Jenzabar CX

Authorize.Net

Reference Guide

Filename: inauth
Distribution date: 11/30/2001

Contact us at www.jenzabar.com

JENZABAR, INC.
AUTHORIZE.NET REFERENCE GUIDE

TABLE OF CONTENTS

GETTING STARTED

Introduction

What is Authorize.Net?

Authorize.Net is a real-time transaction processing system that functions as a payment gateway service using a secure transaction server on the Internet. A real-time system will perform all processing automatically from the Web site 24 hours a day, 7 days a week. Merchants with a valid merchant account can use Authorize.Net to submit, authorize, capture, and settle credit card transactions without the need for a separate transaction terminal or processing software. Automated Clearing House (ACH) collections, also known as Electronic Checks, are also supported on the Authorize.Net system.

Authorize.Net works in conjunction with a merchant account supplied by a Merchant Service Provider. This allows you, as the merchant, to accept credit card payments securely and in real-time over the Internet. All of our Certified Agents are Merchant Account Providers and are able to provide you with both a merchant and Authorize.Net account.

Authorize.Net has rapidly become a leading provider of Internet-based transaction services with thousands of online and traditional business customers around the world. Authorize.Net Corporation has also formed strategic alliances with leading financial institutions and technology partners to deliver the most comprehensive online authorization and processing services in the industry. Authorize.Net's services can only be obtained from banks (or ISOs partnered with banks, such as Jenzabar).

Credit Cards and the Internet

It is important to understand that there is really no such thing as processing credit cards over the Internet. None of the major processing networks (such as Nova, Paymentech, FDC/FDR, Vital, etc.) are connected to the Internet. All credit card processing is done by dedicated dial-up modem and is accessed by a terminal, software, or larger systems found on mainframes. There are, however, real-time systems available to process an approval while the user is at your Web site.

If a school does not want any manual tasks associated with credit card processing, they should opt for a real-time system. A real-time system consists of everything needed to conduct online transactions. These systems will capture the credit card information from the Web site where the user enters the information, downloads it to the Host server, transmits it on to the processing center, obtains the approval (or decline), and then sends a confirmation message back to the user at the Web site, all in a matter of seconds.

Purpose of This Guide

This guide serves as a learning tool and reference guide for setting up and supporting Authorize.Net functionality to work with Jenzabar CX.

Intended Audience

This guide is intended for the system users in your institution’s computer center. System users include the Jenzabar coordinator, Jenzabar system administrator, and programmer/analyst.

Real-Time Web Transactions Process

Overall Process

The following steps outline how real-time Web transactions are processed by Authorize.Net:

- Users enter their credit card information onto a payment submission form. The form needs to be set up to run through a Secured Server (SSL) so that the information is encrypted.
- The encrypted card information flows to the processing center/payment gateway.
- The processor/bank verifies the card information and either approves or declines the cardholder within seconds. An onscreen receipt (or declined message) is displayed, and an e-mail receipt (or declined message) is also sent to both the user and the site administrator.
- If the cardholder is approved, the amount is moved from the cardholder's bank to the merchant's processing bank. The merchant's processing bank then moves the money to the merchant's checking account. The transaction is now complete. Usually, the money will appear in the in any credit card transaction, and is the step that ensures the merchant is paid for the goods or services. It is fairly simple in concept. All transactions ready for settlement (sometimes referred to as posting) are sent as records to the processing host, where they are stored in a capture database.

It is the responsibility of the interface system (such as Authorize.Net) to ensure reliable transport to the host as well as data content as required by the credit card and banking industries. Authorize.Net provides the services and software (ADC and Virtual Terminal) to ensure authorization and capture of funds.

AUTHORIZE.NET COMPONENTS AND PROCESSES

Components

Two Services

Authorize.Net includes the following two separate and distinct services; both services are included with your Authorize.Net account.

- AuthorizeNet Virtual Terminal, used directly by merchants for manual processing
- AuthorizeNet ADC method, used by a Web site for online transactions

AuthorizeNet Virtual Terminal

Virtual Terminal is hosted completely on the Authorize.Net transaction servers, where merchants simply log in using their favorite Web browser and perform live transactions using their merchant account. Virtual Terminal replaces standard authorization terminals or software and provides the best solution for merchants who manually enter credit card transactions for mail or phone order sales.

AuthorizeNet ADC Method

Automated Direct Connect (ADC) is the provision that allows a merchant to link their Web site with Authorize.Net in order to accept credit card payments from customers in real-time with complete automation. It provides a simple and straightforward mechanism to link simple and more complex Web sites with the Authorize.Net transaction server, including support for the merchant's own custom programming.

The ADC method enables a school to authorize, process, and manage credit card transactions without the complicated software, hardware, and expertise normally associated with processing payments over the Web. Institutions only need a computer with an Internet connection and a Web browser; expensive hardware or software is not required. The institution does not need to be concerned with security, downtime, or making sure electronic transactions are processed and settled correctly.

