

**WebLink
Implementation Guide
Card-Not-Present Transactions
Version 1.0**

Table Of Contents

INTRODUCTION	4
WEBLINK	5
What is WebLink?	5
How Does WebLink Work?	5
Submitting and Collecting Payment Data	5
Rendering a Response to the Customer	5
What is Required to Implement WebLink?	6
WebLink Implementation Requirements	6
Merchant Collects Payment Data	6
Merchant Requests the Gateway's Payment Form	6
Configuring the Gateway Response (Receipt or Relay)	7
Using the Merchant Interface to Configure WebLink.....	7
Security Considerations.....	8
STANDARD TRANSACTION SUBMISSION API FOR WEBLINK.....	10
Merchant Account Information	10
Payment Form Fields.....	11
Receipt Page Fields.....	11
Fields Common to both the Payment Form and Receipt Page	12
Relay Response Fields.....	12
Customer Name and Billing Address.....	13
Additional Customer Data.....	14
Email Settings.....	14
Invoice Information	15
Customer Shipping Address	15
Transaction Data	16
Level 2 Data.....	18
TRANSACTION SUBMISSION API FOR WEBLINK & WELLS FARGO SECURESOURCE MERCHANTS	19
Customer Name and Billing Address.....	19
Email Settings.....	20
Additional Customer Data.....	20
SAMPLE WEBLINK INTEGRATIONS – FORM CONSTRUCTION.....	22
Example 1 – Minimum requirements for requesting a Payment Form.....	22
Example 2 – Suggested minimum requirements	23
Example 3 – Using a form to gather information	23
Example 4 – Requesting a Receipt Page	24
Example 5 – Requesting a Payment Form and a Relay Response.....	24
GATEWAY RESPONSE API.....	26
Gateway Relay Response	26
Upgrading the Transaction Version.....	26
Response Code Details.....	28
APPENDIX A – TYPES OF CREDIT CARD TRANSACTIONS	38

Credit Card Transaction Types.....	38
APPENDIX B – FEATURES OF THE GATEWAY	41
Address Verification System.....	41
Credit Card Identification Code (CVV2/CVC2/CID)	42
APPENDIX C – CUSTOMIZING NOTIFICATION TO CUSTOMERS	43
APPENDIX D – SUBMITTING TEST TRANSACTIONS TO THE SYSTEM	44
Test Mode.....	44
Running a Test Transaction	44
Test credit card numbers	45
APPENDIX E – CERTIFICATION	46
APPENDIX F – CURRENCY CODES	47

Introduction

Payment gateways facilitate electronic commerce by enabling merchants to accept credit cards and electronic checks as methods of payment for goods and services sold online. The gateway acts as a bridge between the merchant's Website and the financial institutions that process payment transactions. Payment data is collected online from the shopper and submitted to the gateway for real-time authorization.

Authorization is the process of checking the validity and available balance of a customer's credit card before the transaction can be accepted. To authorize a given credit card transaction, the gateway transmits the transaction information to the appropriate financial institutions for validation, then returns the response (approved or declined) from the institution to the merchant or customer. The payment gateway supports real-time and offline requests for credit card authorization.

Note: The payment gateway is targeted towards merchants that process Card-Not-Present transactions. In a Card-Not-Present transaction, the merchant and the shopper are not in the same physical location and the customer usually calls in the payment data or keys in the details of the credit card on a Website. All e-commerce and mail/telephone orders are Card-Not-Present transactions.

The gateway also supports electronic check transactions. Merchants can collect customer bank account numbers and routing numbers to pay for purchases.

This document describes how transactions can be submitted to the gateway for real-time processing using the WebLink method of integration.

WebLink

What is WebLink?

WebLink is the simplest and quickest method of integrating with the payment gateway, and requires minimal programming expertise to implement. Using this method, the customer's browser interacts directly with the gateway to submit transaction data. Merchants have the option to let the gateway handle all the steps in transaction processing—payment data collection, data submission, and response to the customer.

Important Note: Please be aware that the WebLink method of submitting transactions to the gateway is a temporary integration solution. As security standards for online payment transactions have increased, WebLink has become insufficient to meet the newest security needs of merchants. As a result, this connection method will eventually be discontinued as an acceptable method of submitting transactions to the gateway. All merchants currently using the WebLink method of integration are encouraged to migrate to new integration methods supported by the gateway.

How Does WebLink Work?

When using WebLink, the customer enters payment information into a payment form on the merchant's Website. The customer's browser transmits payment information to the gateway using an HTTPS form POST. The gateway receives transaction information, and sends a response, or a Receipt Page, to the customer's browser.

The merchant can configure their WebLink connection method to allow the gateway to securely host the payment form in which the customer enters payment data, and also to handle the transaction response. The merchant can configure the gateway-hosted Payment Form and Receipt Page in the Merchant Interface.

Submitting and Collecting Payment Data

Using the Weblink method of integration, the merchant can choose to host the Web page in which the customer enters payment data, or allow the gateway to host the page. The gateway-hosted page that collects payment data is called the Payment Form, and can be configured by the merchant to look like their Website. Using the gateway-hosted Payment Form, merchants can collect payment data securely without having a secure Website.

Rendering a Response to the Customer

The response returned directly to the customer's browser by the gateway is a Receipt Page. The merchant can configure certain elements of the Receipt Page to look like their Website. However, the merchant can also request to receive the response from the gateway first, format it, and send it back to the gateway. The gateway then relays the merchant-formatted Receipt Page to the customer's browser. This response option is referred to as Relay Response. Merchants that use Relay Response can exercise a greater degree of control over the messaging of a response to the customer.

What is Required to Implement WebLink?

To use the WebLink connection method, merchants must be able to construct an HTML form that posts information to **https://secure.authorize.net/gateway/transact.dll**. The post string will consist of a set of NAME/ VALUE pairs. The NAME is the field name and indicates to the gateway what information is being submitted. The VALUE contains the content of the field. The response rendered by the gateway depends on the information in the post string. The set of fields that are recognized by the gateway are detailed in the Standard Transaction Submission API for WebLink section of this document.

WebLink Implementation Requirements

The table below details the different WebLink implementation requirements based on whether the Merchant or gateway hosts the Payment Form, and the response method selected by the Merchant.

PAYMENT DATA COLLECTION	RESPONSE TO CUSTOMER	MERCHANT REQUIREMENTS
Gateway hosts the Payment Form.	Gateway renders the Receipt Page.	Ability to write HTML and construct HTML forms.
Merchant hosts the pages to collect payment data and submits the data to the gateway for processing.	Gateway renders the Receipt Page.	Ability to host secure pages to collect payment data. Ability to write HTML and construct HTML forms.
Gateway hosts the Payment Form.	Merchant receives and formats the response from the gateway. The formatted response is posted back to the gateway, which renders the response to the customer.	Ability to construct HTML forms. Ability to write and host scripts that can post data to the gateway, receive a response from the gateway and return the response to the gateway.
Merchant hosts the pages to collect payment data and submits the data to the gateway for processing.	Merchant receives and formats the response from the gateway. The formatted response is posted back to the gateway, which renders the response to the customer.	Ability to host secure pages to collect payment data. Ability to construct HTML forms. Ability to write and host scripts that can post data to the gateway, receive a response from the gateway, and return the response to the gateway.

Merchant Collects Payment Data

If the merchant hosts the Payment Form then the information posted to the gateway needs to contain the minimum set of fields required to process the transaction. The set of required fields is described in the Standard Transaction Submission API for WebLink section of this document.

Merchant Requests the Gateway's Payment Form

If the merchant does not have the ability to host secure pages then the post string needs to contain only the information required to identify the merchant and request the gateway Payment Form. If the merchant has collected information from the customer on their Website before

requesting the gateway's Payment Form, the merchant can include this information in the post string in hidden fields. This results in the gateway rendering a Payment Form populated with the information passed in by the merchant.

Configuring the Gateway Response (Receipt or Relay)

The post string can contain information that indicates to the gateway how the Receipt Page should be formatted. The merchant can also provide a URL in the post string to which the gateway will post a response. The merchant site can receive, parse, and translate the response and post it back to the gateway. The gateway will relay the response to the customer.

Using the Merchant Interface to Configure WebLink

The merchant has the option to submit information in the post string regarding the format of the Receipt Page, the Payment Form, and the destination URL of the Relay Response with each WebLink transaction. It is recommended that you do not submit configuration data on a per-transaction basis. Alternatively, the merchant can configure these in the Merchant Interface. The Merchant Interface is a tool through which merchants can manage their accounts and their transaction activity. A Login ID and password are required to access this tool. The URL to the Merchant Interface is available to the merchant from their merchant service provider.

Payment Form Settings

The merchant can configure the following on the Payment Form:

- The background color of the form
- The color of the text on the form
- The header text
- The footer text

The merchant can also configure the set of Payment Form fields that should be displayed to the customer and the set of fields that the customer will be required to fill out.

To configure the Payment Form fields:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Payment Form* link in the Transaction Submission section

Receipt Page Settings

The merchant can configure the following on the Receipt Page:

- The background color of the form
- The color of the text on the form
- The header text
- The footer text

The merchant can also define the method by which the customer is returned to the merchant Website, and the destination URL to which the customer will be returned.

To configure the Receipt Page:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Receipt Page* link in the Transaction Response section

Note: When creating headers and footers for the Receipt Page, avoid using double quotes.

