirstName	char	64	✓	
7	CILastName	char	32	✓	
8	Address	char	64		
9	City	char	32		
10	State	char	32		
11	Zip	char	16		
12	Phone	char	16		
13	Email	char	64		
14	Account Name	char	64	✓	
15	Amount	numeric	10,2	✓	
16	AcctType	char	2	✓	CC or CK
17	AcctData1	char	16	✓	Routing # or Exp. Date
18	AcctData2	char	18	✓	Account # or CC #
19	CKMemo	char	1024		
20	CKDriversLicenseNo	char	32		
21	CKSocialSecurityNo	char	32		
22	RecurBillingCycle	numeric	1,0	✓	
23	RecurNextBillingDate	char	16	✓	
24	RecurBillingMax	numeric	5,0	✓	
25	MemberUserName	char	64		
26	MemberPassword	char	64		
27	MemberExpirationDate	char	10		MM/DD/YYYY

Table 11 – Required Elememts for Recurring Update

Column	Field Name	Data Type	Size	Required	Value
1	Transaction ID	char	32	✓	*Merchant-supplied*
2	Record Type	char		✓	ns_recurring_update
3	ECXID	char	5	✓	
4	SubID	char	5	✓	
5	CICompanyName	char	64	✓	
6	CIFirstName	char	64	✓	
7	CILastName	char	32	✓	
8	Address	char	64		
9	City	char	32		
10	State	char	32		
11	Zip	char	16		
12	Phone	char	16		
13	Email	char	64		
14	Account Name	char	64	✓	
15	Amount	numeric	10,2	✓	
16	AcctType	char	2	✓	CC or CK
17	AcctData1	char	16	✓	Routing # or Exp. Date
18	AcctData2	char	18	✓	Account # or CC #
19	CKMemo	char	1024		
20	CKDriversLicenseNo	char	32		
21	CKSocialSecurityNo	char	32		
22	RecurBillingCycle	numeric	1,0	✓	
23	RecurNextBillingDate	char	16	✓	
24	RecurBillingMax	numeric	5,0	✓	
25	MemberUserName	char	64		
26	MemberPassword	char	64		
27	MemberExpirationDate	char	10		MM/DD/YYYY

Table 12 – Required Elements for Recurring Cancel

Column	Field Name	Data Type	Size	Required	Value
1	Transaction ID	char	32	✓	*Merchant-supplied*
2	Record Type	char		✓	ns_recurring_cancel
3	ECXID	char	5	✓	
4	SubID	char	5	✓	
5	CancelType	numeric	1,0	✓	0=Immediately, 1=Next Billing, 2=Scrub

Table 13 – Sample Record Format for Generic Sale

Column	Field Name	Data Type	Size	Required	Value
1	Transaction ID	char	32	✓	*Merchant-supplied*
2	Record Type	char		✓	ns_quicksale
3	ECXID	char	5	✓	
4	SubID	char	5	✓	
5	CICompanyName	char	64	✓	
6	CIFirstName	char	64	✓	
7	CILastName	char	32	✓	
8	Address	char	64		
9	City	char	32		
10	State	char	32		
11	Zip	char	16		
12	Phone	char	16		
13	Email	char	64		
14	Account Name	char	64	✓	
15	Amount	numeric	10,2	✓	
16	AcctType	char	2	✓	CC or CK
17	AcctData1	char	16	✓	Routing # or Exp. Date
18	AcctData2	char	18	✓	Account # or CC #
19	CKMemo	char	1024		
20	CKDriversLicenseNo	char	32		
21	CKSocialSecurityNo	char	32		

Security

The FTP batch processing method is not secure! Currently, QuickCommerce PRO does not provide a means for secure data transmission via FTP. Therefore, you can assume that customer data transmitted from your server to the QuickCommerce PRO system server is subject to interception by third parties.

When your batch file is received by the QuickCommerce PRO system, it is processed utilizing secure methods. Transaction authorizations are accomplished in exactly the same manner as secure Web-based procedures. Transaction authorizations post to your QuickCommerce PRO Merchant account and are password-protected against unauthorized viewing.

Getting Information About Your Customers

When you use batch, you supply all customer information, except transaction authorization data; therefore, it is presumed that your customer information is immediately available to support your local requirements. Notwithstanding, you can logon to your QuickCommerce PRO system account and use the various system utilities to view the standard reports or to configure and produce whatever specialized reports you might require:

What Your Programmers Do

When you use the FTP Batch processing, your programmers have more responsibility than with the other methods of transaction processing. They create the entire interface with your customers' browsers and must follow specific protocols in preparation of the batch file.

Batch Payment Submission with E-mail

This method is precisely the same as the FTP transmission method just described. This method is less secure that the FTP transmission method!

As e-mail is transmitted over the Internet, packets including attachments may be present on an indeterminate number of servers from which they can be easily read or altered.

How It Works

1. Prepare a tab-delimited ANSI batch file, in accordance with the same requirements stated for the FTP transmission method, and e-mail to automation@quickcommerce.com
2. Zip the file using the PKZIP™ compression program. Self-extracting zip files with the .exe extension will reject and not be processed.
3. The QuickCommerce PRO server parses the ATS mailbox every five minutes. When it finds a message, the parse program opens the attachment and moves its contents behind the secure firewall for processing. The remainder of the process is identical to the FTP submittal process.

Security

E-mail submission method is the least secure method of submitting requests for transaction processing.

Getting Information About Your Customers

When you use the batch method, you supply all customer information, except transaction authorization data; therefore, it is presumed that your customer information is immediately available to support your local requirements. Notwithstanding, you can logon to your QuickCommerce PRO system account and use various system utilities to view standard reports or to configure and produce the specialized reports you require.

What Your Programmers Do

Like FTP batch submission, your programmers have more responsibility than with the other methods of transaction processing. They create the entire interface to your customers' browsers and must follow specific protocols in preparing the batch file.

Conclusion

After you have integrated QuickCommerce PRO into your e-commerce application, you can test your interface by using **TEST0** as the ECX ID. Every merchant is authorized to use the **TEST0** ID. With **TEST0**, your programmers can enter valid credit card and bank account numbers and simulate transactions in the system, but no money will actually flow.

After satisfactorily testing the interface, make sure you change your ECX ID from **TEST0** to the correct ECX ID for your account.

Then contact E-Commerce Exchange Technical Support for final verification that you have indeed set your application to process live transactions. This step is **vital**. If you do not do so, your customers will process transactions under test mode, but no money will flow. E-Commerce Exchange is not responsible for lost sales due to your system being set in the test mode.

For additional information or clarification in installing the necessary scripts on your server, contact E-Commerce Exchange Customer Service at 800-757-6318 or e-mail: support@ecx.com.