When online users are ready to make payments from the Web, ADC captures the necessary information (name, credit card number, etc.) from the campus Web's secure payment form and submits it to Authorize.Net. This information is encrypted and sent to the transaction server. The server then sends the data through the authorization network to the appropriate card issuer's bank over a secure proprietary connection. When the authorization process is complete (this takes approximately five seconds), the user receives an approval or decline response and the server stores the transaction. Transactions are automatically settled each day.

The Jenzabar CX Web products use the ADC method of integration with the Authorize.Net system.

Types of ADC Methods

Of the two types of ADC methods (Direct Response and Relay Response), Jenzabar CX’s first version of Authorize.Net usage works exclusively with Relay Response.

Relay Response causes the system to pass information in the first connection, while a second connection returns the confirmation.

In the Direct Response method, currently not in use, the information is passed and waits for confirmation through a single connection.

Jenzabar CX and the Automated Direct Connect Method

Integration Method

In ADC Relay Response, the customer's interaction is with the system's gateway server. Your Web page initiates a transaction by creating an HTML form that posts the required transaction information to the gateway server. The customer provides additional information to the gateway server as required or desired. The gateway server processes the transaction, and then transmits the results of the transaction to the merchant's server via HTTP form POST. The merchant's server can respond back in an appropriate manner, based on the results of the transaction. The response back from the merchant's server is sent back to the gateway server, which then relays it on to the customer's browser as if it came directly from the gateway server.

Process Flow

The following details the flow of control when using ADC Relay Response:

- The user fills in the payment information on Jenzabar CX's payment form and submits the data. A confirmation page displays, allowing the user to review the contents prior to sending it.
- The form information is sent to the Authorize.Net site where it is processed, and a response is sent back to the URL specified in the x_ADC_URL form field.
- If the response is a success, then an entry is posted to Jenzabar CX via the *filepost* process (see Appendix A). If the response is a failure, a message displays to explain the failed request to the end user.

The Relay Response Method and Your Firewall

You must be able to supply an accessible return address outside your firewall to be able to implement the ADC Relay Response method. Such a return address can be set up on your external router/gateway via "inbound mapping". For more information in the setup, see http://www.howstuffworks.com/nat.htm. Jenzabar recommends that you use SSL to transmit the sensitive credit card information.

Maintaining Security

Introduction

Security is essential in the use of Authorize.Net. The following components provide the required level of security:

- Authentication
- Message privacy
- Message integrity
- Secure sockets layers
- Digital certificates
- Certificate authority
- Firewall configuration

Background

In physical transactions, the challenges of identification, authentication, and privacy are solved with physical marks, such as seals or signatures.

In electronic transactions, the equivalent of a seal must be coded into the information itself. By checking that the electronic seal is present and has not been broken, the recipient can confirm the identity of the message sender and ensure that the message content was not altered in transit. To create an electronic equivalent of physical security, advanced cryptography is used.

Once activated by digital certificates, SSL immediately begins providing the required components of secure online transactions.

Authentication

By checking the digital certificate, users can verify to whom the Web site belongs. This bolsters their confidence in submitting confidential information.

Message privacy

SSL encrypts all traffic between their Web server and customers, using a unique session key. To securely transmit the session key to the consumer, the server encrypts it with their public key. Each session key is used only once during a single session with a single customer. These layers of privacy protection ensure that information cannot be viewed if unauthorized parties intercept it.

Message integrity

When a message is sent, the sending and receiving computers each generate a code based on the message content. If even a single character in the message content is altered, the receiving computer will generate a different code and then alert the recipient that the message is not legitimate. With message integrity, both parties involved in the transaction know that what they are seeing is exactly what the other party sent.

Secure Sockets Layer (SSL)

This is the technology you need to establish a secure channel for online transactions and should already be enabled on your server. The only requirement for activation is a digital certificate. SSL becomes functional only after you install a digital certificate. SSL employs the essential functions of authentication, data encryption, and data integrity for secure transactions and establishes a secure communications channel between your server and your customer's browser. SSL comes in two strengths, 40-bit and 128-bit, which refers to the length of the session key generated by every encrypted transaction. The longer the key, the more difficult it is to break the encryption

code. Most browsers support 40-bit SSL sessions, and the latest browsers enable users to encrypt transactions in 128-bit sessions.

Digital Certificates (Secure Server IDs)

By using a digital certificate, the information that is being sent cannot be intercepted or decrypted by anyone other than the intended recipient. Digital certificates work in conjunction with the SSL technology that is a standard part of most Web server and Web browser packages. When you obtain and install a digital certificate, you enable the use of SSL at your Web site. When a browser connects to a site with a digital certificate, the browser automatically uses the certificate to verify that it is doing business with a legitimate site. The browser and the server then use the public key contained within the certificate to encrypt all information that passes between them.

Certificate Authority (CA)

This component is responsible for issuing, revoking, renewing, and providing directories of digital certificates. CAs must take steps to establish the identity of the people or organizations to which they issue IDs. Once the CA establishes an organization's identity, it issues a certificate that contains the organization's public key and signs it with the CA's private key. For more information, see *Private and Public Keys* in this document.