Relay Response URL Settings

The merchant can configure the default URL to which the gateway should send the response.

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on *Relay Response* in the Transaction Response section
4. Type in the default URL to which the gateway should send the response

The merchant can configure one or more valid relay response URLs. To configure valid relay response URLs in addition to the default specified:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Response /Receipt URLs* link in the Transaction Response section
4. Click on the *ADD URL* link
5. Add a valid URL
6. Click *Submit*

Security Considerations

Payment Form Requests

Do not submit your password with Payment Form requests to the gateway, as doing so may allow your password to be compromised. All Payment Form requests submitted to the gateway with a password will be rejected.

To avoid rejected Payment Form requests, make sure that Password-Required mode is OFF at all times. To check the status of this setting:

1. Log into the Merchant Interface
2. Click *Settings* in the Main Menu
3. Click *Password-Required Mode* in the Transaction Submission section
4. Uncheck the box labeled "Require Password for ALL Transactions." (If this box is unchecked already, DO NOT check it.)

Referrer URL

Please use the Referrer URL security feature. This feature allows the merchant to configure the set of URLs (Website addresses) that are authorized to submit transactions to the payment gateway. When the gateway receives the transaction, it will check to make sure that the originating URL in the HTTP header matches the list configured by the merchant. The transaction will be accepted only if the URL matches the one configured in the system

To set up a Referrer URL:

1. Log into the Merchant Interface
2. Click *Settings* in the Main Menu
3. Click on *WebLink* in the Transaction Submission section
4. Click on *Referrer URLs*
5. Click on *Add URL*
6. Enter the URL
7. Click *Submit*

Note: Use of port 443 for Relay Response information transfers for reasons of security is recommended.

Standard Transaction Submission API for WebLink

The Standard Transaction Submission API defines how transactions should be submitted to the gateway for real-time transaction processing. The API consists of a set of fields that are required for each transaction, and a set of fields that are optional. Under the API, the gateway accepts a NAME/ VALUE pair. The NAME is the field name and indicates to the gateway what information is being submitted. The VALUE contains the content of the field. In transmitting information to the gateway the merchant site should post transactions to <https://secure.authorize.net/gateway/transact.dll>.

The following tables contain the data fields that may be submitted to the system with any transaction. The fields are grouped logically in the tables, based on the information submitted. Each table contains the following information:

- *Field* – Name of the parameter that may be submitted on a transaction.
- *Required* – Indicates whether the field is required on a transaction. If *Conditional*, indicates that the field is required based on the existence or value of another field. In cases where a dependency exists, an explanation is provided.
- *Value* – Lists the possible values that may be submitted for the field. In cases where a format is validated, an explanation is provided.
- *Max Length* – Indicates the maximum number of characters that may be supplied for each field.
- *Description* – Provides additional details on how the field is used.

Merchant Account Information

The following fields in the API allow the system to identify the merchant submitting the transaction and the state of the merchant's account on the gateway.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_login	Required	Varies by merchant	20	Pass the Login ID used to access the Merchant Interface.
x_version	Optional If no value is specified, the value located in the Transaction Version settings within the Merchant Interface will be used.	2.5, 3.0, 3.1	3	Indicates to the system the set of fields that will be included in the response: <ul style="list-style-type: none"> • 3.0 is the standard version • 3.1 allows the merchant to utilize the Card Code feature
x_test_request	Optional	TRUE, FALSE	5	Indicates whether the transaction should be processed as a test transaction. Please refer to Appendix E for further information on this field.

Payment Form Fields

These fields are optional and need to be used only if the merchant is using the gateway's Payment Form to collect payment data. These fields, except `x_show_form`, can also be configured in the Merchant Interface and do not have to be submitted with each transaction.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
<code>x_show_form</code>	Conditional Required if the merchant is using the gateway's Payment Form	PAYMENT FORM		This field should be passed in with the value of PAYMENT FORM only if the merchant wishes to use the gateway's Payment Form to collect payment data.
<code>x_header_html_payment_form</code>	Optional	Valid Text or HTML		The text submitted in this field will be displayed as the header on the Payment Form.
<code>x_footer_html_payment_form</code>	Optional	Valid Text or HTML		The text submitted in this field will be displayed as the footer on the Payment Form.

Note: Payment Form requests that are submitted to the Payment Gateway with a password will be rejected.

Receipt Page Fields

All fields in this section are optional. The merchant can configure the information captured in these fields in the Merchant Interface instead of submitting it with every transaction.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
<code>x_header_html_receipt</code>	Optional	Valid Text or HTML code.		The text contained in this field will be displayed at the top of the Receipt Page.
<code>x_footer_html_receipt</code>	Optional	Valid Text or HTML code		The text contained in this field will be displayed at the bottom of the Receipt Page.
<code>x_receipt_link_method</code>	Optional	LINK POST GET		This field specifies what kind of link is made back to a merchant's Website.
<code>x_receipt_link_text</code>	Optional	Any Valid text		If the <code>x_receipt_link_method</code> is LINK, the value in this field will become a hyperlinked text on the Receipt Page If the <code>x_receipt_link_method</code> is GET or POST the value in this field becomes the text of a submit button. An HTML form is created in the Receipt Page that has hidden fields

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
				containing the results of the transaction processed.
x_receipt_link_url	Optional	Any Valid URL		<p>If the x_receipt_link_method is LINK the URL specified in the fields becomes the target of the hyperlinked text. Clicking on it will take the user to the specified URL.</p> <p>If the x_receipt_link_method is GET or POST the URL will become the action of the HTML form.</p> <p>The URL should be configured in the Merchant Interface to be accepted as valid by the gateway.</p>

Fields Common to both the Payment Form and Receipt Page

All fields in this section are optional. The merchant can configure the information captured in these fields in the Merchant Interface instead of submitting it with every transaction.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_logo_url	Optional	Any valid URL		This field is ideal for displaying a merchant logo on a page. The target of this URL will be displayed on the header of the Payment Form and Receipt Page.
x_background_url	Optional	Any valid URL		This field will allow the merchant to customize the background image of the Payment Form and Receipt Page. The target of the specified URL will be displayed as the background.
x_color_background	Optional	Any valid HTML color name or color hex code		Value in this field will set the background color for the Payment Form and Receipt Page.
x_color_link	Optional	Any valid HTML color name or color hex code		This field allows the color of the HTML links for the Payment Form and Receipt Page to be set to the value submitted in this field.
x_color_text	Optional	Any valid HTML color name or color hex code		This field allows the color of the text on the Payment Form and the Receipt Page to be set to the value submitted in this field.

Relay Response Fields

The following fields indicate to the gateway that the merchant is requesting a relay response. They should not be passed in if the merchant is requesting a Receipt Page or a Payment Form.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_relay_response	Conditional – if the merchant requires a relay response	TRUE		This is required to be set to TRUE. This indicates to the gateway that the response will be relayed to the customer's browser.
x_relay_url	Optional	Any valid URL		Contains the URL to which the gateway will post the response. The gateway will validate the URL submitted in this field with the values configured through the Merchant Interface. The gateway will reject the transaction if the value submitted does not match any of the URLs configured through the Merchant Interface. If no value is passed in this field, the gateway will post the response to the default relay URL configured in the Merchant Interface.

Customer Name and Billing Address

The customer billing address fields listed below contain information about the customer billing address associated with each transaction.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_first_name	Optional	Any string	50	Contains the first name of the customer associated with the billing address for the transaction.
x_last_name	Optional	Any string	50	Contains the last name of the customer associated with the billing address for the transaction.
x_company	Optional	Any string	50	Contains the company name associated with the billing address for the transaction.
x_address	Optional	Any string	60	Contains the address of the customer associated with the billing address for the transaction.
x_city	Optional	Any string	40	Contains the city of the customer associated with the billing address for the transaction.
x_state	Optional If passed, the value will be verified.	Any valid two-digit state code or full state name	40	Contains the state of the customer associated with the billing address for the transaction.
x_zip	Optional	Any string	20	Contains the zip of the customer associated with the billing address for the transaction.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_country	Optional If passed, the value will be verified.	Any valid two-digit country code or full country name (spelled in English)	60	Contains the country of the customer associated with the billing address for the transaction.
x_phone	Optional	Any string Recommended format is (123)123-1234	25	Contains the phone number of the customer associated with the billing address for the transaction.
x_fax	Optional	Any string Recommended format is (123)123-1234	25	Contains the fax number of the customer associated with the billing address for the transaction.

Additional Customer Data

Merchants may provide additional customer information with a transaction, based on their respective requirements.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_cust_id	Optional	Any string	20	Unique identifier to represent the customer associated with the transaction.
x_customer_ip	Optional	Required format is 255.255.255.255 If this value is not passed, it will default to 255.255.255.255	15	IP address of the customer initiating the transaction.
x_customer_tax_id	Optional	9 digits/numbers only	9	Tax ID or SSN of the customer initiating the transaction.

Email Settings

The following fields describe how and when emails will be sent when transactions are processed by the system.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_email	Optional	Any valid email address	255	Email address to which the customer's copy of the confirmation email is sent. No email will be sent to the customer if the email address does not meet standard email format

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
				checks.
x_email_customer	Optional	TRUE, FALSE If no value is submitted, system will default to the value configured in the Merchant Interface.	5	Indicates whether a confirmation email should be sent to the customer.
x_merchant_email	Optional	Any valid email address	255	Email address to which the merchant's copy of the customer confirmation email should be sent. If a value is submitted, an email will be sent to this address as well as the address(es) configured in the Merchant Interface.

