

JETDIRECT V2 INTEGRATION

TABLE OF CONTENTS

JetDirect Audience and Use	3
The Basics	6
Beginning Your Development	6
Creating the Security Hash.....	6
Generating the Card Form	7
Optional Data Elements that May be Required.	8
Optional Fields that Carry Reportable Data	9
Optional Fields that Server as Pass-Through Data	10
Hidden Element that are Required	11
Special Requirements for URLs	11
Response Messages	12
What’s in the JetDirect Response	12
DataUrl Response Message.....	12
RetUrl Response Message.....	12
Automated Clearing House (ACH).....	13
Required Additional Elements for ACH	13
Special Considerations for ACH Forms.....	14
Specific Requirements for ACH Web Forms.....	14
Authorization for Debit From Checking Account	14
ACH Receipt Pages and Receipt Emails	15
ACH Recurring Notice Letter	16
ACH Opt-Out (Revocation) Notice Letter	16
Retention of ACH Acceptance/Forms/Data	16
Testing and Certification	17
Appendix.....	18
Section 1 – Form Elements.....	18
Section 2 – Response Details.....	21
Section 3 – Code Examples.....	23
Section 4 – ACH Specific Form Elements.....	27
Section 5 – Testing and Certification	28

JETDIRECT AUDIENCE AND USE

WHAT IS JETDIRECT

JetDirect is a blind redirect method for making payments online, where the merchant develops a credit card payment form within their site to match the look and feel of their site as to not impact the user experience. When a customer is presented with the card form page the information entered is sent as an HTTPS POST from the client's browser to the JetPay HTML/XML handler on the JetPay front end. This POST data is converted to meet the JetPay XML Schema (v2) and processed. The transaction details are passed back to the merchant's web server for inserting into their database. A request is sent to the client's browser to transition the browser where the merchant site would present an approved or declined page as well as a print page option as a receipt. By using the JetDirect method, the collection and transmission of card holder data is out sourced to the client's computer and card data is never seen by the merchants system or network. JetDirect is a single payment transaction system that handles only a few transaction types such as SALE, AUTHONLY and TOKENIZE.

AUDIENCE

The JetPay JetDirect payment method is designed for merchants that require a simple a simple web based payment method that keeps their scope of PCI involvement to a minimum while providing basic payment functionality. When implementing JetDirect under the PCI Self-Assessment Questionnaire (SAQ) a merchant will be considered to be at an SAQ level of A-EP; the second lowest of the PCI levels. See Non-Supported Integrations for exclusions

REQUIREMENTS

Merchants that are interested in using the JetDirect method must meet the following criteria.

- 1) The merchants web site cannot be part of a hosted ecommerce platform such as Shopify™, Volusion™, WiX™, Square Space™ or other platform where the web site code base is managed and maintained by the hosting company and where code modification are not allowed.
- 2) The merchant's web site may be a Content Management System (CMS) such as WordPress, Joomla or Drupal and the merchant has the technical abilities or can

- obtain the technical talent to modify the site code base to integrate the JetDirect payment method.
- 3) The merchant must operate the payments portion of their site under HTTPS (SSL) using TLS v1.2 – v2.0.
 - 4) The merchant’s card form page must have NO backwards interaction with the merchant’s web server. Once the page is loaded all actions to the page MUST be handled client side. This includes form field validation, the use of show/hide elements.
 - 5) The development of the JetDirect payment method must be able to do the following:
 - a. Collect the Total Amount to be paid in the transaction, this includes:
 - i. Taxes/VAT
 - ii. Shipping and Handling.
 - iii. Fees and Surcharging
 - b. Capture the customer or session id for this transaction
 - c. Generate a unique order number for each transaction
 - 6) Be able to use built in libraries or script own method to generate a standard SHA512 hash based on pattern outlined in the Security and Authentication section of this document.
 - 7) The merchant has completed the merchant account process and has been issued a JetPay Merchant ID and Terminal ID from JetPay.
 - a. The merchant or the merchants developer MUST obtain a VISA Developer ID* and a JetPay Developer ID* from JetPay prior to testing of their implementation.

BENEFITS OF USING

JetDirect is a very simple and in general easy implementation for any developer with a basic understanding of HTML and has the ability to use various language libraries such as JavaScript, jQuery, JSON, or other as they are needed based on the site language and server type being used for the merchant’s web site.

JetDirect allows for seamless integration in to a site with little to no impact on the user experience, implemented properly a customer visiting the site will not see the interaction of the merchant’s site and JetPay.

NOTE: The level of effort and the time of integration will depend on the complexity of the site being modified and the technical abilities of the developer and or systems personnel.

NON SUPPORTED INTEGRATION

JetDirect is intended for single payment use, meaning that the JetDirect method does not support within its frame work the abilities needed for expanded payment systems. Expanded payment systems in this definition would be:

- a. Cloud based POS Systems.
- b. POS standalone.
- c. Systems requiring the use of a card swiper or EMV chip and pin.
- d. MOTO transaction systems.
- e. Direct XML processing.
- f. Processing using JetPay Secure Tokens.
- g. Any in-house system requiring the ability to perform VOID or CREDIT.

JetPay will not support the use of the JetDirect method in the above configurations and the use of JetDirect in the above will raise the PCI SAQ level to D or Service Provider. This brings the merchant into the full PCI scope requiring scanning, intrusion detection, auditing, PA-DSS certification of application/software/service. Please consult with JetPay if your implementation is for any of the above.

SPECIAL CONSIDERATIONS FOR DEVELOPERS

The developer should take into account the need to disable the “Submit” button action on the first click OR through use of a modal set a progress .gif to alert the user that some action is taking place. Failure to do so or allowing the customer to click the “Submit” more than once can result in duplicate or multiple transactions being submitted. Further, a 30 second time out for your card page is recommended if the client does not receive a redirect in that time; the developer should auto redirect to an error page alerting the user that there is an issue and to contact the merchants customer service.

As stated previously in this document, the card page ***must not*** have any downwards interaction with the merchant’s server or network in order to be PCI SQA Level A compliant. Any form validation for required fields or completeness must take place on the client’s machine via JavaScript, AJAX, jQuery, etc. Checks may need to be implemented to check for JavaScripting being turned off in browsers and ether a notice prior to the card form.

Integration assistance please contact us at devgroupmanager@jetpay.com

THE BASICS

BEGINNING YOUR DEVELOPMENT

SECURITY AND AUTHENTICATION

As stated previously the JetDirect method is a simple POST of card data values to the JetPay front end via HTTPS. You must have installed a security certificate from a recognized certifying authority for both your production and development. This certificate must be for the domain where the implementation will take place. If the implementation is for a subdomain please ensure that your certificate is wild-carded to support all sub domains of the parent domain.

Once you have the proper security certificates installed and verified the next step is to create the authentication routine needed to validate the transaction request using an SHA512 Hash.