Firewall Configuration

The digital certificate enrollment requires that you can make both HTTP and HTTPS connections.

NOTE: SSL imposes some performance overhead. Therefore, most server software applications allow you to apply SSL selectively to Web pages that require encryption, such as payment information pages. For example, there is no benefit from applying SSL to general information pages.

Providers

Verisign, Inc., of Mountain View, California, is the leading provider of digital certificate solutions used by enterprises, Web sites, and consumers to conduct secure communications and transactions over the Internet and private networks. VeriSign is the world's leading CA, having issued over 3,000,000 Digital IDs to individuals for use in identifying themselves on the Internet. VeriSign has also issued more than 75,000 IDs for use on servers, which enable people to conduct secure and authenticated e-commerce and other forms of communication with those servers. The Public Key Infrastructure that VeriSign has helped establish for the Internet will secure billions of dollars in transactions this year.

Thawte is another leading CA and is a subsidiary of Verisign.

Cost of VeriSign Services and Products

For complete VeriSign pricing details, visit:
http://www.verisign.com/products/site/secure/index.html

For complete Thawte pricing details, visit: http://www.thawte.com/pricing.html

Encryption

Standard practice enables 128-bit SSL encryption with domestic-version Microsoft and Netscape browsers and industry 40-bit SSL with export-version browsers. U.S. encryption laws stipulate that browser software will usually only enable SSL connections at 56-bit encryption. However, if the Web server presents a strong encryption certificate, the browser will connect at 128-bit encryption.

In the complex world of encryption technology, security levels increase with the bit count. A large company with computer equipment worth $1 million can crack a weakly encrypted message

within hours. By contrast, a message encrypted with a 128-bit key is considered completely impenetrable by any organization or government using today's technology.

Session Keys

The 128-bit or 40-bit connection refers to the session key. This is a symmetric key created by the browser when it connects to the server that is used to encrypt and decrypt data (transmitted to and from the server) after the initial browser/server handshake. If your server supports full strength sessions and the browser connecting to your site supports 128-bit, then a 128-bit session key (128-bit connection) will be created and used. Browsers that have been exported from the U.S. are limited to creating 40-bit session keys. Browsers that have been distributed within the U.S. or manufactured by companies outside of the U.S. can create 128-bit session keys and thus connect to similarly manufactured and distributed servers in full strength cryptography.

Private and Public Keys

Digital certificate technology employs advanced public-key cryptography, *Public Key Infrastructure (PKI).* In public key cryptography, an individual or organization has two complimentary keys, a public key, and a private key. Any information encrypted using the private key can only be decrypted using the public key. Conversely, any information encrypted using the public key can only be decrypted using the private key. Rather than using the same key to both encrypt and decrypt data, a digital certificate uses a matched pair of keys that uniquely complement each another.

When a key pair is generated for the school, the private key is installed on their server and nobody else has access to it. Their matching public key, in contrast, is freely distributed as part of their digital certificate. They can share it with anyone and even publish it in directories. Customers or correspondents who want to communicate with the school privately can use the public key in the school's digital certificate to encrypt information before sending it to them. Only the school can decrypt the information, because only they have the private key. The digital certificate contains the school's name and identifying information and their public key. It tells students and correspondents that the public key belongs to the school.

Refer to Appendix D for answers to some commonly asked questions regarding security.

EXCERPTS OF AUTHORIZE.NET DOCUMENTATION

Introduction

Information Included in This Section

The following excerpts were extracted from Authorize.Net documents found on their Web site, http://www.authorizenet.com. They have been included here to help you better understand the approach to implementing the online transaction process using Authorize.Net's system. All of the necessary coding is in place within the Jenzabar CX Web products. For information about the required configurations for the Jenzabar CX Web products, see the README associated with SMO 12650.

Note: Where applicable and for your convenience, comments regarding the specifics of your usage of Authorize.Net with Jenzabar CX are included throughout this section.

Purpose of Included Documentation

User institutions can use the information contained in this section to complete the following tasks so they can use Authorize.Net:

- Obtain a merchant account
- Secure the Web server with SSL
- Define macros that are specific to the institution's Web server
- Define macros for posting to a General Ledger account

Implementing Real Time Processing

Introduction

To implement real-time processing, you must:

- Obtain a merchant account
- Gain a secure server connection
- Define macros within the SMO 12650 README
- Test the connection
- Verify access to Virtual Terminal

Obtaining a Merchant Account

A merchant account allows you to accept major credit cards, electronic checks, and ATM/Debit cards. The bank deposits the daily credit sales (minus applicable fees) into your institution's account. All funds are directly transferred to the checking account of your choice. A merchant account is a contract relationship between a merchant (the school) and a merchant account provider (a processing bank or independent sales organization such as Jenzabar) for the clearing and settlement of credit card transactions. A merchant account is *not* a bank account. It does not carry a balance, and you do not deposit or withdraw from it.