Invoice Information

Based on their respective requirements, merchants may submit invoice information with a transaction. Two invoice fields are provided in the gateway API.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_invoice_num	Optional	Any string	20	Merchant assigned invoice number.
x_description	Optional	Any string	255	Description of the transaction.

Customer Shipping Address

The following fields describe the customer shipping information that may be submitted with each transaction.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_ship_to_first_name	Optional	Any string	50	Contains the customer shipping first name.
x_ship_to_last_name	Optional	Any string	50	Contains the customer shipping last name.
x_ship_to_company	Optional	Any string	50	Contains the customer shipping company.
x_ship_to_address	Optional	Any string	60	Contains the customer shipping address.
x_ship_to_city	Optional	Any string	40	Contains the customer shipping city.
x_ship_to_state	Optional If passed, the value	Any valid two-digit state code or full state name	40	Contains the customer shipping state.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
	will be verified.			
x_ship_to_zip	Optional	Any string	20	Contains the customer shipping zip.
x_ship_to_country	Optional If passed, the value will be verified.	Any valid two-digit country code or full country name (spelled in English)	60	Contains the customer shipping country.

Transaction Data

The following fields contain transaction specific information such as amount, payment method, and transaction type.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_amount	Required	Any amount	15	Total value to be charged or credited inclusive of tax.
x_currency_code	Optional	Valid currency code	3	Currency of the transaction amount. If left blank, this value will default to the value specified in the Merchant Interface. See Appendix G for other values.
x_method	Required	CC, ECHECK	N/A	Indicates the method of payment for the transaction being sent to the system. If left blank, this value will default to CC.
x_type	Required	AUTH_CAPTURE, AUTH_ONLY, CAPTURE_ONLY, CREDIT, VOID, PRIOR_AUTH_CAPTURE	N/A	Indicates the type of transaction. If the value in the field does not match any of the values stated, the transaction will be rejected. If no value is submitted in this field, the gateway will process the transaction as an AUTH_CAPTURE.
x_recurring_billing	Optional	YES, NO	3	Indicates whether the transaction is a recurring billing transaction.
x_bank_aba_code	Conditional Required if x_method = ECHECK	Valid routing number	9	Routing number of a bank for eCheck.Net transactions.
x_bank_acct_num	Conditional	Valid account	20	Checking or savings

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
	Required if x_method = ECHECK	number		account number.
x_bank_acct_type	Conditional Required if x_method = ECHECK	CHECKING, SAVINGS	8	Describes the type of bank account; if no value is provided, default is set to CHECKING.
x_bank_name	Conditional Required if x_method = ECHECK	Valid bank name	50	Contains the name of the customer's financial institution.
x_bank_acct_name	Conditional Required if x_method = ECHECK	Name on the Customer's bank account		Is the customer's name as it appears on their bank account
x_echeck_type	Conditional Required if x_method = ECHECK	WEB		This indicates that the eCheck payment request originated from a Website. The system will default this value to WEB if no value is sent.
x_card_num	Conditional Required if x_method = CC	Numeric credit card number	22	Contains the credit card number.
x_exp_date	Conditional Required if x_method = CC	MMYY, MM/YY, MM-YY, MMYYYY, MM/YYYY, MM-YYYY, YYYY-MM-DD, YYYY/MM/DD	N/A	Contains the date on which the credit card expires.
x_card_code	Optional	Valid CVV2, CVC2 or CID value	4	Three- or four-digit number on the back of a credit card (on the front for American Express).
x_trans_id	Conditional Required if x_type = CREDIT, VOID, or PRIOR_AUTH_CA PTURE	Valid transaction ID	10	ID of a transaction previously authorized by the gateway.
x_auth_code	Conditional	Valid authorization code	6	Authorization code for a previous transaction not

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
	Required if x_type = CAPTURE_ONLY			authorized on the gateway that is being submitted for capture.

Level 2 Data

The system supports Level 2 transaction data by providing the following fields as part of the transaction submission API.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_po_num	Optional	Any string	25	Contains the purchase order number.
x_tax	Optional	Any valid amount	15	Contains the tax amount.
x_tax_exempt	Optional	TRUE, FALSE	5	Indicates whether the transaction is tax exempt.
x_freight	Optional	Any valid amount	10	Contains the freight amount charged.
x_duty	Optional	Any valid amount	10	Contains the amount charged for duty.

Transaction Submission API for WebLink & Wells Fargo SecureSource Merchants

For merchants who process transactions through the Wells Fargo SecureSource product, some additional rules apply to transaction processing. Fields that are optional in the standard gateway API are required for Wells Fargo SecureSource merchants. The following tables describe these required fields. Only those fields that are different from the standard API are called out in this section.

Customer Name and Billing Address

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_first_name	Required	Any string	50	Contains the first name of the customer associated with the billing address for the transaction.
x_last_name	Required	Any string	50	Contains the last name of the customer associated with the billing address for the transaction.
x_company	Required	Any string	50	Contains the company name associated with the billing address for the transaction.
x_address	Required	Any string	60	Contains the address of the customer associated with the billing address for the transaction.
x_city	Required	Any string	40	Contains the city of the customer associated with the billing address for the transaction.
x_state	Required	Any valid two-digit state code or full state name	40	Contains the state of the customer associated with the billing address for the transaction.
x_zip	Required	Any string	20	Contains the zip of the customer associated with the billing address for the transaction.
x_country	Required	Any valid two-digit country code or full country name (spelled in English)	60	Contains the country of the customer associated with the billing address for the transaction.
x_phone	Required	Any string	25	Contains the phone number of the customer

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
		Recommended format is (123)123-1234		associated with the billing address for the transaction.

Email Settings

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_email	Required	Any valid email address	255	Email address to which a confirmation email is sent.

Additional Customer Data

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_customer_ip	Required	Required format is 255.255.255.255. If this value is not passed, it will default to 255.255.255.255	15	IP address of the customer initiating the transaction.
x_customer_organization_type	Required	I, B I = Individual B = Business	N/A	Required for all eCheck transactions for Wells Fargo SecureSource Merchants.
x_customer_tax_id	Conditional IF x_type = ECHECK, merchant must provide EITHER x_customer_tax_id OR x_drivers_license_num AND x_drivers_license_state AND x_drivers_license_dob	9 digits or numbers only	9	Tax ID or SSN of the customer initiating the transaction. If the Tax ID or SSN is not available, the customer's driver's license number, driver's license state, and date of birth must be used in its place.
x_drivers_license_num	Conditional IF x_type = ECHECK, merchant must provide EITHER x_customer_tax_id OR x_drivers_license_num AND x_drivers_license_state AND x_drivers_license_dob		50	Required for all eCheck transactions for Wells Fargo SecureSource Merchants where the Tax ID or SSN is not provided.
x_drivers_license_state	Conditional	2-character state abbreviation	2	Required for all eCheck transactions for Wells

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
	IF x_type = ECHECK, merchant must provide EITHER x_customer_tax_id OR x_drivers_license_num AND x_drivers_license_state AND x_drivers_license_dob			Fargo SecureSource Merchants where the Tax ID or SSN is not provided.
x_drivers_license_dob	Conditional IF x_type = ECHECK, merchant must provide EITHER x_customer_tax_id OR x_drivers_license_num AND x_drivers_license_state AND x_drivers_license_dob	YYYY-MM-DDD, YYYY/MM/DD, MM/DD/YYYY, MM-DD-YYYY,	N/A	Required for all eCheck transactions for Wells Fargo SecureSource Merchants where the Tax ID or SSN is not provided.

Sample WebLink Integrations – Form Construction

The basic HTML tags used to construct a form to post a transaction to the gateway would be written as follows:

```
<FORM METHOD=POST  
ACTION="https://secure.authorize.net/gateway/transact.dll">  
</FORM>
```

Any form fields that need to be sent to the system would be enclosed as `<INPUT>` fields within the opening `<FORM>` tag and the closing `</FORM>` tag. For example, a form that contained a merchant's login ID would look like this:

```
<INPUT TYPE=HIDDEN NAME="x_login" VALUE="your login ID goes here">
```

Some things to note about the sample line above:

- `<INPUT>` is an HTML tag that does not need a closing tag (unlike `<FORM>` which needs a closing tag `</FORM>`).
- `TYPE` in the sample above is an attribute of the `INPUT` tag and has a value `HIDDEN`. In this case it means the information being passed to the gateway server would be hidden from the customer's HTML page (since the information that is being passed is the merchant's Login ID).
- The `NAME="x_login"` is the category, or field, of information that is being passed, in this case the Login ID of the merchant.
- The `VALUE="your Login ID goes here"` is where the merchant would put their merchant Login ID (inside the double quotes).
- The `NAME/VALUE` pair is one of the things that the gateway server looks for when a transaction is submitted. If part of the `NAME/VALUE` pair is not present or improperly formed, the transaction will be rejected.

Note: Never send in the password as a hidden field in the post string.

The gateway will accept fields that are not specified in the API so long as the name used to describe the field is not identical to a field name in the API. These fields are called "user" fields and can contain any information that might be useful in the transaction. User fields will be echoed back with the results of the transaction and will be displayed on the merchant's receipts; they will not be displayed on the customer's receipts.