CREATING THE SECURITY HASH

SHA 512 HASH GENERATION

JetDirect uses an SHA512 hash as an authentication method to validate the authenticity of the transaction data sent in the HTTPS POST. You will need to generate the SHA512 hash before presenting the card form page.

The SHA512 hash is generated using shared and secret elements of the transaction and merchant configuration. These elements are ordered:

1. Terminal ID (TID)
2. Total Amount of the transaction
3. JetDirect Security Token
4. Unique Order Number.

These elements are joined together and used in for the creation of the SHA512 hash.

SHA512 Hash Generation Example

Elements to be hashed.

`TerminalID+TransactionAmount+JetDirectSecurityToken+UniqueOrderNumber`

SAMPLE Elements in hash string.

`TESTTERMINAL15.001234567890ABCDEFGHIJKabcdefghijk158898-12528`

Resulting SHA512 hash of above elements.

`2b9a00d7b8e293e2fa1a746f70379489d438ba10eacb62f2a30b530dd57c75d4f30fd7b544785f1f7db452acd8b8bb332b2e2048bf68becee15d42a4b7b0add4`

NOTE: The SHA512 Hash must be in lowercase HEX. See Appendix Section 2 Sample code for a sample of the SHA512 Generation Process used. Further, the SHA512 Hash generation should always be handled server side with your JetDirect Security Token never accessible by the card page or viewable on a “view source” of the web page.

GENERATING THE CARD FORM

Now that your certificate is installed and you have generated the authentication hash its time to create your card form. Your card form will have a number of visible and hidden elements; let’s start with the required visible elements.

For a full breakdown of all elements see: [Appendix Section 1 - Form Elements](#). See [Section 4 – Code Samples](#) for sample code scripts and a sample form.

REQUIRED VISIBLE CARD FORM ELEMENTS

Common Name	Input Name	Required	Length
Card Holder Name	name	Yes	50
*First Name	fName	dep	15
*Last Name	lName	dep	25
Card Number	cardNum	Yes	19
Expiration Month	expMo	Yes	2
Expiration Year	expYr	Yes	2
Security Code	cvv	yes	4

Example of Input field

```
<input type="text" id="req" name="name" maxlength="50" value=" " />
```

NOTE: First Name, Last Name is only to be used if you must split the Card Holder Name into separate fields, otherwise use only a single field. Elements such as Card Number and expiration Dates should always have AutoComplete="off" as part of the input.

These are the most basic card form elements that you will need input from the user in order to send a payment transaction through the JetDirect method. While there are other data elements that can be required they are based on the merchants account setting at JetPay, let’s take a look at these now.

*The use of fName and lName or split name form fields has been deprecated for v2 and is only preset for legacy information.

OPTIONAL DATA ELEMENTS THAT MAY BE REQUIRED.

BILLING ADDRESS

Common Name	Input Name	Required	Length
Address 1	billingAddress1	OPT	50
Address 2	billingAddress2	OPT	50
City	billingCity	OPT	20
State	billingState	OPT	2
Zip/Postal Code	billingZip	OPT	2
Country	billingCntry	OPT	3

While these are considered optional fields, if the merchant profile is set up for Address Verification Services (AVS) or ZIP Matching then these fields become required fields. The developer will have to discuss with the merchant if AVS or ZIP Matching has been added to the merchant account and code accordingly.

- The developer may present form fields for the customer to fill in
- Pre-populate the form fields if the billing information is stored in the customers profile
- If the information is stored under the customer profile the developer may add them as hidden fields.

Example of Input field for Address 1

Pre-filled

```
<label for=" address1"> Address 1</label>
<input type="text" id="req" name="billingAddress1" value="3361 Boyington Dr" >
```

Un-filled

```
<label for=" address1"> Address 1</label>
<input " type="text" id="req" name="billingAddress1" value=" ">
```

Hidden

```
<label for="address1"> Address 1</label>
<input type="hidden" id="req" name="billingAddress1" value="3361 Boyington Dr" >
```


OPTIONAL FIELDS THAT CARRY REPORTABLE DATA

As part of the JetDirect method there are three (3) optional field elements that can be utilized by the developer to send reportable data to JetPay as part of the standard transaction. These are User Data Fields or UD Fields.

UDFIELDS

Common Name	Input Name	Required	Length
OPEN	ud1	OPT	16 Alphanumeric
OPEN	ud2	OPT	46 Alphanumeric
OPEN	ud3	OPT	32 Alphanumeric

UD Fields can be utilized anywhere on your form as input fields or hidden fields. UD Fields are good option fields to send data that needs to be reported on with the transaction such as coupon code, fee added to transaction, terms and conditions acceptance data/time stamp etc.

Example of UD Field in use

As Input Field

```
<label for="coupon"> Coupon Code</label>  
<input type="text" id="req" name="ud1" value="">
```

Hidden

```
<input type="hidden" id="req" name="ud1" value="300FF" >
```

OPTIONAL FIELDS THAT SERVER AS PASS-THROUGH DATA

In response to numerous requests from the merchant and development community we have added an additional ten (10) pass-through data fields for your use. These fields are free form alphanumeric fields that allow you to add additional data to the transaction and have that information sent back as part of the transaction response that can be stored with the transaction data for internal use only. This data is NOT captured, stored, or available as part of the reporting from JetPay. These fields are pass-through data only. These fields can be used as live input on the card form or as hidden fields.

MER DATA FIELDS

Common Name	Input Name	Required	Length
OPEN	merData0-9	OPT	120 Alphanumeric

Example of merDataField in use

As Input Field

```
<label for="comments"> Comments</label>
<input type="text" id="req" name="merData0" value="">
```

Hidden

```
<input type="hidden" id="req" name="merData0" value="T&C Accepted
01/19/2016" >
```

HIDDEN ELEMENTS THAT ARE REQUIRED

Now that we have gone over the elements that can be viewable (or hidden as an option) in your card form let's talk about the required hidden elements. Hidden elements are elements on the form page that you need to send via JetDirect that are part of the transaction and must be present for the transaction to process.

REQUIRED HIDDEN ELEMENTS

Name	Values
trans_type	Type of transaction requested. Supported types are: SALE, AUTHONLY, TOKENIZE
cid	Customer ID, Session ID etc.
jp_tid	Merchant JetPay Terminal ID 12 character
jp_key	Key Part of the Key & Security Token Set
order_number	Unique order number used in the sha512 hash generation
jp_request_hash	The generated SHA512 hash
jpDevId	Your JetPay Developer ID
dataUrl	URL where JetDirect will POST the full response of transaction data after processed.
retUrl	URL where the customer will be redirected to after the transaction. This will generally be a receipt page.

The "cid" is as stated in the table can be the stored customer id, session id or other id used to identify the customer or the session back to your system. This is required so you can match the returned transaction data to the customer profile.