A merchant account enables your school to process credit card transactions through a payment gateway (i.e., a processing center such as Authorize.Net), the route used to quickly transfer card information for processing and verifying online in real-time (usually within a matter of seconds). Authorize.Net's services also have the ability to process non real-time manual transactions. Detailed information is included with your merchant account. Obtain a merchant account from Jenzabar by contacting your Account Executive (AE) for an application and merchant account setup information.

Gaining a Secure Server Connection

You must obtain an SSL-enabled link to the campus Web server. It will encrypt confidential ordering data through the use of digital certificates.

Defining Macros Within the SMO 12650 README

Linking your school's campus Web online payment forms with the ADC method requires you to set the value of macro WEB_ENABLE_ONLINE_PAYMENT to 'Y'. If a student has an outstanding balance from the Course and Fee Statement, a button appears. When students click the **Make an Online Payment** button, they can complete the forms that send information to the Authorize.Net site.

Testing the Connection

Institutions can check the status of transactions or run reports on past activity by going to the Merchant Menu Login and logging on to their own password-protected area. To view this area in Test Mode, use login ID *inttest3* and password *testing3*.

Verifying Access to Virtual Terminal

From the Merchant menu, a school can also access the Virtual Terminal to enter payment information manually (for users who either call in a payment or pay in person with their credit card). Virtual Terminal is a manual processing system that works from a browser connected to the Internet. Virtual Terminal is included with Authorize.Net's services. These tools are available *only* through Authorize.Net.

Basic Integration Concepts

Introduction

Constructing a Perl script/HTML form, which does an HTTPS POST to https://secure.authorize.net/gateway/transact.dll, performs integration between a school's campus Web site and the Authorize.Net system. In this form are several hidden fields that are passed to the system to indicate information about the transaction that the system requires. Some of this information is required by Authorize.Net, while other information is needed by Jenzabar CX. Any custom hidden form variables that are sent can be returned from the gateway server back to the campus Web.

Fields for ADC Relay Response

The Relay Response does not require a direct connection from the requesting site to the Authorize.Net site. Hidden form fields are passed back to the URL specified by x_ADC_URL. The server specified must be a fully qualified address (e.g., *myserver.myschool.edu*). Also, it must be accessible through a firewall. The specified port could be opened on the firewall only if Authorize.Net is connecting.

```
<FORM METHOD=POST ACTION="https://secure.authorize.net/gateway/transact.dll">
<INPUT TYPE=HIDDEN NAME="x_Version" VALUE="3.0">
<INPUT TYPE=HIDDEN NAME="x_Login" VALUE="your merchant login ID goes here">
<INPUT TYPE=HIDDEN NAME="x_ADC_URL" VALUE="server:port/cgi-script">
<INPUT TYPE=HIDDEN NAME="x_ADC_Relay_Reponse" VALUE="TRUE">
<INPUT TYPE=HIDDEN NAME="x_Amount" VALUE="amount goes here">
<INPUT TYPE=TEXT NAME="x_card_num" VALUE="A card number goes here">
<INPUT TYPE=TEXT NAME="x_exp_date" VALUE="Expiration date goes here">
<INPUT TYPE=SUBMIT VALUE="Submit">
</FORM>
```

Testing the Authorize.Net Connection

Test Mode via Account Settings

Test mode is a special mode of interacting with the system that is useful during the initial setup phase when you may want to test your setup without processing live card data. When an account is set to Test mode, all transactions appear to be processed as real transactions; however, a payment processor is never contacted, so all transactions are approved. Transactions submitted in Test mode are not stored on the system and will not appear in any reports or lists. It is strongly suggested that you leave an account in Test mode until you are sure that all aspects of a merchant's interaction with the system are functioning properly to avoid possibly incorrectly charging real credit cards.

When an account is not in Test mode, it is required to make a secure encrypted connection with the system. This is necessary to protect the integrity of the live credit card data that is being sent across a public network. When an account is in Test mode, it is assumed that real credit card data is not being used and, therefore, connections are not required to be secure and encrypted. Therefore, it is the responsibility of merchants to ensure that they are not using real credit card data in their testing unless a secure encrypted connection to the system is being made.

One other key difference between Test mode and normal usage of the system is that batch processing is not supported in Test mode. Since the system verifies the format of the batch file as it is being uploaded, it becomes impossible to upload incorrect data. This makes batch uploading unnecessary in Test Mode.

To set an account to Test mode, select the **Test mode** check box in the **General Settings** area of the **Settings** menu.

x_Test_Request Using Web Site Source Code

When you put the *x_test request* form field in your source code, it only puts your Web site in Test mode and you can run live transactions through your Virtual Terminal. In other words, only the transaction that you send with the *x_test_request* form field will be considered to be a test while your account remains in live mode.

To put only your Web site in Test mode, set the **x_test_request** value to **TRUE** in the stuapost.cgi script. Once you are ready to go live with online processing, you must set this value to **FALSE**. Also, within your Merchant Menu settings, the value must be set to **FALSE**.