The following are some examples of HTML code that can be customized and inserted into a merchant's Website to provide easy ways to integrate the Website with the system.

Example 1 – Minimum requirements for requesting a Payment Form

The following HTML code represents the minimum data that would need to be inserted into a page to provide a connection to the system: The result of this HTML code snippet is a button. Clicking on the button will take the user to the gateway's Payment Form. The transaction amount

will be pre-filled on the form. The user will enter the remaining data required for the transaction on the form.

```
<FORM METHOD=POST ACTION="https://secure.authorize.net/gateway/transact.dll">
<INPUT TYPE=HIDDEN NAME="x_login" VALUE="your login ID goes here">
<INPUT TYPE=HIDDEN NAME="x_show_form" VALUE="PAYMENT_FORM">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="amount goes here">
<INPUT TYPE=SUBMIT VALUE="Click here for secure payment form">
</FORM>
```

Example 2 – Suggested minimum requirements

The following HTML code allows for including additional information in the Payment Form or Receipt Page. The result of this HTML code snippet is a button. Clicking on the button will take the user to the gateway's Payment Form. The Payment Form will already display the description, the customer ID number, and the invoice number. This information will appear on the receipt and in the e-mail confirmation.

The merchant server will have to create the customer ID number and invoice number dynamically in order to include this information in the POST. The gateway will not perform this function.

```
<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 login ID goes here">
<INPUT TYPE=HIDDEN NAME="x_show_form" VALUE="PAYMENT_FORM">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="amount goes here">
<INPUT TYPE=HIDDEN NAME="x_cust_id" VALUE="a customer identification number
or string goes here">
<INPUT TYPE=HIDDEN NAME="x_description" VALUE="description of transaction">
<INPUT TYPE=HIDDEN NAME="x_invoice_num" VALUE="invoice number goes here">
<INPUT TYPE=SUBMIT VALUE="Click here for secure payment form">
</FORM>
```

Example 3 – Using a form to gather information

The following HTML code demonstrates the ability to send additional information to the system, including information that is specified by the customer. The result of this HTML code snippet is a page that displays a form allowing the customer to specify their name and any specific shipping instructions. The *x_first_name* and *x_last_name* fields are normal fields recognized by the system. The *shipping_instructions* field is not a field recognized by the system, and so it is treated as a user field. System-recognized fields can also be used for normal shipping information.

```
<FORM METHOD=POST ACTION="https://secure.authorize.net/gateway/transact.dll">
<INPUT TYPE=HIDDEN NAME="x_login" VALUE="your login ID goes here">
<INPUT TYPE=HIDDEN NAME="x_show_form" VALUE="PAYMENT_FORM">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="amount goes here">
<INPUT TYPE=HIDDEN NAME="x_cust_id" VALUE="a unique customer ID goes here">
<INPUT TYPE=HIDDEN NAME="x_description" VALUE="description of transaction">
<INPUT TYPE=HIDDEN NAME="x_invoice_num" VALUE="invoice number goes here">
Enter your first name:
```

```
<INPUT TYPE=TEXT NAME="x_first_name"><BR>
Enter your last name:
<INPUT TYPE=TEXT NAME="x_last_name"><BR>
Enter Any special shipping instructions:
<INPUT TYPE=TEXT NAME="shipping_instructions"><BR>
<INPUT TYPE=SUBMIT VALUE="Click here for secure payment form">
</FORM>
```

Example 4 – Requesting a Receipt Page

The following are the minimum requirements for a requesting the Receipt Page.

```
<FORM METHOD=POST ACTION="https://secure.authorize.net/gateway/transact.dll">
<INPUT TYPE=HIDDEN NAME="x_version" VALUE="3.1">
<INPUT TYPE=HIDDEN NAME="x_login" VALUE="your login ID goes here">
<INPUT TYPE=HIDDEN NAME="x_card_num" VALUE="customer's credit card number">
<INPUT TYPE=HIDDEN NAME="x_exp_date" VALUE="customer's credit card expiration
date goes here">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="amount goes here">
<INPUT TYPE=SUBMIT VALUE="Submit">
</FORM>
```

Example 5 – Requesting a Payment Form and a Relay Response

The following are the minimum requirements for a Relay Response transaction.

```
<FORM METHOD=POST ACTION="https://secure.authorize.net/gateway/transact.dll">
<INPUT TYPE=HIDDEN NAME="x_login" VALUE="your login ID goes here">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="amount goes here">
<INPUT TYPE=HIDDEN NAME="x_relay_response" VALUE="True">
<INPUT TYPE=HIDDEN NAME="x_relay_url" VALUE="Any valid URL">
<INPUT TYPE=SUBMIT VALUE="Click here for secure payment form">
</FORM>
```

Note: The line `<INPUT TYPE=HIDDEN NAME="x_show_form" VALUE="PAYMENT_FORM">` is required only if the merchant wishes to use the gateway's Payment Form. This line is not required if the merchant decides to construct their own HTML form to collect payment data. In this scenario, the user's card number and card expiration date will have to be passed to the gateway.

- Since the HTML code that is returned to the browser appears to come from the gateway server, any links to images need to be absolute URLs with the full path to the server on which the images reside. If relative URLs are used, the customer's browser will unsuccessfully try to load the images from the gateway server.
- The URL specified in `x_relay_url` should be a script that can interactively parse posted information. The URL can be a plain HTML page if a static response is desired for every transaction. In this scenario, the merchant's Web server will need to be configured to allow the POST method to plain HTML pages.

- Use of frames is not recommended. Even though the Payment Form used will be secure, the lock icon on the user's taskbar will key off the surrounding frame and not off the Payment Form. The page will not look secure to the customer.
- Redirects in the relay scripts are not recommended, as the information may not be transferred properly.
- When the merchant's response is relayed back to the customer, anything that is in the HTTP headers will be replaced. For example, if a developer is relying on custom information to be included in the headers, such as cookies, their implementation will most likely fail.

Gateway Response API

The standard gateway response to a transaction submitted via WebLink is either a Receipt Page that is displayed to the customer or a POST string to a site designated by the merchant. The merchant can then parse the POST string, translate it to a response, and submit it back to the gateway. The gateway will relay the response to the customer's browser.

Gateway Relay Response

The response from the gateway to a request via WebLink for a Relay Response consists of a set of fields returned as a POST string to the merchant server at the location indicated in the `x_relay_url` field. There are two versions of the response string. The set of fields in the response differ based on the response version. Version 3.0 is the Payment Gateway default.

The 3.1 response version includes the Card Code as an additional field in the response.

Merchants wishing to use the Card Code security feature must use transaction version 3.1.

Upgrading the Transaction Version

To upgrade the transaction version:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on *Transaction Version* in the Transaction Response section
4. Change the Transaction Version using the drop-down box
5. Click *Submit* to save changes

Note: You can only upgrade to a higher transaction version. You cannot set your transaction version to a previous version.

The following table indicates the fields returned in the response from the gateway to the merchant server in the case of a Relay Response. Any fields that were sent to the gateway in addition to the fields described in the Standard Transaction Submission API are returned in the response after the set of fields described in the table below.

FIELD NAME OF VALUE IN RESPONSE	INFORMATION	DESCRIPTION
<code>x_response_code</code>	Response Code	Indicates the result of the transaction: 1 = Approved 2 = Declined 3 = Error
<code>x_response_subcode</code>	Response Subcode	A code used by the system for internal transaction tracking.
<code>x_response_reason_code</code>	Response Reason Code	A code representing more details about the result of the transaction.
<code>x_response_reason_text</code>	Response Reason Text	Brief description of the result, which corresponds with the Response Reason Code.
<code>x_auth_code</code>	Approval Code	The six-digit alphanumeric authorization or approval code.
<code>x_avs_code</code>	AVS Result Code	Indicates the result of Address Verification System

FIELD NAME OF VALUE IN RESPONSE	INFORMATION	DESCRIPTION
		(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	Transaction ID	This number identifies the transaction in the system and can be used to submit a modification of this transaction at a later time, such as voiding, crediting or capturing the transaction.
x_invoice_num	Invoice Number	Echoed from form input value for x_invoice_num.
x_description	Description	Echoed from form input value for x_description.
x_amount	Amount	Echoed from form input value for x_amount.
x_method	Method	Echoed from form input value for x_method.
x_type	Transaction Type	Echoed from form input value for x_type.
x_cust_id	Customer ID	Echoed from form input value for x_cust_id.
x_first_name	Cardholder First Name	Echoed from form input value for x_first_name.
x_last_name	Cardholder Last Name	Echoed from form input value for x_last_name.
x_company	Company	Echoed from form input value for x_company.
x_address	Billing Address	Echoed from form input value for x_address.
x_city	City	Echoed from form input value for x_city.
x_state	State	Echoed from form input value for x_state.
x_zip	Zip	Echoed from form input value for x_zip.
x_country	Country	Echoed from form input value for x_country.
x_phone	Phone	Echoed from form input value for x_phone.
x_fax	Fax	Echoed from form input value for x_fax.
x_email	Email	Echoed from form input value for x_email.
x_ship_to_first_name	Ship to First Name	Echoed from form input value for x_ship_to_first_name.
x_ship_to_last_name	Ship to Last Name	Echoed from form input value for x_ship_to_last_name.
x_ship_to_company	Ship to Company	Echoed from form input value for x_ship_to_company.
x_ship_to_address	Ship to Address	Echoed from form input value for x_ship_to_address.
x_ship_to_city	Ship to City	Echoed from form input value for x_ship_to_city.

