



Gateway Integration Specifications Credit Card Processing

Version 3.1

Trine Commerce Systems, Inc.

2613 Wilson Street
Austin, TX 78704
512-586-2736
legal@TrineCS.com
techsupport@TrineCS.com

Legal Notice

All content of this manual, the Web site and Programming
Copyright © by Trine Commerce Systems, Inc.
All Rights Reserved

Version 06/07/2011

Introduction

The following information is provided to clients of Trine Commerce Systems, Inc. and its system and network partners for integration of the payment gateway into their internal networks. These instructions can also be used to integrate a secure Web page form with the gateway.

Content

1.	HTML Form Requirements	2
2.	Swipe/Magnetic Card Requirements	3
3.	Fields Available But Not Required	4
4.	XML Processing	5
4. a.	Processing VOID Transactions	6
4. b.	Special Case: Fee Information Request	7
4. c.	Optional Fields Not Communicated to Network – Internal Use Only	8
4. d.	Transaction Report Retrieval	9
5.	HTML Responses From The Network	10
6.	Legal Notice	11

1. HTML Form Requirements

Gateway URL: `https://www.trinecs.net/processor/transact/form`
Form `method="post"`
`action="https://www.trinecs.net/processor/transact/form"`
 (Example form can be found at www.TrineCS.com/docs)

Test server: Use the Test Server URL for integration programming and testing. The hash provided for testing is unique to the testing environment. A new production hash will be provided upon completion of integration programming.

In all cases, replace the live server URL with the following:
`https://test.trinecs.com/...`

Required fields:

hash Hidden 32 character identification string assigned by the server.

response_type Hidden number telling the gateway what type of response it should have for a successful or a failed transaction. 1 means that the pages will be specified in the form fields **response_success** and **response_failure** and the browser will be forwarded to one of these pages. 0 means that the gateway will simply print out a page to the browser, successful or not.

response_success

This is the web page that a person should be forwarded to if the transaction has been successful. Must be a fully qualified and valid URL. Is only required if **response_type** = 1 and is a hidden field.

response_failure

The web page that a person will be forwarded to if the transaction is not successful. A form field called **error** will be appended to the URL and contains a comma separated list of reasons why the transaction failed. Must be a fully qualified and valid URL. Is only required if **response_type** = 1 and is a hidden field.

See the “HTML Response from Network” section below.

card_number	The credit card number being processed.
card_date	Expiration date on the card in the format of MM/YY
card_name	Name of the cardholder.
amount	Total amount of the transaction
transaction_type	Specifies the type of transaction. In most cases: <option value= 'SALE'>Purchase/Card Not Present</option> The only transaction type supported in HTML forms is SALE.

2. Swipe/Magnetic Card Transactions

It is recommended to use the XML method for processing magnetic card transactions. It tends to be cleaner and provides more feedback fields. (See also 4. XML Processing.)

Gateway URL: https://www.trinecs.net/processor/transact/form (http post method)
 https://www.trinecs.net/processor/transact/xml (xml method)

Required fields:

hash	Hidden 32 character identification string assigned by the server.
mag_data	Data from the card magnetic swipe. There are two methods for sending magnetic stripe data. Either A) send the entire mag stripe in the variable mag_data <u>without the initial % or closing ?</u> or B) send two separate fields: mag_track1 with track one data in it, and mag_track2 with track two data in it. For many purposes, track 1 is optional.
amount	Total amount of the transaction

transaction_type	Specifies the type of transaction. In most cases: <option value= 'SALE'>Purchase/Card Not Present</option> Transaction types supported are: SALE, CREDIT, PREAUTH, and FORCE
pin_block	For PIN debit transactions only. The pin number encrypted with the key provided to the card stripe reader device. The user enters their PIN on device, which encrypts the pin with the key provided by the network, and sends the results in this field.

3. Fields Available But Not Required

cvv2	The last three digits of the number printed on the back of the credit card (Mastercard/Visa, Discover). While not required, omission of this number may result in a downgrade of the transaction by the network. American Express uses a 4-digit card ID.
card_type	Type of card (Visa, Mastercard, etc.)
name	The billing name. While in most cases this would be the same as the cardholder's name, it is kept separate just in case it is needed.
merchant_1 merchant_2	User-configurable fields. These fields may be visible or hidden. They do appear in email confirmations. 30 character limit each.

Other Billing/Information Fields:

company address1 address2 city state country zip phone fax email	Address is required for AVS verification. 2 digit state code, CA, TX, MB (Manitoba), etc. 2 digit ISO country code, US, CA, etc. Zip code is required for AVS verification
purchase_data	Level 2 data for corporate purchase cards [varchar(17)].