Using a Test Credit Card Number

Because a payment processor is never contacted in Test mode, all properly formatted transactions appear to be approved, even if invalid credit card numbers are used. There are many situations, however, where a developer will need the system to not approve transactions or to generate errors in order to test all possible responses from the system.

The system has been designed so that a special test credit card number, *4222222222222*, can be used to generate error responses from the system. To cause the system to generate a specific error, send a transaction with the card number **4222222222222** and an amount equal to the number of the error you want the system to return, as enumerated in Appendix C.

For example, if you send a transaction to the system in Test Mode with a credit card number of **4222222222222** and an amount of **12 dollars**, the system will respond with error **12**, "Authorization Code is required but is not present."

APPENDIX A – JENZABAR CX FILES USED FOR AUTHORIZE.NET

Scripts and Descriptions

Included Scripts

The following scripts are included with Authorize.Net:

Directory	Script names
modules/regist/cgi	Stuaform.cgi, stuapost.cgi, stuarslt.cgi
web/cgi/student/secure	Stuaform.cgi, stuapost.cgi, stuarslt.cgi
web/html/includes	Ccnumval.js, isReady.js

Script Descriptions

File: stuaform.cgi

The stuaform.cgi module is called by stubill.cgi module. This module generates the initial entry form. Default values for the current ID are placed in the form for submission. The ccnumval.js and isReady.js JavaScript modules are used by the generated html to validate the form entries.

File: stuapost.cgi

The stuapost.cgi module is called by stuaform.cgi module. This is the confirmation page that allows the user to review all the entries for correctness. The majority of the hidden form fields are declared here. The hidden form fields are used by Authorize.Net to process the transaction. The complete listing of Authorize.Net fields is in Appendix B.

File: stuarslt.cgi

The stuarslt.cgi module is called by the Authorize.Net process. At this point, the transaction response code indicates a failure or success. If the transaction is a success, then the transaction is posted to the GL via the *filepost* process. Macros can be changed in macros/custom/Web to indicate which accounts are to be used for the *filepost* process.

Customizations

If the Jenzabar CX Web Product configurations do not allow direct access to the server, then an alternative must be coded to capture the Authorize.Net transaction.

APPENDIX B – FUNCTIONAL REFERENCE AND RESULT FIELDS

Form Fields

Overview

In Authorize.Net Version 3.0, there are many ways to integrate a merchant's server with the system. This section provides a convenient reference to all of the types of information than can be sent to the system and all of the things that could be expected in return. All integration with the system is done by performing an HTML form POST to https://secure.authorize.net/gateway/transact.dll**.**

Table of Fields

The following table provides an alphabetical list of all of the possible values that will be recognized by Authorize.Net when sent in an HTML form.

Note: This table includes values for both the Direct Response and Relay Response methods so you can determine which fields to use or omit, since customizations at your campus may make use of the Direct Response fields. For this reason, the Direct Response information is shown here and can help you determine the use of each field.

v3.0 Gateway Interface Form Field Names	Possible Values (Bold = Default if not present or stored)	WebLink	ADC Direct Response	ADC Relay Response
x_ADC_Delim_Character	any character **(,)**		optional	
x_ADC_Delim_Data	TRUE		required	
x_ADC_Encapsulate_ Character	any character **(none**)		optional	
x_ADC_Relay_ Response	TRUE			required
x_ADC_URL	any valid URL (or FALSE for ADC Direct Response)		required (it is required for x_ADC_URL to have a value of "FALSE" for ADC Direct Response)	required
x_Address	any string	optional	optional	optional
x_Amount	any valid amount	required	required	required
x_Auth_Code	any valid Authorization Code	optional	optional	optional
x_Background_URL	any valid URL	optional	optional	optional
x_Bank_ABA_Code	any valid bank ABA code	optional	optional	optional
x_Bank_Acct_Num	any valid bank account number	optional	optional	optional
x_Bank_Acct_Type	**CHECKING**, SAVINGS	optional	optional	optional
x_Bank_Name	any valid bank name	optional	optional	optional
x_Card_Num	any valid credit card number	required if not using x_Show_For m	required	required if not using x_Show_Form
x_City	any string	optional	optional	optional
x_Color_Background	any valid color or color hex code (**White** or **#FFFFFF**)	optional	optional	optional
x_Color_Link	any valid color or color hex	optional	optional	optional