FIELD NAME OF VALUE IN RESPONSE	INFORMATION	DESCRIPTION
x_ship_to_state	Ship to State	Echoed from form input value for x_ship_to_state.
x_ship_to_zip	Ship to Zip	Echoed from form input value for x_ship_to_zip.
x_ship_to_country	Ship to Country	Echoed from form input value for x_ship_to_country.
x_tax	Tax Amount	Echoed from form input value for x_tax.
x_duty	Duty Amount	Echoed from form input value for x_duty.
x_freight	Freight Amount	Echoed from form input value for x_freight.
x_tax_exempt	Tax Exempt Flag	Echoed from form input value for x_tax_exempt.
x_po_num	PO Number	Echoed from form input value for x_po_num.
x_md5_hash	MD5 Hash	System-generated hash that may be validated by the merchant to authenticate a transaction response received from the gateway.
x_cvv2_resp_code	Card Code (CVV2/CVC2/CID) Response Code	Indicates the results of Card Code verification: M = Match N = No Match P = Not Processed S = Should have been present U = Issuer unable to process request
Position 40 - 68		Reserved for future use.
Position 69 -		Echo of merchant-defined fields.

Response Code Details

The fields in the response that describe the status of the transaction are Response Code, Response Reason Code, and Response Reason Text. The following tables define the values that the gateway may return in these fields.

Description of Response Fields

The three status fields in the transaction response are defined as follows:

- The *Response Code* indicates the overall status of the transaction with possible values of approval, decline, or error.
- The *Response Reason Code* gives merchants more information about the transaction status.
- The *Response Reason Text* is a text string that will give more detail on why the transaction resulted in a specific response code. This field is a text string that can be echoed back to the customer to provide them with more information about their transaction. It is strongly suggested that merchants do not parse this string expecting certain text. Instead, a merchant should test for the Response Reason Code if they need to programmatically know these results; the Response Reason Code will always represent these meanings, even if the text descriptions change.

Response Codes

RESPONSE CODE	DESCRIPTION
1	This transaction has been approved.
2	This transaction has been declined.
3	There has been an error processing this transaction.
4	This transaction is being held for review.

Response Reason Codes & Response Reason Text

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
1	1	This transaction has been approved.	
2	2	This transaction has been declined.	
2	3	This transaction has been declined.	
2	4	This transaction has been declined.	The code returned from the processor indicating that the card used needs to be picked up.
3	5	A valid amount is required.	The value submitted in the amount field did not pass validation for a number.
3	6	The credit card number is invalid.	
3	7	The credit card expiration date is invalid.	The format of the date submitted was incorrect.
3	8	The credit card has expired.	
3	9	The ABA code is invalid.	The value submitted in the x_bank_aba_code field did not pass validation or was not for a valid financial institution.
3	10	The account number is invalid.	The value submitted in the x_bank_acct_num field did not pass validation.
3	11	A duplicate transaction has been submitted.	A transaction with identical amount and credit card information was submitted two minutes prior.
3	12	An authorization code is required but not present.	A transaction that required x_auth_code to be present was submitted without a value.
3	13	The merchant Login ID is invalid or the account is inactive.	
3	14	The Referrer or Relay Response URL is invalid.	The Relay Response or Referrer URL does not match the merchant's configured value(s) or is absent. Applicable only to SIM and WebLink APIs.
3	15	The transaction ID is invalid.	The transaction ID value is non-numeric or was not present for a transaction that requires it (i.e., VOID, PRIOR_AUTH_CAPTURE, and CREDIT).
3	16	The transaction was not found.	The transaction ID sent in was properly formatted but the gateway had no record of the

			transaction.
3	17	The merchant does not accept this type of credit card.	The merchant was not configured to accept the credit card submitted in the transaction.
3	18	ACH transactions are not accepted by this merchant.	The merchant does not accept electronic checks.
3	19	An error occurred during processing. Please try again in 5 minutes.	
3	20	An error occurred during processing. Please try again in 5 minutes.	
3	21	An error occurred during processing. Please try again in 5 minutes.	
3	22	An error occurred during processing. Please try again in 5 minutes.	
3	23	An error occurred during processing. Please try again in 5 minutes.	
3	24	The Nova Bank Number or Terminal ID is incorrect. Call Merchant Service Provider.	
3	25	An error occurred during processing. Please try again in 5 minutes.	
3	26	An error occurred during processing. Please try again in 5 minutes.	
2	27	The transaction resulted in an AVS mismatch. The address provided does not match billing address of cardholder.	
3	28	The merchant does not accept this type of credit card.	The Merchant ID at the processor was not configured to accept this card type.
3	29	The PaymentTech identification numbers are incorrect. Call Merchant Service Provider.	
3	30	The configuration with the processor is invalid. Call Merchant Service Provider.	
3	31	The FDC Merchant ID or Terminal ID is incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	32	The merchant password is invalid or not present.	
3	33	<i>FIELD</i> cannot be left blank.	The word <i>FIELD</i> will be replaced by an actual field name. This error indicates that a field the merchant specified as required was not filled in.
3	34	The VITAL identification	The merchant was incorrectly set up at the

		numbers are incorrect. Call Merchant Service Provider.	processor.
3	35	An error occurred during processing. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	36	The authorization was approved, but settlement failed.	
3	37	The credit card number is invalid.	
3	38	The Global Payment System identification numbers are incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	39	The supplied currency code is either invalid, not supported, not allowed for this merchant or doesn't have an exchange rate.	
3	40	This transaction must be encrypted.	
2	41	This transaction has been declined.	Only merchants set up for the FraudScreen.Net service would receive this decline. This code will be returned if a given transaction's fraud score is higher than the threshold set by the merchant.
3	42	There is missing or invalid information in a required field.	This is applicable only to merchants processing through the Wells Fargo SecureSource product who have requirements for transaction submission that are different from merchants not processing through Wells Fargo.
3	43	The merchant was incorrectly set up at the processor. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
2	44	This transaction has been declined.	The merchant would receive this error if the Card Code filter has been set in the Merchant Interface and the transaction received an error code from the processor that matched the rejection criteria set by the merchant.
2	45	This transaction has been declined.	This error would be returned if the transaction received a code from the processor that matched the rejection criteria set by the merchant for both the AVS and Card Code filters.
3	46	Your session has expired or does not exist. You must log in to continue working.	
3	47	The amount requested for settlement may not be greater than the original amount authorized.	This occurs if the merchant tries to capture funds greater than the amount of the original authorization-only transaction.
3	48	This processor does not accept partial reversals.	The merchant attempted to settle for less than the originally authorized amount.
3	49	A transaction amount greater than \$99,999 will not be accepted.	

3	50	This transaction is awaiting settlement and cannot be refunded.	Credits or refunds may only be performed against settled transactions. The transaction against which the credit/refund was submitted has not been settled, so a credit cannot be issued.
3	51	The sum of all credits against this transaction is greater than the original transaction amount.	
3	52	The transaction was authorized, but the client could not be notified; the transaction will not be settled.	
3	53	The transaction type was invalid for ACH transactions.	If x_method = ECHECK, x_type cannot be set to CAPTURE_ONLY.
3	54	The referenced transaction does not meet the criteria for issuing a credit.	
3	55	The sum of credits against the referenced transaction would exceed the original debit amount.	The transaction is rejected if the sum of this credit and prior credits exceeds the original debit amount
3	56	This merchant accepts ACH transactions only; no credit card transactions are accepted.	The merchant processes eCheck transactions only and does not accept credit cards.
3	57	An error occurred in processing. Please try again in 5 minutes.	
3	58	An error occurred in processing. Please try again in 5 minutes.	
3	59	An error occurred in processing. Please try again in 5 minutes.	
3	60	An error occurred in processing. Please try again in 5 minutes.	
3	61	An error occurred in processing. Please try again in 5 minutes.	
3	62	An error occurred in processing. Please try again in 5 minutes.	
3	63	An error occurred in processing. Please try again in 5 minutes.	
3	64	The referenced transaction was not approved.	This error is applicable to Wells Fargo SecureSource merchants only. Credits or refunds cannot be issued against transactions that were not authorized.
2	65	This transaction has been declined.	The transaction was declined because the merchant configured their account through the Merchant Interface to reject transactions with certain values for a Card Code mismatch.
3	66	This transaction cannot be accepted for processing.	The transaction did not meet gateway security guidelines.
3	67	The given transaction type is not supported for this merchant.	This error code is applicable to merchants using the Wells Fargo SecureSource product only. This product does not allow transactions of type CAPTURE_ONLY.