SPECIAL REQUIREMENTS FOR URLS

The dataUrl, retUrl have some special requirements for you to consider.

Both URLs are required and both must be unique.

Examples:

retUrl: <https://www.yourdomain.com/customer-receipt-page.php>

dateUrl: <https://www.yourdomain.com/process-trasnaction-response.php>

JetDirect v2 supports the addition of dynamic variable data within the urls and will pass back the dynamic data as it is sent.

Example: <https://www.yourdomain.com/customer-receipt-page.php?bookId=357dsaf&coupon=30OFF>

The JetDirect response will add its response values to your URL's provided, as an HTTPS POST.

This is the same information that you would generally see on a receipt from a terminal or from an online payment.

See Appendix Section 2 – Response Detail for a full breakdown of the response message elements.

AUTOMATED CLEARING HOUSE (ACH)

JetDirect has the ability to handle payments using personal checks via Automated Clearing House or ACH. The use of personal checks as a payment option, especially for recurring payments such as memberships, is a popular option for customers. However, ACH processing comes with a very specialized set of requirements that must be met by the merchant for not only the web standards that have to incorporate but also for their back office.

Integrating ACH within the JetDirect platform is very straight forward and is virtually the same as with integration for credit and debit card use.

REQUIRED ADDITIONAL ELEMENTS FOR ACH

Name	Values
aba	This is the routing number of the checking account where the draft of funds will be taken.
dda	This is the direct deposit account or account number of the checking account where the draft of funds will be taken.
achType	This is the type of account where the draft of funds will be taken.
chkNumber	One time use check number of the checking account where the draft of funds will be taken. Note: This number should be 1000 or greater.

Note: These elements will replace the following fields in your form: Credit Card Number, Expiration Dates, CVV2 Number.

UPDATE NAMES OF EXISTING FORM FIELDS

Name	Values
accountName	This is an update to the form field "name" used for card processing. <pre><input type="text" name="accountName" id="your-id" value="" maxlength="50"/></pre>

See Appendix Section 4 – ACH Form Elements.

SPECIAL CONSIDERATIONS FOR ACH FORMS

While the generation of the ACH form is virtually the same as it is for credit card transactions there are a few changes and additions that must be made for your form to meet the compliance standards set forth by NACHA.

SPECIFIC REQUIREMENTS FOR ACH WEB FORMS

While the form in general is similar to ones created for card transactions the ACH form must contain verification fields for both the Account Number and Routing Number.

The development of the Account and Routing number fields should do a comparison match between the two fields to validate the input.

Account Number	12345678
Confirm Account Number	••••••••
Routing Number	123456780
Confirm Routing Number	••••••••

AUTHORIZATION FOR DEBIT FROM CHECKING ACCOUNT

On the form page where the collection of the account information takes place AND before the account holder may be allowed to submit their information for payment; the merchant **MUST** collect agreement from the account holder for the draft. This can be considered a digital signature to ACH terms and conditions. This should be a checkbox mechanism and should contain the following statement.

I THE ABOVE ACCOUNT HOLDER AGREES TO THE TERMS AND CONDITIONS LISTED BELOW.

I AUTHORIZE THIS ELECTRONIC PAYMENT FROM THE FINANCIAL INSTITUTION ACCOUNT ABOVE IN THE AMOUNT LISTED ABOVE ALONG WITH ANY APPLICABLE CONVENIENCE FEE.

I ALSO UNDERSTAND THAT ANY PAYMENT MADE IS SUBJECT TO THE CURRENT TERMS AND CONDITIONS OF (THE COMPANY OR ENTITY WHICH PAYMENT IS BEING MADE) INCLUDING, BUT NOT LIMITED TO, THE BUSINESS DAY IN WHICH THIS PAYMENT WILL BE APPLIED TO MY ACCOUNT DEPENDING UPON THE DAY AND TIME THIS PAYMENT IS BEING MADE.

Once agreed to a date/time stamp **MUST** be created and added to the POST data being submitted to JetDirect. This date/time stamp should be set in User Data Field 3 (ud3). This is a required field and all steps must be taken at the time of the merchant development to ensure that the account holder CAN NOT by pass the agreement. If the account holder DOES NOT agrees to the terms and conditions the merchant must provide a mechanism for the account holder to cancel the order OR select another form of payment at that time. Further, all receipts must include the same verbiage as above and the data/time stamp of the account holders agreement to the terms and conditions.

ACH RECEIPT PAGES AND RECEIPT EMAILS

The receipts for ACH are very specific as well, you must provide an online receipt as well as an email receipt and in some instances a receipt sent via the postal services. The following MUST be included immediately after the general receipt details, in conspicuous form and no less than 10pt font.

For Single Debit or One-time ACH Draft.

ACH payments are processed at 2:00 pm Central Standard Time (CST). If you wish to revoke your authorization for this payment you must contact (MERCHANT NAME HERE) at the number below no later than 12:00 PM (CST). All ACH Draft payments made after 2:00 PM CST will be processed the next day.

(Insert Contact Number Here)

Note: On a single ACH draft a customer may agree to the authorization for debit, as an example, at 10:00 AM and call in to revoke that authorization at 10:01 AM. The Merchant will need to contact JetPay Customer Service no later than 1:00 PM CST to have the pending ACH request removed from the ACH batch to be processed at 2:00 PM CST. IF the Merchant does not contact JetPay Customer Service before 1:00 PM CST the merchant will need to issue a refund to the customer in the amount of the ACH Debit.

Recurring ACH Draft.

In the event that Recurring ACH Authorization is to be revoked or canceled, please contact (MERCHANT NAME HERE) at the number or address below at least 3 business days prior to the next billing cycle. Recurring ACH Drafts are processed on (NUMBERED DAY OF THE MONTH *example: the 15th of each month*). Note that revocation of your ACH draft does not cancel any contract or agreement that is in place. You must contact (MERCHANT NAME HERE) to arrange another payment form.

(INSERT CONTACT PHONE AND BUSINESS ADDRESS HERE)

Note: It is considered a best practice for recurring ACH payments to be handled by the merchant on the same date each month. The most common date is the 15th of the month. If the set date falls on a Saturday or Sunday the ACH Debit MUST be done on the next regular business day. Example: Recurring ACH processed on the 15th each month. The 15th falls on a Saturday during any given month the ACH should be processed on the next business day i.e. Monday the 17th.

ACH RECURRING NOTICE LETTER

Accounts that are to be recurring debits, the Merchant will send notification to the account holders listed mailing address that outlines that they have agreed to the terms and conditions of the recurring ACH debit, the date that the debits will occur each month and the amount, including any fees, taxes, and surcharges. A form for the account holder to attach a VOID'ed check from the account where the draft is to occur and a place for the account holders signature, as well as information and or a paper form for the account holder to Opt-Out (Revoke) their authorization for the recurring ACH draft. The account holder will need to return the signed form with the VOID'ed check before the next ACH draft.