v3.0 Gateway Interface Form Field Names	Possible Values (Bold = Default if not present or stored)	WebLink	ADC Direct Response	ADC Relay Response
	code (**Blue** or **#0000FF**)			
x_Color_Text	any valid color or color hex code (**Black** or **#000000**)	optional	optional	optional
x_Company	any string	optional	optional	optional
x_Country	any string	optional	optional	optional
x_Cust_ID	any string	optional	optional	optional
x_Description	any string	optional	optional	optional
x_Duty	any valid amount	optional	optional	optional
x_Email	any valid email address	optional	optional	optional
x_Email_Customer	**TRUE**, FALSE	optional	optional	optional
x_Email_Merchant	**TRUE**, FALSE	optional	optional	optional
x_Exp_Date	mmyy, mm/yy, mm/yyyy	required if not using x_Show_Form	required	required if not using x_Show_Form
x_Fax	any string	optional	optional	optional
x_First_Name	any string	optional	optional	optional
x_Footer_Email_Receipt	any valid text	optional	optional	optional
x_Footer_Html_Payment_Form	any valid text or HTML	optional	optional	optional
x_Footer_Html_Receipt	any valid text or HTML	optional	optional	optional
x_Freight	any valid amount	optional	optional	optional
x_Header_Email_Receipt	any valid text	optional	optional	optional
x_Header_Html_Payment_Form	any valid text or HTML	optional	optional	optional
x_Header_Html_Receipt	any valid text or HTML	optional	optional	optional
x_Invoice_Num	any string	optional	optional	optional
x_Last_Name	any string	optional	optional	optional
x_Login	any valid merchant login ID	required	required	required
x_Logo_URL	any valid URL	optional	optional	optional
x_Merchant_Email	any valid email address	optional	optional	optional
x_Method	**CC**, ECHECK	optional	optional	optional
x_Password	valid password for the Login ID specified in x_Login		optional - only required for x_Type values of CREDIT, VOID, CAPTURE_ONLY, and PRIOR_AUTH_CAPTURE	optional - only required for x_Type values of CREDIT, VOID, CAPTURE_ONLY, and PRIOR_AUTH_CAPTURE
x_Phone	any string	optional	optional	optional
x_PO_Num	any string	optional	optional	optional
x_Receipt_Link_Method	**LINK**, POST, GET	optional	optional	optional
x_Receipt_Link_Text	any string (**"Continue"**)	optional	optional	optional
x_Receipt_Link_URL	any valid URL (must also exist in Merchant Configuration)	optional	optional	optional
x_Rename	[OldVariableName], [NewVariableName]	optional	optional	optional
x_Ship_To_Address	any string	optional	optional	optional

v3.0 Gateway Interface Form Field Names	Possible Values (Bold = Default if not present or stored)	WebLink	ADC Direct Response	ADC Relay Response
x_Ship_To_City	any string	optional	optional	optional
x_Ship_To_Company	any string	optional	optional	optional
x_Ship_To_Country	any string	optional	optional	optional
x_Ship_To_First_Name	any string	optional	optional	optional
x_Ship_To_Last_Name	any string	optional	optional	optional
x_Ship_To_State	any string	optional	optional	optional
x_Ship_To_Zip	any string	optional	optional	optional
x_Show_Form	PAYMENT_FORM (do not use this field if you do not want to show the system's payment form)	optional		optional
x_State	any string	optional	optional	optional
x_Tax	any valid amount	optional	optional	optional
x_Tax_Exempt	TRUE, **FALSE**	optional	optional	optional
x_Test_Request	TRUE, **FALSE**	optional	optional	optional
x_Trans_ID	any valid Transaction ID	optional	optional	optional
x_Type	**AUTH_CAPTURE**, AUTH_ONLY, CAPTURE_ONLY, CREDIT, VOID, PRIOR_AUTH_ CAPTURE	optional	optional	optional
x_Use_Fraudscreen	(Reserved for future use)	Not yet supported		
x_Version	2.5, **3.0**	Optional, but **strongly** recommended to be set to 3.0 to inform the system that you are passing 3.0 code.	Optional, but **strongly** recommended to be set to 3.0 to inform the system that you are passing 3.0 code.	Optional, but **strongly** recommended to be set to 3.0 to inform the system that you are passing 3.0 code.
x_Zip	any string	optional	optional	optional

Result Fields

Overview

The following table provides a reference to all of the fields of information that are included in the batch download file or are returned by one of the Automated Direct Connect (ADC) methods.

The ADC methods provide a way by which a merchant's server can integrate directly with the system to send transactions and receive responses.

The ADC Relay Response method, the method used in the first version of Jenzabar CX's interface with Authorize.Net, will return an HTML form POST containing the fields (see table below) to a script written by the merchant to complete the purchasing process and/or generate an HTML page to be displayed to the customer. The URL of this script is designated using the x_ADC_URL field and must be set as one of the Valid ADC or Receipt Link URLs in the URL Manager section of the Settings menu. This post will occur after the transaction is processed, and the response given by the merchant's script will be displayed to the customer as the result of the transaction. It is possible that none of the system's pages will ever be displayed using this method.