3	68	The version parameter is invalid.	The value submitted in x_version was invalid.
3	69	The transaction type is invalid.	The value submitted in x_type was invalid.
3	70	The transaction method is invalid.	The value submitted in x_method was invalid.
3	71	The bank account type is invalid.	The value submitted in x_bank_acct_type was invalid.
3	72	The authorization code is invalid.	The value submitted in x_auth_code was more than six characters in length.
3	73	The driver's license date of birth is invalid.	The format of the value submitted in x_drivers_license_num was invalid.
3	74	The duty amount is invalid.	The value submitted in x_duty failed format validation.
3	75	The freight amount is invalid.	The value submitted in x_freight failed format validation.
3	76	The tax amount is invalid.	The value submitted in x_tax failed format validation.
3	77	The SSN or tax ID is invalid.	The value submitted in x_customer_tax_id failed validation.
3	78	The Card Code (CVV2/CVC2/CID) is invalid.	The value submitted in x_card_code failed format validation.
3	79	The driver's license number is invalid.	The value submitted in x_drivers_license_num failed format validation.
3	80	The driver's license state is invalid.	The value submitted in x_drivers_license_state failed format validation.
3	81	The requested form type is invalid.	The merchant requested an integration method not compatible with the Advanced Integration Method (AIM) API.
3	82	Scripts are only supported in version 2.5.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	83	The requested script is either invalid or no longer supported.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	84	This reason code is reserved or not applicable to this API.	
3	85	This reason code is reserved or not applicable to this API.	
3	86	This reason code is reserved or not applicable to this API.	
3	87	This reason code is reserved or not applicable to this API.	
3	88	This reason code is reserved or not applicable to this API.	
3	89	This reason code is reserved or not applicable to this API.	
3	90	This reason code is reserved or not applicable to this API.	
3	91	Version 2.5 is no longer supported.	
3	92	The gateway no longer supports the requested method of integration.	

3	93	A valid country is required.	This code is applicable to Wells Fargo SecureSource merchants only. Country is a required field and must contain the value of a supported country.
3	94	The shipping state or country is invalid.	This code is applicable to Wells Fargo SecureSource merchants only.
3	95	A valid state is required.	This code is applicable to Wells Fargo SecureSource merchants only.
3	96	This country is not authorized for buyers.	This code is applicable to Wells Fargo SecureSource merchants only. Country is a required field and must contain the value of a supported country.
3	97	This transaction cannot be accepted.	Applicable only to SIM API. Fingerprints are only valid for a short period of time. This code indicates that the transaction fingerprint has expired.
3	98	This transaction cannot be accepted.	Applicable only to SIM API. The transaction fingerprint has already been used.
3	99	This transaction cannot be accepted.	Applicable only to SIM API. The server-generated fingerprint does not match the merchant-specified fingerprint in the x_fp_hash field.
3	100	The eCheck type is invalid.	Applicable only to eCheck. The value specified in the x_echeck_type field is invalid.
3	101	The given name on the account and/or the account type does not match the actual account.	Applicable only to eCheck. The specified name on the account and/or the account type do not match the NOC record for this account.
3	102	This request cannot be accepted.	A password or transaction key was submitted with this WebLink request. This is a high security risk.
3	103	This transaction cannot be accepted.	A valid fingerprint, transaction key, or password is required for this transaction.
3	104	This transaction is currently under review.	Applicable only to eCheck. The value submitted for country failed validation.
3	105	This transaction is currently under review.	Applicable only to eCheck. The values submitted for city and country failed validation.
3	106	This transaction is currently under review.	Applicable only to eCheck. The value submitted for company failed validation.
3	107	This transaction is currently under review.	Applicable only to eCheck. The value submitted for bank account name failed validation.
3	108	This transaction is currently under review.	Applicable only to eCheck. The values submitted for first name and last name failed validation.
3	109	This transaction is currently under review.	Applicable only to eCheck. The values submitted for first name and last name failed validation.
3	110	This transaction is currently under review.	Applicable only to eCheck. The value submitted for bank account name does not contain valid characters.
3	111	A valid billing country is	This code is applicable to Wells Fargo

		required.	SecureSource merchants only.
3	112	A valid billing state/province is required.	This code is applicable to Wells Fargo SecureSource merchants only.
3	120	An error occurred during processing. Please try again.	The system-generated void for the original timed-out transaction failed. (The original transaction timed out while waiting for a response from the authorizer.)
3	121	An error occurred during processing. Please try again.	The system-generated void for the original errored transaction failed. (The original transaction experienced a database error.)
3	122	An error occurred during processing. Please try again.	The system-generated void for the original errored transaction failed. (The original transaction experienced a processing error.)
2	127	The transaction resulted in an AVS mismatch. The address provided does not match billing address of cardholder.	The system-generated void for the original AVS-rejected transaction failed.
2	141	This transaction has been declined.	The system-generated void for the original FraudScreen-rejected transaction failed.
2	145	This transaction has been declined.	The system-generated void for the original card code-rejected and AVS-rejected transaction failed.
3	152	The transaction was authorized, but the client could not be notified; the transaction will not be settled.	The system-generated void for the original transaction failed. The response for the original transaction could not be communicated to the client.
2	165	This transaction has been declined.	The system-generated void for the original card code-rejected transaction failed.
3	170	An error occurred during processing. Please contact the merchant.	Concord EFS – Provisioning at the processor has not been completed.
3	171	An error occurred during processing. Please contact the merchant.	Concord EFS – This request is invalid.
3	172	An error occurred during processing. Please contact the merchant.	Concord EFS – The store ID is invalid.
3	173	An error occurred during processing. Please contact the merchant.	Concord EFS – The store key is invalid.
3	174	The transaction type is invalid. Please contact the merchant.	Concord EFS – This transaction type is not accepted by the processor.
3	175	The processor does not allow voiding of credits.	Concord EFS – This transaction is not allowed. The Concord EFS processing platform does not support voiding credit transactions. Please debit the credit card instead of voiding the credit.
3	180	An error occurred during processing. Please try again.	The processor response format is invalid.
3	181	An error occurred during processing. Please try again.	The system-generated void for the original invalid transaction failed. (The original transaction included an invalid processor

			response format.)
2	200	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The credit card number is invalid.
2	201	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The expiration date is invalid.
2	202	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The transaction type is invalid.
2	203	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The value submitted in the amount field is invalid.
2	204	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The department code is invalid.
2	205	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The value submitted in the merchant number field is invalid.
2	206	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant is not on file.
2	207	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant account is closed.
2	208	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant is not on file.
2	209	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Communication with the processor could not be established.
2	210	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant type is incorrect.
2	211	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The cardholder is not on file.
2	212	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The bank configuration is not on file
2	213	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant assessment code is incorrect.
2	214	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This function is currently unavailable.
2	215	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The encrypted PIN field format is invalid.
2	216	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The ATM term ID is invalid.
2	217	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This transaction experienced a general message format problem.
2	218	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The PIN block format or PIN availability value is invalid.
2	219	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The ETC void is unmatched.

2	220	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The primary CPU is not available.
2	221	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The SE number is invalid.
2	222	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Duplicate auth request (from INAS).
2	223	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This transaction experienced an unspecified error.
2	224	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Please re-enter the transaction.
2	250	This transaction has been declined.	This transaction was submitted from a blocked IP address.
2	251	This transaction has been declined.	The transaction was declined as a result of triggering a Fraud Detection Suite filter.
4	252	Your order has been received. Thank you for your business!	The transaction was accepted, but is being held for merchant review. The merchant may customize the customer response in the Merchant Interface.
4	253	Your order has been received. Thank you for your business!	The transaction was accepted and was authorized, but is being held for merchant review. The merchant may customize the customer response in the Merchant Interface.
2	254	Your transaction has been declined.	The transaction was declined after manual review.

Note: Response code reasons that are not included in numerical order are reserved, or may not be applicable to this API.

Appendix A – Types of Credit Card Transactions

There are two steps to credit card transaction processing:

1. *Authorization* is the process of checking the validity and available balance of a customer's credit card before the transaction is accepted. The transaction submission methods describe the request for authorization.
2. *Settlement*, also referred to as “Capture,” is the process by which the funds are actually transferred from the customer to the merchant for goods and services sold. Based on the transaction type specified in the authorization request, the gateway will initiate the settlement step. As part of the settlement process, the gateway will send a settlement request to the financial institution to request transfer of funds. Please note that the timeframe within which funds are actually transferred is not controlled by the gateway.

Note: The merchant can specify when the last transaction is picked up for settlement by the gateway. To modify the Transaction Cut-Off Time, do the following:

1. Log into the Merchant Interface
2. Select *Settings*
3. Select *Transaction Cut-Off Time* from the General section
4. Using the drop-down boxes, select the desired cut-off time
5. Click *Submit* to save changes

Credit Card Transaction Types

The following table describes the type of transactions that can be submitted to the gateway and how the gateway will process them.

TRANSACTION TYPE	DESCRIPTION
AUTH_CAPTURE	Transactions of this type will be sent for authorization. The transaction will be automatically picked up for settlement if approved. This is the default transaction type in the gateway. If no type is indicated when submitting transactions to the gateway, the gateway will assume that the transaction is of the type AUTH_CAPTURE.
AUTH_ONLY	Transactions of this type are submitted if the merchant wishes to validate the credit card for the amount of the goods sold. If the merchant does not have goods in stock or wishes to review orders before shipping the goods, this transaction type should be submitted. The gateway will send this type of transaction to the financial institution for approval. However this transaction will not be sent for settlement. If the merchant does not act on the transaction within 30 days, the transaction will no longer be available for capture.
PRIOR_AUTH_CAPTURE	This transaction is used to request settlement for a transaction that was previously submitted as an AUTH_ONLY. The gateway will accept this transaction and initiate settlement if the following conditions are met: <ul style="list-style-type: none"> • The transaction is submitted with the ID of the original authorization-only transaction, which needs to be settled. • The transaction ID is valid and the system has a record of the original authorization-only transaction being submitted. • The original transaction referred to is not already settled or expired or errored.

