

# Gateway Link Requirements

Trine Commerce Systems, Inc.  
MMS-Trine, LP

## Credit Card Processing

The following information is provided to clients of Trine Commerce Systems, Inc. and MMS-Trine, LP 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	1
2.	Swipe/Magnetic Card Requirements	2
3.	Fields Available But Not Required	2
4.	XML Processing	3
	4.a. Processing VOID Transactions	5
	4.b. Special Case: Fee Information Request	5
5.	HTML Responses From The Network	6
6.	Legal Notice	7

## 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](http://www.TrineCS.com/docs))

### 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.**

---

<b>card_number</b>	The credit card number being processed.
<b>card_date</b>	Expiration date on the card in the format of MM/YY
<b>card_name</b>	Name of the cardholder.
<b>amount</b>	Total amount of the transaction
<b>transaction_type</b>	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:

<b>hash</b>	Hidden 32 character identification string assigned by the server.
<b>mag_data</b>	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 <b>mag_data</b> <i>without the initial % or closing ?</i> or B) send two separate fields: <b>mag_track1</b> with track one data in it, and <b>mag_track2</b> with track two data in it. For many purposes, track 1 is optional.
<b>amount</b>	Total amount of the transaction
<b>transaction_type</b>	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

## 3. Fields Available But Not Required

<b>cvv2</b>	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.
<b>name</b>	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.
<b>merchant_1</b> <b>merchant_2</b>	User-configurable fields. These fields may be visible or hidden. They do appear in email confirmations. 30 character limit each.

<b>recurring</b>	Recurring transaction flag: daily, weekly, weekend, monthly (every 30 days), monthend, quarterly, semiend, yearend.
<b>cycles</b>	Number of recurring cycles, -20 for perpetual

### Other Billing/Information Fields:

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

#### Shipping Information

<b>shipping_name</b>
<b>shipping_company</b>
<b>shipping_address1</b>
<b>shipping_address2</b>
<b>shipping_city</b>
<b>shipping_state</b>
<b>shipping_country</b>
<b>shipping_zip</b>

#### Order Information

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

## 4. XML Processing

<b>Gateway URL</b>	<a href="https://www.trinecs.net/processor/transact/xml">https://www.trinecs.net/processor/transact/xml</a> Transaction types supported are: SALE, CREDIT, PREAUTH, FORCE and VOID
--------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Example XML Request (with example text):

```
<request>
<hash>1a3255abdbe5b764d66fef79eb85d4dc</hash>
<transaction_type>SALE</transaction_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/void/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>

## 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.