Sample Text of Letter

(Customer Name) as per your agreement with (Merchant Name) on (DATE OF AUTHORIZATION) your payments will be deducted from checking account *****1234 on (NUMBER DAY OF THE MONTH) each month for the next (TIME FRAME HERE).

If you wish to cancel the recurring payment draft from your account *****1234 you will need to contact (MERCHANT NAME HERE) at the number or address below at least 3 business days prior to the next payment date.

(INSERT MERCHANT CONTACT INFORMATION)

ACH OPT-OUT (REVOCATION) NOTICE LETTER

In regards to the Opt-Out (Revocation of Authorization to Debit Checking Account).

The Checking Account holder can Opt-Out or Revoke their permission for debiting from their account at any time. However, you may now set a time frame, for recurring ACH payments only, this time frame must be considered reasonable by general business standards. The average is 3 to 5 business days prior to the next scheduled debit. Time frames that are in excess of 10 business days are not acceptable. The Opt-Out verbiage is the same as in the Recurring ACH Debit information above.

RETENTION OF ACH ACCEPTANCE/FORMS/DATA

All forms and data with regard to ACH will be retained for a period of two (2) years after the account holder stops being a customer of the merchant.

TESTING AND CERTIFICATION

Once you have completed your development and are ready to test your application you will need to do the following:

- 1) Set the action of your form to POST the form data to the JetDirect front end by using the url – <https://testapp1.jetpay.com/jetdirect/jdv2/gateway/jp-handler.php>
- 2) Add the test credentials listed in Appendix Section 5 – Testing and Certification.
- 3) Complete the required test cases in the section listed above.

After completing the test cases, submit your transaction ids from each test case to devgroupmanager@jetpay.com subject JetDirect Certification. Once the test cases have been validated you will receive the production URL as well as a statement of certification of your project or application.

APPENDIX

SECTION 1 – FORM ELEMENTS

Form Field Name	Input Name	Required	Visible/Hidden	Description
Card Holder Name	name	Y	V	Card Holder or Account Holder Name if used in a single form field.
Sample: <code><input type="text" name="name" id="your-id" value="" maxlength="50"/></code>				
Card Number	cardNum	Y	V	Credit Card Number Autocomplete="off"
Sample: <code><input type="text" name="cardNum" id="your-id" value="" maxlength="19" autocomplete="off"/></code>				
Expiration Month	expMo	Y	V	2 digit 01-12 Autocomplete="off"
Sample: <code><input type="text" name="expMo" id="your-id" value="" maxlength="2" autocomplete="off"/></code>				
Expiration Year	expYr	Y	V	2digit year i.e. 2021 = 21 Autocomplete="off"
Sample: <code><input type="text" name="expYr" id="your-id" value="" maxlength="2" autocomplete="off"/></code>				
Security Code	cvv	Y	V	Security code from card max 4 digits length numeric only Autocomplete="off"
Sample: <code><input type="text" name="cvv" id="your-id" value="" maxlength="4" autocomplete="off"/></code>				
Total Amount	amount	Y	H	Amount to be charged including tax/vat, shipping & handling and fees
Sample: <code><input type="hidden" name="amount" id="your-id" value="" maxlength="10" /></code>				
Address 1	billingAddress1	N^	V/H	First line of billing address
Sample: <code><input type="hidden" name="billingAddress1" id="your-id" value="" maxlength="50" /></code>				
Address2	billingAddress2	N^	V/H	Second line of billing address
Sample: <code><input type=" hidden" name="billingAddress2" id="your-id" value="" maxlength="50" /></code>				
City	billingCity	N^	V/H	City of billing address
Sample: <code><input type=" hidden" name="billingCity" id="your-id" value="" maxlength="12" /></code>				
State	billingState	N^	V/H	State of billing address. 2 Characters i.e. Texas = TX
Sample: <code><input type=" hidden" name="billingState" id="your-id" value="" maxlength="2" /></code>				

Zip/Postal Code	billingZip	N^	V/H	Zip of billing address Maxlength = 8 characters Does not support ZIP+ for US numbers.
Sample: <code><input type=" hidden" name="billingZip" id="your-id" value="" maxlength="8" /></code>				
Country	billingCntry	N^	V/H	Country of card holder 3 Characters international standard i.e. United States of America = "USA"
Sample: <code><input type=" hidden" name="billingCntry" id="your-id" value="" maxlength="3" /></code>				
Email Address	customerEmail	N*	V/H	Customer Email Address
Sample: <code><input type=" text" name="customerEmail" id="your-id" value="" maxlength="25" /></code>				
Phone Number	customerPhone	N*	V/H	Maxlength=12 may or may not include "-" between numbers do NOT send "(" or ")"
Sample: <code><input type=" text" name="customerPhone" id="your-id" value="" maxlength="12" /></code>				
Open Field(s)	ud1, ud2, ud3	N	V/H	User Data fields that will be captured and will be available in the GetReporting system.
Sample: <code><input type=" hidden" name="ud1" id="your-id" value="" maxlength="20" /></code> Sample: <code><input type=" hidden" name="ud2" id="your-id" value="" maxlength="40" /></code> Sample: <code><input type=" hidden" name="ud3" id="your-id" value="" maxlength="30" /></code>				
Merchant Data Fields	merData0-9	N	V/H	Non captured fields used only for pass-through information for merchant to record extra data about customer, transaction. This data not available in GetReporting
Sample: <code><input type=" hidden" name="merData0-9" id="your-id" value="" maxlength="120" /></code>				
CustomerID	cid	Y	H	Value used to connect customer and transaction in the merchants system.
Sample: <code><input type=" hidden" name="cid" id="your-id" value="" maxlength="50" /></code>				
JetPay TID	jp_tid	Y	H	Terminal ID assigned by JetPay for use with JetDirect
Sample: <code><input type=" hidden" name="jp_tid" id="your-id" value="YOUR-TID-HERE" /></code>				
JetDirect Key	jp_key	Y	H	Key portion of the JetDirect Key and Security Token set generated for use by the merchant with the