Table of Fields

Field Name	Description
x_response_code	Indicates the result of the transaction: 1 = Approved 2 = Declined 3 = Error
x_response_subcode	A code used by the system for internal transaction tracking.
x_response_reason_code	A code representing more details about the result of the transaction.
x_response_reason_text	Brief description of result, which corresponds with the Response Reason Code.
x_auth_code	6-digit approval code.
x_avs_code	Indicates the result of Address Verification System (AVS) checks: A = Address (Street) matches, ZIP does not B = Address Information Not Provided for AVS Check E = AVS error G = Non U.S. Card Issuing Bank N = No Match on Address (Street) or ZIP P = AVS not applicable for this transaction R = Retry – System unavailable or timed out S = Service not supported by issuer U = Address information is unavailable W = 9 digit ZIP matches, Address (Street) does not X = Address (Street) and 9 digit ZIP match Y = Address (Street) and 5 digit ZIP match Z = 5 digit ZIP matches, Address (Street) does not
x_trans_id	This number identifies the transaction in the system and can be used to submit a modification of this transaction at a later time via HTML form POST (such as voiding the transaction, or capturing an Auth Only transaction).
x_invoice_num	Echoed from form input values

Field Name	Description
x_description	Echoed from form input values
x_amount	Echoed from form input values
x_method	Echoed from form input values
x_type	Echoed from form input values
x_cust_id	Echoed from form input values
x_first_name	Echoed from form input values
x_last_name	Echoed from form input values
x_company	Echoed from form input values
x_address	Echoed from form input values
x_city	Echoed from form input values
x_state	Echoed from form input values
x_zip	Echoed from form input values
x_country	Echoed from form input values
x_phone	Echoed from form input values
x_fax	Echoed from form input values
x_email	Echoed from form input values
x_ship_to_first_name	Echoed from form input values
x_ship_to_last_name	Echoed from form input values
x_ship_to_company	Echoed from form input values
x_ship_to_address	Echoed from form input values
x_ship_to_city	Echoed from form input values
x_ship_to_state	Echoed from form input values
x_ship_to_zip	Echoed from form input values
x_ship_to_country	Echoed from form input values
x_tax	Echoed from form input values
x_duty	Echoed from form input values
x_freight	Echoed from form input values
x_tax_exempt	Echoed from form input values
x_po_num	Echoed from form input values
x_md5_hash	Generated by the system and to be validated by merchant for added security
Any merchant defined fields in the order the system received them	Echoed from form input values

APPENDIX C – RESPONSE CODES

Introduction

Source of Information

The information in this Appendix was obtained from Authorize.Net. It has been included here for quick reference to the Response Codes associated with the WebLink system.

Use of Response Codes

The transaction-processing gateway will attempt to always provide information about the status of a transaction. In the case of ADC or Virtual Terminal transactions, the system will report the status of the transaction in the Web browser using one of the text strings in the table of Response Reason Codes below. In the case of an ADC transaction, the response that is returned to the merchant's server will include more information in the form of a Response Code, a Response Subcode, a Response Reason Code, and Response Reason Text.

Response Fields

Description

Responses include the following components:

Response Code
Indicates the general state of the transaction. The general states indicated by the Response Code are approval, decline, or error.

Response Subcode
Internal tracking code for use by the transaction-processing gateway.

Response Reason Code
Code that can give a merchant more information about the transaction, such as what particular error occurred.

Response Reason Text
Text string that will briefly explain the type of response encountered. This text string can be echoed back to the customer to provide them with more information about their transaction. The Response Reason Text can change at any time in cases where additional clarification might be necessary, so it is *strongly* suggested that merchants do not parse this string expecting certain text to be there. Instead, a merchant should test for the Response Reason Code if they need to programmatically know these results, since the Response Reason Code will always represent these meanings, even if the text of those meanings changes.

Response Codes

Response Code	Meaning	Notes
1	This transaction has been approved.	
2	This transaction has been declined.	
3	There has been an error processing this transaction.	

Response Reason Codes and Response Reason Text

Response Reason Code	Response Reason Text	Notes
1	This transaction has been approved.	
2	This transaction has been declined.	General decline
3	This transaction has been declined.	Voice referral, equivalent to “Call Center” response
4	This transaction has been declined.	Pick up card (if possible)
5	Invalid Amount	
6	Invalid Credit Card Number	
7	Invalid Credit Card Expiration Date	
8	Credit Card Is Expired	
9	Invalid ABA Code	Invalid bank routing number
10	Invalid Account Number	Invalid bank account number
11	Duplicate Transaction	Try again in 2 minutes if this was not caused by a double-click
12	Authorization Code is Required but is not present	
13	Invalid Merchant Login	
14	Invalid Referrer URL	Occurs when a merchant has configured a list of Valid Referrer URLs in the settings in the Merchant Menu, but the referrer URL for this transaction does not match any entries on the list
15	Invalid Transaction ID	Transaction ID is not an integer or was not sent with a transaction that requires the Transaction ID (PRIOR_AUTH_CAPTURE or VOID)
16	Transaction Not Found	Used when a transaction is referenced by a correctly formatted transaction ID, but the transaction ID doesn’t appear in the system
17	The Merchant does not accept this type of Credit Card	
18	ACH Transactions are not accepted by this Merchant	
19	An error occurred during processing. Please try again in 5 minutes.	
20	An error occurred during processing. Please try again in 5 minutes.	
21	An error occurred during processing. Please try again in 5 minutes.	
22	An error occurred during processing. Please try again in 5 minutes.	
23	An error occurred during	General processor error