	<ul style="list-style-type: none"> • The amount being requested for settlement in this transaction is less than or equal to the original authorized amount. <p>If no amount is submitted in this transaction, the gateway will initiate settlement for the amount of the originally authorized transaction.</p> <p>In addition to the required fields in the API, the following is required to submit a PRIOR_AUTH_CAPTURE type transaction:</p> <ul style="list-style-type: none"> • x_version = 3.1 • x_login = merchant Login ID • x_password = merchant password • x_trans_id = the transaction ID of the previously authorized transaction
CREDIT	<p>This transaction is also referred to as a “Refund” and indicates to the gateway that money should flow from the merchant to the customer. The gateway will accept a credit or a refund request if the transaction submitted meets the following conditions:</p> <ul style="list-style-type: none"> • The transaction is submitted with the ID of the original transaction against which the credit is being issued (x_trans_id). • The gateway has a record of the original transaction. • The original transaction has been settled. • The sum of the amount submitted in the Credit transaction and all credits submitted against the original transaction is less than the original transaction amount. • The full or last four digits of the credit card number submitted with the credit transaction match the full or last four digits of the credit card number used in the original transaction. • The transaction is submitted within 120 days of the settlement date of the original transaction. <p>A password is required to submit a credit to the system (i.e., x_password should have a valid value when a CREDIT transaction is submitted).</p> <p>For details about how to submit CREDIT transactions to the Payment Gateway, please see the Issuing Credits Guide at http://www.authorizenet.com/files/creditreturnssummary.pdf.</p>
CAPTURE_ONLY	<p>This is a request to settle a transaction that was not submitted for authorization through the payment gateway. The gateway will accept this transaction if an authorization code is submitted. x_auth_code is a required field for CAPTURE_ONLY type transactions.</p>
VOID	<p>This transaction is an action on a previous transaction and is used to cancel the previous transaction and ensure it does not get sent for settlement. It can be done on any type of transaction (i.e., CREDIT, AUTH_CAPTURE, CAPTURE_ONLY, and AUTH_ONLY). The transaction will be accepted by the gateway if the following conditions are met:</p> <ul style="list-style-type: none"> • The transaction is submitted with the ID of the transaction that has to be voided. • The gateway has a record of the transaction referenced by the ID. • The transaction has not been sent for settlement. <p>For a transaction of type VOID, the following fields are required (in addition to the other required fields in the API):</p> <ul style="list-style-type: none"> • x_version = 3.1 • x_login = merchant Login ID

	<ul style="list-style-type: none">• x_password = merchant password• x_trans_id = the transaction ID that needs to be voided
--	--

Appendix B – Features of the Gateway

The following features are supported by the gateway in an effort to reduce merchant’s chargeback liability.

Address Verification System

The Address Verification System (AVS) helps merchants to detect suspicious transaction activity. To use this system, the merchant must submit the customer’s credit card billing address to the gateway for validation. This information is submitted by the gateway to the financial institutions. The financial institutions compare the submitted address with the billing address on file for that particular credit card and return an AVS response code to the gateway. The gateway includes this code in the response back to the merchant.

The merchant can configure the gateway to reject or accept transactions based on the AVS code returned. To configure rejection or acceptance of a transaction based on the AVS code, do the following:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Address Verification System (AVS)* link from the Security section
4. Check the box(es) next to the AVS codes that the system should reject
5. Click *Submit* to save changes

AVS CODE	DESCRIPTION (<i>Italics denote a default setting</i>)
A	Address (Street) matches, ZIP does not
<i>B</i>	<i>Address information not provided for AVS check</i>
<i>E</i>	<i>AVS error</i>
<i>G</i>	<i>Non-U.S. Card Issuing Bank</i>
<i>N</i>	<i>No Match on Address (Street) or ZIP</i>
P	AVS not applicable for this transaction
<i>R</i>	<i>Retry – System unavailable or timed out</i>
<i>S</i>	<i>Service not supported by issuer</i>
<i>U</i>	<i>Address information is unavailable</i>
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

Note: It is recommended that merchants enable some level of Address Verification to avoid non-qualified transaction surcharges that can be levied by merchant banks and merchant service providers. Please note, however, that the merchant will incur applicable transaction fees for transactions that are declined due to an AVS mismatch (as with any other declined transaction). System defaults are marked in italics in the table above.

Credit Card Identification Code (CVV2/CVC2/CID)

The Credit Card Identification Code, or “Card Code,” is a three- or four-digit security code that is printed on the back of credit cards in reverse italics in the card's signature panel (or on the front for American Express cards). The merchant can collect this information from the customer and submit the data to the gateway. The gateway will pass this information to the financial institution along with the credit card number. The financial institution will determine if the value matches the value on file for that credit card and return a code indicating whether the comparison failed or succeeded, in addition to whether the card was authorized. The gateway passes back this response code to the merchant. The merchant can configure the gateway to reject or accept the transaction based on the code returned.

To configure the filter to reject certain Card Code responses, do the following:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Card Code Verification* link from the Security section
4. Check the box(es) next to the Card Codes that the system should reject
5. Click *Submit* to save changes

CARD CODE RESPONSE	DESCRIPTION
M	Card Code matched
N	Card Code does not match
P	Card Code was not processed
S	Card Code should be on card but was not indicated
U	Issuer was not certified for Card Code

Appendix C – Customizing Notification to Customers

Merchants will be sent a confirmation email after the gateway completes processing on a transaction submitted to the system. The confirmation email enables merchants to know the results of a given transaction. Multiple contacts can be configured to receive these email notifications. Additionally, merchants can choose to send a confirmation email to their customers.

Configuration of these contacts can be done through the Merchant Interface:

1. Log into the Merchant Interface
2. Click on the *Settings* link from the left navigation bar
3. Click on the *Email Receipts* link from the Transaction Response section
4. Check the box if email receipts should be sent to the customer
5. Configure the header and footer of the email message
6. Click *Submit* to save changes

It is possible to configure the confirmation email on a per-transaction basis by submitting the information with each transaction. The following table describes the fields used in the API to configure email notification to the customer; all fields are optional.

FIELD	VALUE	DESCRIPTION
x_email_customer	TRUE, FALSE	<p>If set to TRUE, the gateway will send an email to the customer after the transaction is processed using the customer email supplied in the transaction. If FALSE, no email will be sent to the customer.</p> <p>If no value is submitted, the gateway will look up the configuration in the Merchant Interface and send an email only if the merchant has configured the option to be TRUE.</p> <p>If there are no incoming parameters and the merchant has not configured this option, no email will be sent to the customer.</p>
x_header_email_receipt	Any valid text	This text will appear as the header on the transaction confirmation email sent to the customer.
x_footer_email_receipt	Any valid text	This text will appear as the footer on the transaction confirmation email sent to the customer.

Appendix D – Submitting Test Transactions to the System

Test Mode

Test Mode is a special mode of interacting with the system that is useful during the initial setup phase, when a merchant may want to test their setup without processing live card data.

To set an account to Test Mode, do the following:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Test Mode* link in the General section
4. Click on the *Turn Test On* button

In Test Mode, all transactions appear to be processed as real transactions. The gateway accepts the transactions, but does not pass them on to the financial institutions. Accordingly, all transactions will be approved by the gateway when Test Mode is turned on. Transactions submitted in Test Mode are not stored on the system, and will not appear in any reports or lists.

Note: Test Mode is only supported if the merchant is submitting transactions from a Website or through the Virtual Terminal. If the merchant uploads a file of transactions for offline processing, the gateway will reject the file.

Running a Test Transaction

It is possible to run a test transaction if Test Mode has been turned off. This can be done by indicating to the gateway in the transaction submission request that the transaction should be processed as a test transaction. The corresponding field in the transaction submission API is `x_test_request`. If a test transaction is desired, the value of this field should be set to TRUE.

The following table describes the gateway behavior based on the incoming field value and the mode configured through the Merchant Interface.

VALUE PASSED IN X_TEST_REQUEST	CONFIGURATION IN MERCHANT INTERFACE	GATEWAY BEHAVIOR
TRUE	ON	Transaction processed as test
FALSE	ON	Transaction processed as test
TRUE	OFF	Transaction processed as test
FALSE	OFF	Transaction processed as a live transaction

If there is no value submitted in the `x_test_request` field, the system will use the configuration specified in the Merchant Interface.

Test credit card numbers

Any of the following card numbers can be used to run test transactions. Please note that these numbers do not represent real card accounts; they will return a decline in live mode, and an approval in test mode. Any expiration dates after the current day's date can be used with these numbers.

TEST CARD NUMBER	CARD TYPE
370000000000002	American Express
601100000000012	Discover
542400000000015	MasterCard
4007000000027	Visa

There is also a test credit card number that can be used to generate errors. **THIS CARD IS INTENDED TO PRODUCE ERRORS**, and should only be used if that is the intent.

To cause the system to generate a specific error, set the account to Test Mode and submit a transaction with the card number 422222222222. The system will return the response reason code equal to the amount of the submitted transaction. For example, to test response reason code number 27, a test transaction would be submitted with the credit card number, "422222222222," and the amount, "27.00."