Shipping Information

shipping_name
shipping_company
shipping_address1
shipping_address2
shipping_city
shipping_state
shipping_country
shipping_zip

Order Information

description	A free-text field describing what's been sold.
sub_total	Product/Service cost not including shipping or taxes
shipping_type	A free text field describing the shipping method. e.g. USPS Priority Mail
shipping_cost	Cost of shipping and/or handling
tax	Total sales tax charged

4. XML Processing

Gateway URL	https://www.trinecs.net/processor/transact/xml Transaction types supported are: SALE, CREDIT, PREAUTH, FORCE and VOID (see 4.a. below)
--------------------	--

Additional fields available for transactions with fees:

These fields are used only in very specific cases unique to government merchant transactions. They may be omitted if not used by the merchant.

fee	amount of fee
feetrans_ident	identifier of the fee
feetrans_invnum	invoice number of fee

Example XML Request (with example text):

```

<request>
<hash>1a3255abdbe5b764d66fef79eb85d4dc</hash>
<transaction_type>SALE</transaction_type>
<response_type>0</response_type>
<name>Buyer Name</name>
<card_name>Name On Card</card_name>
<card_number>0123456789</card_number>
<card_type>VISA</card_type>
<card_date>10/10</card_date>
<card_entry>KEYED</card_entry> (alt: SWIPED)
<company>Company Name</company>
<address1>130 Company Address</address1>
<address2>Suite #512</address2>
<city>Company Town</city>
<state>WA</state>
<country>US</country>
<zip>12345-6789</zip>
<phone>123-456-7890</phone>
<fax>123-456-7891</fax>
<email>email@email.com</email>
<purchase_data>12345678901234567</purchase_data>
<shipping_name>Name Shipped To</shipping_name>
<shipping_company>Company Name</shipping_company>
<shipping_address1>130 Company Road</shipping_address1>
<shipping_address2>Suite #512</shipping_address2>
<shipping_city>Company Town</shipping_city>
<shipping_state>WA</shipping_state>
<shipping_country>US</shipping_country>

```

```

<shipping_zip>12345-6789</shipping_zip>
<description>Name of tem ordered</description>
<invoice_num>1000</invoice_num>
<sub_total>10.00</sub_total>
<shipping_type>USPS Priority Mail</shipping_type>
<shipping_cost>2.00</shipping_cost>
<tax>1.00</tax>
<amount>13.00</amount>
<fee>1.00</fee> *
<total>14.00</total>
<feetrans_ident>000123456</feetrans_ident> *
<feetrans_invnum>12345</feetrans_invnum> *
</request>

```

Example XML Response (with example text):

```

Success      <response>
              <isError>No</isError>
              <time>2001-06-13 20:24:30</time>
              <invoice_num>1000</invoice_num>
              <merchant_id>0001234</merchant_id> (from network)
              <cardholder_name>JOHN SMITH</cardholder_name> (from mag stripe)
              <card_type>VISA</card_type>
              <card_entry>KEYED</card_entry>
              <account>xxxxxxxxxxx7890</account> (masked to last 4 digits)
              <expiration_date>10/10</expiration_date>
              <feetrans_ident>000123456</feetrans_ident> *
              <feetrans_invnum>12345</feetrans_invnum> *
              <amount>13.00</amount>
              <fee>1.00</fee> *
              <total>14.00</total>
            </response>

```

**Fee fields may be omitted if no fee is included. (Fees are set by the Merchant per terminal and are applied automatically, if set.)*

```

Error      <response>
            <isError>Yes</isError>
            <error>Bad Card Number</error>
            <error>Bad Name on card</error>
          </response>

```

4. a. Processing Void Transactions

Purpose	To Void a previously processed transaction.
Process	The original transaction will return an identifier that is unique to the transaction. This identifier may vary by processor. The identifier is then used in a second transaction in the invoice_num field, with VOID as the transaction_type.
URL	https://www.trinecs.net/processor/transact/xml

Example XML Request (with example text):

```

<request>
  <transaction_type>VOID</transaction_type>
  <invoice_num>123</invoice_num>
  <hash_id>123124124124</hash_id>
</request>

```

Example XML Response (with example text):

```

Success      <response>
              <isError>No</isError>
              <time>2007-20-10 12:12:34</time>
              <invoice_num>123</invoice_num>
              </response>

Error        <response>
              <isError>Yes</isError>
              <error>Some Error Message</error>
              </response>

```

4. b. Special Case: Fee Information Request

Purpose: Through an xml api call, request a fee calculation for an amount on a particular terminal.

Card URL: <https://www.trinecs.net/processor/transact/feeinfo>

Fields:

amount	amount requested
fee	calculated fee for given amount
basefee	base fee applied
feepercent	percentage applied

Calculation algorithm:

The fee field has been pre-calculated for you. The basefee and feepercent is extra data that you may or may not need (e.g. in case you need to explain how the fee was calculated).

```

if (amount * feepercent/100) > basefee)
  return amount * feepercent;
else
  return basefee;

```

Example XML Request (with example text):

```
<request>
<hash>123124124214</hash> *
<amount>10.00</amount>
</request>
```

* The hash for a terminal with an associated terminal to accept the fee is required. This is set on the gateway by the merchant account representative.