Response Reason Code	Response Reason Text	Notes
	processing. Please try again in 5 minutes.	
24	Nova Bank Number or Terminal ID is incorrect. Call Merchant Service Provider.	
25	An error occurred during processing. Please try again in 5 minutes.	
26	An error occurred during processing. Please try again in 5 minutes.	The processor could not be contacted
27	Address provided does not match billing address of cardholder.	Used if the transaction is configured to reject AVS mismatches
28	The Merchant does not accept this type of Credit Card.	
29	Paymentech identification numbers are incorrect. Call Merchant Service Provider.	Invalid Paymentech Client #, Merchant #, or Terminal #
30	Invalid configuration with Processor. Call Merchant Service Provider.	
31	FDC Merchant ID or Terminal ID is incorrect. Call Merchant Service Provider.	
32	Merchant Password Is Invalid Or Not Present.	
33	*Field* cannot be left blank.	The word *field* will be replaced with the actual name of the field that is causing the error by being left blank. This result occurs when a merchant has configured a field to be required in the settings in the Merchant Menu, but has not sent it with the transaction.
34	VITAL identification numbers are incorrect. Call Merchant Service Provider.	Invalid VITAL account
35	An error occurred during processing. Call Merchant Service Provider.	General processor error

APPENDIX D – FREQUENTLY ASKED QUESTIONS (FAQs)

How does the user's browser get loaded with the public key for the digital certificate?

Browsers come with lots of well-known built-in public keys. Navigate to IE Tools | Internet Options | Content | Certificates, and then browse the various tabs to see which ones come with IE. In Netscape, open the Security Info window by going to Communicator | Tools | Security Info and look in the sections that are listed beneath the *Certificates* heading.

What happens if the user deletes the digital certificates on their browser and then goes to a secured site?

If this scenario happens, the user's browser will present the user with a warning message saying that the CA who issues the certificate being used by the server cannot be verified as authentic and that you can proceed but there is no guarantee of the identity of the certificate's issuer or of the Web site owner.

What are the specifications for applying for a digital certificate for your particular requirement?

In order to apply for a digital certificate, you must first select the CA you want to use and then follow their guidelines about proving that you are who you say you are. There are various ways of doing that. The most prevalent way is to send required documents (i.e., a business license or some other official document that has been notarized).

Anyone can access an https:// site, so where is the security? Does the certificate come into play only when data is transmitted to and from the Web server?

When someone visits a site by going to https://, the server and the client browser establish a secure link through which data can be passed in an encrypted state so as not to be intercepted by a third party. Even if someone else is able to sniff the line between the browser and the server, they will not be able to decrypt the data being transferred.

How and where do we set up the public and private keys on our Web server once we get the digital certificate for a CA?

It varies by the Web server. In the Internet Information Server (IIS), there is a Key Manager that keeps track of all the certificates installed on the server. Each of the Web sites within IIS can then use the certificates stored in the Key Manager's database to establish SSL transactions.

Note: For more information on FAQs, refer to the Authorize.Net site.

APPENDIX E – UNDERSTANDING CREDIT CARD RATES AND FEES

Introduction

Overview

All banks and merchant providers require transaction fees from you for accepting credit cards. Typically, these fees are broken down into three categories: a discount rate, a transaction fee, and monthly fees. For the bank's purposes, a transaction is usually defined as any communication between you and the processing network. A credit transaction is treated the same as a regular transaction. Settling a batch is usually considered a transaction as well, as it involves communication with the processing network.

All the fees and charges must be disclosed to you prior to your commitment to the Merchant Agreement between you and your bank or merchant provider. Typically, the Merchant Agreement itself enumerates the applicable fees.

Discount Rates

A discount rate is a percentage of the total transaction amount that the bank will usually deduct prior to transferring your deposit into your bank account. Typical discount rates range from 2.5% to 5%, depending on the type of business and other factors. A higher rate may be charged on individual transactions if the transaction does not conform to certain qualifications as described by your bank or merchant provider. For instance, accepting a Visa Business Card credit card may cost you 1% more than regular transactions. The reasons for these non-qualified transaction surcharges and complete details on all transaction qualifications should be discussed with your bank or merchant provider.

Transaction Fees

Transaction fees are flat amounts that you pay for each transaction. Typical transaction fees range from 30 cents to 50 cents per transaction.

Monthly Fees

Monthly fees are charged for other account-related services, such as customer service, your monthly statement, and network access fees (gateway fee).

Example Transaction

Assume the following:

- Web site payment $100.00
- Discount rate 2.44%
- Transaction fee 0.40
- Monthly statement fee $7.50
- Monthly gateway $10.00

In this situation, the net deposit is $100.00 minus a .40 cent transaction fee minus a $2.44 transaction fee, or $97.16.

INDEX

V

W

X