Appendix E – Certification

It is possible for a merchant to test their integration using a test gateway system. In order to test the integration, the merchant should post transactions to **<https://certification.authorize.net/gateway/transact.dll>**. The test gateway behavior will be identical to the primary gateway. Transactions sent to the test gateway are not submitted to financial institutions for authorization, will not be stored on the system, and cannot be retrieved from the system (as is the case when using Test Mode set to TRUE with the primary gateway system).

Appendix F – Currency Codes

CURRENCY COUNTRY	CURRENCY CODE
Afghani (Afghanistan)	AFA
Algerian Dinar (Algeria)	DZD
Andorran Peseta (Andorra)	ADP
Argentine Peso (Argentina)	ARS
Armenian Dram (Armenia)	AMD
Aruban Guilder (Aruba)	AWG
Australian Dollar (Australia)	AUD
Azerbaijani Manat (Azerbaijan)	AZM
Bahamian Dollar (Bahamas)	BSD
Bahraini Dinar (Bahrain)	BHD
Baht (Thailand)	THB
Balboa (Panama)	PAB
Barbados Dollar (Barbados)	BBD
Belarussian Ruble (Belarus)	BYB
Belgian Franc (Belgium)	BEF
Belize Dollar (Belize)	BZD
Bermudian Dollar (Bermuda)	BMD
Bolivar (Venezuela)	VEB
Boliviano (Bolivia)	BOB
Brazilian Real (Brazil)	BRL
Brunei Dollar (Brunei Darussalam)	BND
Bulgarian Lev (Bulgaria)	BGN
Burundi Franc (Burundi)	BIF
Canadian Dollar (Canada)	CAD
Cape Verde Escudo (Cape Verde)	CVE
Cayman Islands Dollar (Cayman Islands)	KYD
Cedi (Ghana)	GHC
CFA Franc BCEAO (Guinea-Bissau)	XOF
CFA Franc BEAC (Central African Republic)	XAF
CFP Franc (New Caledonia)	XPF
Chilean Peso (Chile)	CLP
Colombian Peso (Colombia)	COP
Comoro Franc (Comoros)	KMF
Convertible Marks (Bosnia And Herzegovina)	BAM
Cordoba Oro (Nicaragua)	NIO
Costa Rican Colon (Costa Rica)	CRC
Cuban Peso (Cuba)	CUP
Cyprus Pound (Cyprus)	CYP
Czech Koruna (Czech Republic)	CZK
Dalasi (Gambia)	GMD
Danish Krone (Denmark)	DKK
Denar (The Former Yugoslav Republic Of Macedonia)	MKD
Deutsche Mark (Germany)	DEM
Dirham (United Arab Emirates)	AED
Djibouti Franc (Djibouti)	DJF
Dobra (Sao Tome And Principe)	STD
Dominican Peso (Dominican Republic)	DOP

Dong (Vietnam)	VND
Drachma (Greece)	GRD
East Caribbean Dollar (Grenada)	XCD
Egyptian Pound (Egypt)	EGP
El Salvador Colon (El Salvador)	SVC
Ethiopian Birr (Ethiopia)	ETB
Euro (Europe)	EUR
Falkland Islands Pound (Falkland Islands)	FKP
Fiji Dollar (Fiji)	FJD
Forint (Hungary)	HUF
Franc Congolais (The Democratic Republic Of Congo)	CDF
French Franc (France)	FRF
Gibraltar Pound (Gibraltar)	GIP
Gold	XAU
Gourde (Haiti)	HTG
Guarani (Paraguay)	PYG
Guinea Franc (Guinea)	GNF
Guinea-Bissau Peso (Guinea-Bissau)	GWP
Guyana Dollar (Guyana)	GYP
Hong Kong Dollar (Hong Kong)	HKD
Hryvnia (Ukraine)	UAH
Iceland Krona (Iceland)	ISK
Indian Rupee (India)	INR
Iranian Rial (Islamic Republic Of Iran)	IRR
Iraqi Dinar (Iraq)	IQD
Irish Pound (Ireland)	IEP
Italian Lira (Italy)	ITL
Jamaican Dollar (Jamaica)	JMD
Jordanian Dinar (Jordan)	JOD
Kenyan Shilling (Kenya)	KES
Kina (Papua New Guinea)	PGK
Kip (Lao People's Democratic Republic)	LAK
Kroon (Estonia)	EEK
Kuna (Croatia)	HRK
Kuwaiti Dinar (Kuwait)	KWD
Kwacha (Malawi)	MWK
Kwacha (Zambia)	ZMK
Kwanza Reajustado (Angola)	AOR
Kyat (Myanmar)	MMK
Lari (Georgia)	GEL
Latvian Lats (Latvia)	LVL
Lebanese Pound (Lebanon)	LBP
Lek (Albania)	ALL
Lempira (Honduras)	HNL
Leone (Sierra Leone)	SLL
Leu (Romania)	ROL
Lev (Bulgaria)	BGL
Liberian Dollar (Liberia)	LRD
Libyan Dinar (Libyan Arab Jamahiriya)	LYD
Lilangeni (Swaziland)	SZL
Lithuanian Litas (Lithuania)	LTL

Loti (Lesotho)	LSL
Luxembourg Franc (Luxembourg)	LUF
Malagasy Franc (Madagascar)	MGF
Malaysian Ringgit (Malaysia)	MYR
Maltese Lira (Malta)	MTL
Manat (Turkmenistan)	TMM
Markka (Finland)	FIM
Mauritius Rupee (Mauritius)	MUR
Metical (Mozambique)	MZM
Mexican Peso (Mexico)	MXN
Mexican Unidad de Inversion (Mexico)	MXV
Moldovan Leu (Republic Of Moldova)	MDL
Moroccan Dirham (Morocco)	MAD
Mvdol (Bolivia)	BOV
Naira (Nigeria)	NGN
Nakfa (Eritrea)	ERN
Namibia Dollar (Namibia)	NAD
Nepalese Rupee (Nepal)	NPR
Netherlands (Netherlands)	ANG
Netherlands Guilder (Netherlands)	NLG
New Dinar (Yugoslavia)	YUM
New Israeli Sheqel (Israel)	ILS
New Kwanza (Angola)	AON
New Taiwan Dollar (Province Of China Taiwan)	TWD
New Zaire (Zaire)	ZRN
New Zealand Dollar (New Zealand)	NZD
Ngultrum (Bhutan)	BTN
North Korean Won (Democratic People's Republic Of Korea)	KPW
Norwegian Krone (Norway)	NOK
Nuevo Sol (Peru)	PEN
Ouguiya (Mauritania)	MRO
Pa'anga (Tonga)	TOP
Pakistan Rupee (Pakistan)	PKR
Palladium	XPD
Pataca (Macau)	MOP
Peso Uruguayo (Uruguay)	UYU
Philippine Peso (Philippines)	PHP
Platinum	XPT
Portuguese Escudo (Portugal)	PTE
Pound Sterling (United Kingdom)	GBP
Pula (Botswana)	BWP
Qatari Rial (Qatar)	QAR
Quetzal (Guatemala)	GTQ
Rand (Financial) (Lesotho)	ZAL
Rand (South Africa)	ZAR
Rial Omani (Oman)	OMR
Riel (Cambodia)	KHR
Rufiyaa (Maldives)	MVR
Rupiah (Indonesia)	IDR
Russian Ruble (Russian Federation)	RUB
Russian Ruble (Russian Federation)	RUR

Rwanda Franc (Rwanda)	RWF
Saudi Riyal (Saudi Arabia)	SAR
Schilling (Austria)	ATS
Seychelles Rupee (Seychelles)	SCR
Silver	XAG
Singapore Dollar (Singapore)	SGD
Slovak Koruna (Slovakia)	SKK
Solomon Islands Dollar (Solomon Islands)	SBD
Som (Kyrgyzstan)	KGS
Somali Shilling (Somalia)	SOS
Spanish Peseta (Spain)	ESP
Sri Lanka Rupee (Sri Lanka)	LKR
St Helena Pound (St Helena)	SHP
Sucre (Ecuador)	ECS
Sudanese Dinar (Sudan)	SDD
Surinam Guilder (Suriname)	SRG
Swedish Krona (Sweden)	SEK
Swiss Franc (Switzerland)	CHF
Syrian Pound (Syrian Arab Republic)	SYP
Tajik Ruble (Tajikistan)	TJR
Taka (Bangladesh)	BDT
Tala (Samoa)	WST
Tanzanian Shilling (United Republic Of Tanzania)	TZS
Tenge (Kazakhstan)	KZT
Timor Escudo (East Timor)	TPE
Tolar (Slovenia)	SIT
Trinidad and Tobago Dollar (Trinidad And Tobago)	TTD
Tugrik (Mongolia)	MNT
Tunisian Dinar (Tunisia)	TND
Turkish Lira (Turkey)	TRL
Uganda Shilling (Uganda)	UGX
Unidad de Valor Constante (Ecuador)	ECV
Unidades de fomento (Chile)	CLF
US Dollar (Next day) (United States)	USN
US Dollar (Same day) (United States)	USS
US Dollar (United States)	USD
Uzbekistan Sum (Uzbekistan)	UZS
Vatu (Vanuatu)	VUV
Won (Republic Of Korea)	KRW
Yemeni Rial (Yemen)	YER
Yen (Japan)	JPY
Yuan Renminbi (China)	CNY
Zimbabwe Dollar (Zimbabwe)	ZWD
Zloty (Poland)	PLN