				jp_tid.
Sample: <code><input type="hidden" name="jp_key" value="YOUR-KEY-HERE" /></code>				
SHA512 HASH	jp_request_hash	Y	H	Authentication hash generated from the tid, amount, jp_token, order_number
Sample: <code><input type="hidden" name="jp_request_hash" value="GENERATED-SHA512-HASH" /></code>				
Order Number	order_number	Y	H	Unique order number generated by merchant for this transaction and used in the generation of the SHA512 hash.
Sample: <code><input type="hidden" name="order_number" value="UNIQUE-ORDER-NUMBER" /></code>				
Developer ID	jpDevId	Y	H	Developer ID issued by JetPay and is required by VISA as part of transaction data as of 2016.
Sample: <code><input type="hidden" name="jpDevId" value="YOU-JETPAY-DEV-ID" /></code>				
Data URL	dataUrl	Y	H	URL where JetDirect will pass transaction response data as an HTTPS POST sting.
Sample: <code><input type="hidden" name="dataUrl" value="HTTPS://DATAURL" /></code>				
Return URL For both approval and decline.	retUrl	Y	H	URL where customer will be redirected to if the transaction is approved. This redirect will include basic transaction receipt information sent as a base 64 encode.
Sample: <code><input type="hidden" name="retUrl" value="HTTPS://RETURL" /></code>				
Transaction Type	trans_type	Y	H	The transaction type can vary between: SALE, AUTHONLY, TOKENIZE, & CHECK
Sample: <code><input type="hidden" name="trans_type" value="SALE" /></code>				
User IP Address	userIPAddress	N	H	JetPay can store the user's IP address to show up in transaction reporting.
Sample: <code><input type="hidden" name="userIPAddress" value="127.0.0.1" /></code>				

SECTION 2 – RESPONSE DETAILS

Response Variable	Value	Description
responseText	APPROVED DECLINED	The response text provides an easily discernible value of the transaction status.
cid	Input from POST	This value is a pass-through of the value provided in the POST from the card form.
name	Input from POST	This value is a pass-through of the value provided in the POST from the card form
card	VS, MC, AX, DC	This value is the card type that was used VS=VISA MC=MASTERCARD AX=AMERICAN EXPRESS DC=DISCOVER <small>Note: Some card brands such as China Union Pay and Diners Club are issued and processed by Discover and MasterCard and will be returned as DC or MC respectively</small>
cardNum	Input from POST	This value is the last four (4) digits of the card used. Example: Card Number 4000300020001000 Returned = 1000 <small>Note: Last 4 can be displayed in the user profile as a card on file for them to use again.</small>
extendedCardNum	Input from POST	This value is the first six (6) and last four (4) of the card used with “*” filling in the missing digits. Example: Card Number 4000300020001000 Returned 400030*****1000 <small>Note: just as cardNum above this can be displayed in the user profile as a card on file for them to use again. This returned value is again PCI compliant and was added as a request of the merchant and developer community.</small>
expDate	Input from POST	This value is the expiration date of the card in MMY format. This can be stored and used by cron-jobs for finding cards that are about to expire to alert the customer to update their information.
amount	Input from POST	This value is the amount the transaction was processed.
transId	Transaction ID	This value is the transaction id assigned by JetDirect for this payment attempt. The transId is based on YYYYMMDDHHmmss+4 random characters. Example: 2016 02 01 12 12 34 07KQ <small>Note: The JetDirect method will assign Transaction ID's for each transaction you may NOT insert a custom Transaction ID for JetDirectProcessing.</small>
actCode	000-999	This value is the action code for the transaction. An actCode of 000 or 085 indicate an APPROVED transaction, all other are indications of a DECLINED transaction and have special meaning based on the number. <small>NOTE: A full listing of Action Codes can be found in the developer area of JetPay.com</small>
apprCode	6 alphanumeric characters	This value is the approval code generated by the card issuer that is connected to this transaction.
cvvMatch	1 alpha character	This value is to notify if validation of the security code passed.
addressMatch	1 alpha character	This value is to notify if the validation of the billing address was successful.
zipMatch	1 alpha character	This value is to notify if the validation of the zip code was

		successful. NOTE: Merchant must subscribe to ZIP Checking when creating the merchant account. If not subscribed the return value will always be "X". Zip Matching can be set up independently of AVS checking.
avsMatch	1 alpha character	This value is to notify if the validation of the address and zip code was successful. NOTE: Merchant must subscribe to AVS checking when creating the merchant account. If not subscribed the returned value will always be "X"
ccToken	24 alpha character	This value is a JetPay Safe Token returned by JetDirect. This token represents the card to JetPay. Safe Tokens are 100% PCI compliant and may be stored in the clear. When sending XML based transactions the Safe Token will be used in place of the full card number.
customerEmail	Input from POST	This value is the email address of the customer provided in the POST from the card form. The email address is kept as part of the transaction record at JetPay and is available in GetReporting.
order_number	Input from POST	This is the unique order number submitted in the POST from the card form.
jp_return_hash	SHA512 response hash	When a response is received from JetDirect, the merchant can validate the response is from JetPay by utilizing the hash generation used to make the jp_request_hash and appending the variable string with the value in the "responseText" variable and doing a comparison of the hash received and the hash generated. Example: <code>TerminalID+TransactionAmount+JetDirectSecurityToken+UniqueOrderNumber+responseText</code>
billingAddress1	Input from POST	If the Billing Address is supplied JetDirect will return values provided as in the HTTPS POST from the merchant.
billingAddress2	Input from POST	
billingCity	Input from POST	
billingState	Input from POST	
billingZip	Input from POST	
billingCountry	Input from POST	
rrn	Retrieval Reference Number	Number assigned when a transaction is sent to the card associations. This is sometimes needed when working with the associations in dispute resolution as well as if doing terminal certification.
uniqueid	Index reference	This value is a unique index reference that directly references the specific transaction at JetPay. This id can be used in place of a card number or Safe Token when doing VOID and CREDIT transaction requests via XML.
rawResponse	Response as received by the associations	The card association will respond with a 2 digit action code. These codes are translated into the 3 digit actCode received in the response. Most codes by the associations are mapped to the same 2 digit code. However, some are not the rawResponse code will help with troubleshooting and dispute resolution.
ud1, ud2, ud3	Input from POST	This value is the returned information that was supplied as extra reportable data from the card form page. These should be mapped to the merchant database accordingly.
merData0-9	Input from POST	This value is the pass-through data that was supplied by the card form as ancillary data that needed to be associated with the

		transaction. These should be mapped to the merchant database accordingly.
END	END	This value notates the END of the JetPay Response and closes the JetPay Connection to the merchant dataUrl and ret/decUrl.

SECTION 3 – CODE EXAMPLES

The following are code examples of some of the more complex parts of integrating JetDirect to your project. NOTE: Examples are in PHP, integration tracking puts PHP integrations at 18 to 1 in language choices.