Example XML Response (with example text):

```
Success Case: <response>
<isError>No</isError>
<time>2007-07-23 14:23:00</time>
<amount>10.00</amount>
<fee>1.00</amount>
<basefee>1.00</basefee>
<feepercent>2.3</feepercent>
</response>

Failure Case: <response>
<isError>Yes</isError>
<error>Invalid User</error>
</response>
```

4. c. Optional Fields Not Communicated to Network – Internal Use Only

terminal_vendor	maxlen: 40 char Indicates the vendor of the device sending the query.
terminal_model	maxlen : 40 char indicates the model of the device sending the query
terminal_serial_number	maxlen: 40 char Indicates the serial number of the device sending the query
terminal_prev_response	maxlen: 40 char
terminal_sequence	maxlen: 40 char Sequence number that device has for transaction.
device_transaction	maxlen:40 char This should be a UNIQUE identifier for EACH transaction done on the device. It is up to the developer to insure uniqueness within the set of transactions done on that device.

The device_transaction field can be used to query the results of the transaction after the transaction is complete using the QUERY xml request and passing the device_transaction in the request.

If the developer does not maintain uniqueness, a QUERY request will indicated invalid device_transaction as an error.

Sample QUERY request:

```

<request>
<hash>EEC2FA7D9BDB327C4BC307B513279558</hash>
<response_type>0</response_type>
<transaction_type>QUERY</transaction_type>
<device_transaction>12312412412412</device_transaction>
</request>

```

4. d. Transaction Report Retrieval

Purpose: Transactions for any date and terminal can be downloaded for integration to client networks and programs.

URL: <https://www.trinecs.net/virtual/report/transactions>

Fields

<hash>	Required
<passcode>	Required – Set as an option in the merchant terminal
<date>	Optional – Default is current date

Fields Returned

The following fields are returned if not blank. Blank fields are NOT returned.

transaction_date	shipping_cost
display_trantype	card_type_code
status	shipping_name
terminal_id	shipping_address1
card_number	shipping_address2
approval_code *	shipping_city
name	shipping_state
company	shipping_zip
address1	card_name
address2	merchant_1
city	merchant_2
state	ipaddress
country	transaction_identifier
phone	convenience_fee
fax	is_pintx
email	is_mag
amount	is_pinless
sub_total	cvv_meaning
tax	auth_network_name
description	void_indicator
shipping_type	

* <approval_code> is the network's authorization (auth) code.

Sample Request (Actual on test server)

<http://test.trinecs.com/virtual/report/transactions?hash=6aefa59be60983b3a6a5a6bfbe5988ee&date=2009-05-10&passcode=fancypass>

Example XML Response (with example text):

```
<report_transactions date='2009-05-10'>
  <transaction>
    <id>622</id>
    <transaction_date>2009-05-10 02:00:07</transaction_date>
    <display_trantype>SALE</display_trantype>
    <status>FAIL</status>
    <terminal>fdadmin</terminal>
    <card_number>*****1111</card_number>
    <name>Tom</name>
    <address1>789</address1>
    <city>Harrisburg</city>
    <state>PA</state>
    <country>US</country>
    <email>denavinson@austin.rr.com</email>
    <amount>5.00</amount>
    <sub_total>0</sub_total>
    <tax>0</tax>
    <shipping_cost>0</shipping_cost>
    <shipping_city>Harrisburg</shipping_city>
    <shipping_state>PA</shipping_state>
    <card_name>Tom</card_name>
    <ipaddress>74.92.222.101</ipaddress>
    <convenience_fee>0</convenience_fee>
    <is_pintx>N</is_pintx>
    <is_mag>N</is_mag>
    <is_pinless>Y</is_pinless>
    <void_indicator>n</void_indicator>
  </transaction>
</report_transactions>
```

5. HTML Responses From The Network

Responses from the network to HTML form processing are handled by hidden field settings in the form. (See response_type in “HTML Form Requirements” above.) Either a simple page with the transaction information will be displayed, or the Gateway can be set to forward the response information to a client-defined Web page. Separate “success” and “error” pages are needed for custom network responses. These are defined in the Required form fields section.

A field named “message” is returned upon a successful transaction, or a field named “error” on a failed transaction. The existence of one of those fields will indicate a failed or successful transaction. The field returned will contain either transaction information, if successful, or an error message, and is automatically added to the end of the redirecting URL. Standard HTML does not support this functionality. It will require a programming to display this.

6. Legal Notice

This program and coding information is intellectual property owned by Trine Commerce Systems, Inc., Austin, Texas. No part of the coding or program may be used without authorization.