SHA512 HASH GENERATION.

```

$jp_token = '1234567890ABCDEFGHIJKabcdefghijk';
$tid = 'TESTMERCHANT';
$amount = '100.00';
$orderNum = '1111111-A';
$jp_request_hash = hash("sha512",$tid.$amount.$jp_token.$orderNum);

```

UNIQUE ORDER NUMBER GENERATION

While most cart style sites will have a unique order number created by the system, if you are doing a site for donations as an example there may not be an order number. The code below will generate a unique order number to use in the SHA512 generation.

```

//Generate Random Order Number

function getrand() {
    // random key paramters
    $keyset =
    "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    $length = 12;

    // Random Key Generator
    $randkey = "";
    $max = strlen($keyset)-1;
    for ($i=0; $i<$length; $i++) {
        $randkey .= substr($keyset, rand(0,$max), 1);
    }
    return $randkey;
}

$orderNum = getrand();

```

BASE64 DECODE ROUTINE FOR RECEIPT RESPONSE

```
if (!strlen($_SERVER['QUERY_STRING'])) {
    exit ();
}

$string_out = base64_decode($_SERVER['QUERY_STRING']);
parse_str($string_out, $_SESSION);
```

SINGLE GENERATION SCRIPT FOR ORDER NUMBER AND SHA512

Sample SHA512 hash generator

```
//Remove special char from any of the input of the user.
function StripSpecChar($val) {
    return (preg_replace('/[^\a-zA-Z0-9" "-.\@:\/_\|/\'', ''], $val));
}

function getrand() {
    // random key paramters
    $keyset =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    $length = 12;
    $randkey = "";
    $max = strlen($keyset)-1;
    for ($i=0; $i<$length; $i++) {
        $randkey .= substr($keyset, rand(0,$max), 1);
    }
    return $randkey;
}

$orderNum = getrand();
$_SESSION['order_number'] = $orderNum;
$tid = 'TESTTERMINAL'; //once tesing is complete change this to
your production terminal id.
$amount = $_SESSION['amount'];
$jp_token = '1234567890ABCDEFGHIJKabcdefghijk';
$jp_request_hash = hash("sha512",
$tid.$amount.$jp_token.$orderNum); $_SESSION['jp_request_hash'] =
$jp_request_hash; $_SESSION['order_number'] = $orderNum;

header('location:lg-cardform.php');
```


MYSQL DATABASE SQL BUILD SAMPLE

```
CREATE TABLE IF NOT EXISTS `TABLE_NAME_HERE` (  
  `id` int(200) NOT NULL AUTO_INCREMENT,  
  `cid` varchar(72) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  `jp_return_hash` varchar(250) CHARACTER SET utf8 COLLATE utf8_unicode_ci  
  DEFAULT NULL, `responseText` varchar(20) CHARACTER SET utf8 COLLATE  
  utf8_unicode_ci DEFAULT NULL, `name` varchar(50) CHARACTER SET utf8  
  COLLATE utf8_unicode_ci DEFAULT NULL, `cardNum` varchar(19) CHARACTER SET  
  utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `extendedCardNum` varchar(19)  
  CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `card` varchar(2)  
  CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `expDate`  
  varchar(2) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  `amount` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT  
  NULL, `transId` varchar(18) CHARACTER SET utf8 COLLATE utf8_unicode_ci  
  DEFAULT NULL, `actCode` varchar(3) CHARACTER SET utf8 COLLATE  
  utf8_unicode_ci DEFAULT NULL, `apprCode` varchar(10) CHARACTER SET utf8  
  COLLATE utf8_unicode_ci DEFAULT NULL, `cvvMatch` varchar(2) CHARACTER SET  
  utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `addressMatch` varchar(2)  
  CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `zipMatch`  
  varchar(2) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  `avsMatch` varchar(2) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT  
  NULL, `ccToken` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci  
  DEFAULT NULL, `customerEmail` varchar(50) CHARACTER SET utf8 COLLATE  
  utf8_unicode_ci DEFAULT NULL, `customerPhone` varchar(10) CHARACTER SET  
  utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `order_number` varchar(15)  
  CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `rrn`  
  varchar(15) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  `uniqueId` varchar(15) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT  
  NULL, `order_number` varchar(15) CHARACTER SET utf8 COLLATE  
  utf8_unicode_ci DEFAULT NULL, `ud1` varchar(50) CHARACTER SET utf8 COLLATE  
  utf8_unicode_ci DEFAULT NULL, `ud2` varchar(50) CHARACTER SET utf8  
  COLLATE utf8_unicode_ci DEFAULT NULL, `ud3` varchar(50) CHARACTER SET  
  utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `merData1` varchar(120)  
  CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `merData2`  
  varchar(120) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  `merData3` varchar(120) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT  
  NULL, `merData4` varchar(120) CHARACTER SET utf8 COLLATE utf8_unicode_ci  
  DEFAULT NULL, `merData5` varchar(120) CHARACTER SET utf8 COLLATE  
  utf8_unicode_ci DEFAULT NULL, `merData6` varchar(120) CHARACTER SET utf8  
  COLLATE utf8_unicode_ci DEFAULT NULL, `merData7` varchar(120) CHARACTER  
  SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `merData8` varchar(120)  
  CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL, `merData9`  
  varchar(120) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  `merData0` varchar(120) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT  
  NULL, PRIMARY KEY (`transId`)) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Note: While the above can be used for your testing and production database build, you may need to modify by adding columns to support your project and any special data that you may need to capture from the full transaction response message. This would refer to dynamic data added to your dataUrl.

MYSQL INSERT ROUTINE SAMPLE

```

$hostname='YOUR_HOST_NAME';
$username='YOUR_USER_NAME';
$password='YOUR_PASSWORD';
$dbname='YOUR_DATABASE_NAME';
$con = mysql_connect($hostname, $username, $password, dbname);

if (!$con) {
die('Could not connect: ' . mysql_error());
}

$sql="INSERT INTO `DBNAME`.`transactions` (`cid`,`jp_return_hash`
`,`responseText`,`name`,`cardNum`,`card`,`cardNum`,`extendedCardNum`
`,`amount`,`transId`,`actCode`,`apprCode`,`cvvMatch`,`addressMatch`
`,`zipMatch`,`avsMatch`,`ccToken`,`customerEmail`,`customerPhone`
`,`order_number`,`ud1`,`ud2`,`ud3`,`merData1`,`merData2`,`merData3`
`,`merData4`,`merData5`,`merData6`,`merData7`,`merData8`,`merData9`
`,`merData0`,`rrn`,`cuiuniqueid`) VALUES ('$_REQUEST[cid]',
'$_REQUEST[jp_return_hash]', '$_REQUEST[responseText]', '$_REQUEST[name]',
'$_REQUEST[cardNum]', '$_REQUEST[card]', '$_REQUEST[cardNum]',
'$_REQUEST[extendedCardNum]', '$_REQUEST[amount]', '$_REQUEST[transId]',
'$_REQUEST[actCode]', '$_REQUEST[apprCode]', '$_REQUEST[cvvMatch]',
'$_REQUEST[addressMatch]', '$_REQUEST[zipMatch]', '$_REQUEST[avsMatch]',
'$_REQUEST[ccToken]', '$_REQUEST[customerEmail]',
'$_REQUEST[customerPhone]', '$_REQUEST[order_number]', '$_REQUEST[ud1]',
'$_REQUEST[ud2]', '$_REQUEST[ud3]', '$_REQUEST[merData1]',
'$_REQUEST[merData2]', '$_REQUEST[merData3]', '$_REQUEST[merData4]',
'$_REQUEST[merData5]', '$_REQUEST[merData6]', '$_REQUEST[merData7]',
'$_REQUEST[merData8]', '$_REQUEST[merData9]', '$_REQUEST[merData0]',
'$_REQUEST[rrn]', '$_REQUEST[uniqueid]');";

if (!mysql_query($sql,$con)) {
die('Error: ' . mysql_error());
}

mysql_close($con);

```

SECTION 4 – ACH SPECIFIC FORM ELEMENTS

Form Field Name	Input Name	Required	Visible/Hidden	Description
Account Holder	accountName	Y	V	Account Holder Name
Sample: <code><input type="text" name="name" id="your-id" value="" maxlength="50"/></code>				
Account Number	ddaDrop	Y	V	Visible Account Number
Sample: <code><input type="text" name="abaDrop" id="your-id" value="" maxlength="10" autocomplete="off"/></code>				
Confirm Account Number	dda	Y	V	Visible Account Number with password mask
Sample: <code><input type="password" name="expMo" id="your-id" value="" maxlength="10" autocomplete="off"/></code>				
Routing Number	abaDrop	Y	V	Visible Routing Number
Sample: <code><input type="text" name="abaDrop" id="your-id" value="" maxlength="10" autocomplete="off"/></code>				
Confirm Routing Number	aba	Y	V	Visible Routing Number with password mask
Sample: <code><input type="password" name="aba" id="your-id" value="" maxlength="4" autocomplete="off"/></code>				
Check Number	chkNumber	Y	H	Check number to be used for the ACH debit. Number should be greater than 1000
Sample: <code><input type="text" name="chkNumber" id="your-id" value="" maxlength="5" /></code>				

All other JetDirect element with the exception of Billing Address, Expiration Dates, CVV2 fields may be used with the ACH Specific Form Elements.

SECTION 5 – TESTING AND CERTIFICATION

MERCHANT VARIABLES

Variable	Value
tid	TESTTERMINAL
jp_key	1234567890abcdefghijk
jp_token	1234567890ABCDEFGHIJKabcdefghijk

You will receive a new jp_key and jp_token based on the production Terminal ID (TID) assigned by JetPay prior to going into production.

STANDARD TRANSACTION TESTING

The following test cases consist of common JetDirect transactions that many merchants will need to be able to perform. To certify, these test cases will need to be submitted for review.

The following test cases will certify standard sales transactions. The credit card used for these test cases will be 4111 1111 1111 XXXX where XXXX is the PAN listed in the table.

Test Case	Type	PAN	Exp Date	Amount	Result
STND001	SALE	1111	12/22	\$10.00	Approved
STND001a	TOKENIZE	1111	12/22		Tokenized
STND001b	AUTHONLY	1111	12/22	\$10.00	Approved
STND002	SALE	1129	01/15	\$10.00	Expired Card
STND003	SALE	1128	12/22	\$10.00	Invalid Card Number
STND004 Through STND051	SALE	1137	12/22	Amounts between \$1.00 and \$2.00 i.e. \$1.01, \$1.02 etc.	Assorted responses both approved and declined. These test cases are always required.

- If your development does not allow for expired card you may bypass test case STND002.
- If your development does not process AUTHONLY transaction types you may bypass test case STND001b
- If your development will not be tokenizing credit card information as a stand-alone process, meaning not related to a SALE or AUTHONLY transaction type, you may bypass test case STND001a

CVV 2 TEST CASES

The credit card companies support multiple methods for validating credit cards and verifying a cardholder's identity. JetPay supports CVV2, CVC2, and CID card validation, available from Visa, MasterCard, and American Express, as well as all the other major credit card brands. JetPay supports address verification ("AVS") as implemented by Visa, MasterCard, and American Express (as well as Amex's name verification feature). With these two common features, a credit card's integrity can be validated and a cardholder's billing address can be verified. Even more importantly, these two features reduce transaction charges and help to protect a merchant against charge backs.

The following test cases will certify CVV2 transactions. The credit card used for these test cases use the following format: 4666 6666 6666 XXXX where XXXX is the PAN listed in the table.

Test Case	Type	PAN	Exp Date	Amount	CVV2	Result
CVV201	SALE Or AUTHONLY	6669	12/22	\$10.00	321	Approved, CVV Matches
CVV202	SALE Or	6677	12/22	\$10.00	432	Approved, CVV Doesn't Match
CVV203	SALE Or AUTHONLY	6685	12/22	\$10.00	543	Approved, CVV not processed
CVV204	AUTHONLY	6693	12/22	\$0.00	021	Approved, CVV Matches
CVV205	SALE or AUTHONLY	6610	12/22	\$10.05	123	Declined, CVV Not Processed
CVV206	SALE or AUTHONLY	6628	12/22	\$12.51	089	Declined, CVV Matches

CVV VALUES THAT AFFECT THE CVV RESULT

By varying the third digit of the CVV2 value, you can get multiple different CVV2 results. The following table lists the various CVV2 responses sent by the back-end simulator for specific values:

3 rd CVV Digit 4 th CVV Digit	CVV Result	Description
0	M (all)	CVV2/CVC2/CID Match
1	M (all)	CVV2/CVC2/CID Match
2	N (all)	CVV2.CVC2/CID Does Not Match
3	P (all)	CVV2/CVC2/CID Not Processed
4	U (VS/MC/DS) P (AMEX)	Issuer not certified CID Not Processed

5	S (VS/MC/DS) P (Amex)	Merchant indicates no CVV2/CVC2 on card. CID Not Processed
6	M (all)	CVV2/CVC2/CID Match
7	M (all)	CVV2/CVC2/CID Match
8	M (all)	CVV2/CVC2/CID Match
9	M (all)	CVV2/CVC2/CID Match

CVV RESULT CODES

CODE	DEFINITION
M	Match on CVV2/CVC2/CID. Valid CVV2/CVC2/CID
N	No match on CVV2/CVC2/CID. Invalid CVV2/CVC2/CID
P	Not processed or unable to process.
S	CVV2/CVC2 should be on the card, but the merchant indicates that it is not.
U	Issuer is not certified for CVV2 or has not provided encryption keys.

NOTE: Participation in CVV2/CVC2 is optional for issuers. The subscribing issuers submit their CVV2/CVC2 keys to Visa and MasterCard, and these keys are kept secret. The individual CVV2/CVC2 implementation policies of the tens of thousands of issuers are confidential, and issuers may change their internal CVV2/CVC2 policies without notification. Statistics are unavailable as to how many issuers subscribe (or don't subscribe) to CVV2/CVC2.

ADDRESS VERIFICATION (AVS) TEST CASES

The following test cases will certify address verification transactions using AVS. The credit card used for these test cases use the following format: 4777 7777 7777 XXXX where XXXX is the PAN listed in the table.

Note: In the production system AVS must be set up as part of the JetPay merchant profile. If your application will not be doing AVS checking you will not need to complete this section of test certification.

Test Case	Type	PAN	Exp Date	Amount	Address Postal Code	Result
AVS01	SALE Or AUTHONLY	7711	12/22	\$11.00	1234 Fifth Street 77708	Approved, both address & zip match
AVS02	SALE Or AUTHONLY	7729	12/22	\$11.00	1234 Fifth Street 11100	Approved, address match, zip not match
AVS03	SALE Or AUTHONLY	7737	12/22	\$11.00	1234 Fifth Street 88809	Approved, zip match, address does not match
AVS04	SALE or AUTHONLY	7745	12/22	\$11.00	1234 Fifth Street 44403	Approved, address and zip do not match
AVS05	SALE or AUTHONLY	7778	12/22	\$11.00	1234 Fifth Street 33312	Approved, international non-AVS participant
AVS06	SALE or AUTHONLY	7810	12/22	\$12.51	1234 Fifth Street 00006	Declined, AVS Unavailable

POSTAL CODE VALUES THAT AFFECT THE AVS RESPONSE

Last 2 Digits	AVS Result	Description
00	A (all)	Address matches, postal code absent or does not match. (All)
01	D (all)	Street address and postal codes match. (Visa/MC)
02	M (all)	Street address and postal codes match. (Visa/MC)
03	N (all)	Neither address nor postal code matches.
04	R (all)	Retry (all)
05	S (all)	AVS Unavailable (all)
06	U (all)	AVS Unavailable (all)
07	Z (Visa) W (MC/AMEX)	W replaced by Z (VS) Postal code matches, address absent or does not match. (MC/AMEX)
08	Y (all)	Street address and postal code match. (All)
09	Z (all)	Postal code matches, address absent or does
10	B (VS/MC) E (AMEX)	Street address and postal code have invalid formats. CM Name incorrect. Street address and postal

		code match.
11	C (VS/MC) K (AMEX)	Street address and postal code have invalid formats. CM Name matches.
12	G (VS/MC) L (AMEX)	Non-AVS participant outside U.S. Address not verified. CM Name and postal code match.
13	I (VS/MC) O (AMEX)	Street address not verified for international transaction. CM Name and address match.
14	P (VS/MC) F (AMEX)	Postal Code match. Street address has invalid format. CM Name incorrect. Street address matches.
15	X (VS/MC)	Nine-digit postal code match in U.S. postal code and address match for outside U.S.
16	F (VS)	Street address and postal code match for UK only.

AVS RESULT CODES

JetPay offers cardholder address verification through AVS, a system available widely throughout North America and much of Europe. With address verification, the merchant submits a customer's address and postal code information in a transaction; the credit card issuer verifies and validates the address information against their internal billing address data.

Visa and MasterCard both call this feature AVS, which stands for "Address Verification Service." American Express calls this feature AAV, which stands for "Address Authentication and Verification."

Almost all North American issuers support AVS using Visa and MasterCard association rules. According to these rules, an issuer compares the numeric digits within a cardholder's address and postal code information with their own stored billing information. All alphabetic information is ignored. The address and postal code match (or fail to match) and an AVS result code is generated by the issuer. This is returned to the merchant during an authorization.

American Express adds cardholder name matching to their AAV. A merchant can submit their customer's name in addition to the billing address information and may receive a number of additional AVS result codes. Shipping address information may also be submitted in an Amex transaction, and additional result codes are anticipated from Amex in the future.

For most issuers, the AVS works independent of the action code. In other words, a transaction may be approved even though billing address information doesn't match. Because transaction approval is independent of billing address integrity, it's up to the merchant to decide the importance of correct billing address information when completing a sale. A merchant may simply allow a transaction to proceed, or a merchant might independently discontinue the sale in spite of an approval.

JetPay, based on the AVS result code, automatically fills the AddressMatch and ZipMatch elements. The possible AddressMatch and ZipMatch values are "Y", "N", or "X", and the value is deterministic with the AVS result code.

JetPay offers an "Automatic AVS Rejection" feature, where merchants may automatically decline any transactions showing degraded AVS results. Automatic AVS Rejection is an optional subscription service that requires a merchant to subscribe before JetPay will automatically reject transactions on behalf of the merchant.

A result code indicating the AVS ("Address Verification Service") results for a transaction may be:

A	Address matches, but postal code does not match.
B	Visa only: Street address matches, postal code not verified because of
C	Visa only: Street address and postal code not verified because of
D	Visa only: Street address and postal code match. Amex only: Cardholder
E	Amex only: Cardholder name incorrect, address and postal code matches.
F	Amex only: Cardholder name incorrect, billing address matches.
G	Visa only: Global non-AVS participant outside the U.S.; address is not
K	Amex only: Cardholder name matches.
L	Amex only: Cardholder name and postal code match.
M	Amex only: Cardholder name, address and postal code match. Visa only: Street address and postal code match.
N	Neither the address nor the postal code matches.
O	Amex only: Cardholder name and address match.
P	Visa only: Postal code matches, street address not verified because of
R	Retry, system unable to process.
S	MasterCard only: AVS currently not supported.
U	Unavailable. Address not verified. AVS attempted, but no AVS information was available (issuer is not an AVS participant or issuer
W	MasterCard only: Nine-digit postal code and address match. Amex only: Cardholder name, address, and postal code are all incorrect.
X	MasterCard only: Five-digit postal code and address match.
Y	Both the address and the postal code match exactly.
Z	Postal code matches, but address does not match.

ACH TEST CASES

Merchants may submit bank drafts to JetPay. JetPay can enable properly configured merchants to send checks and savings account drafts directly into their bank account. By submitting the ABA routing code listed on every check along with the bank account number and the accountholder's name, JetPay will route these bank drafts directly into the merchant's bank account. Refunds of bank drafts are also available.

Bank accounts submitted through ACH require an ABA number, and account number, an accountholder's name, and an identifying check number. The following processes prescribe the test cases for ACH processing. For Approval Testing use routing code of 122000496, for Decline Testing use routing number of 654654654. For the cardholder name, use "John Q. Public" or the name of your favorite Star Wars character.

Test Case	Type	Account Number	Check Number	Amount	Expected Result
ACH01	CHECK	123411	234	\$10.00	Approved
ACH02	CHECK	552777	567	\$10.00	Approved
ACH03	CHECK	765432	890	\$10.00	Approved
ACH04	CHECK	3213211	12222	\$10.00	Decline
ACH05	CHECK	1598887	55622	\$10.00	Decline