



TRUSTCOMMERCE
COMPREHENSIVE | SECURE | TRUSTED

TC Link API Developer Guide

Version 5.1.0



Copyright © 2013 TrustCommerce. All rights reserved.

The information contained in this document belongs to TrustCommerce. It may not be used, reproduced or disclosed without the written approval of TrustCommerce. While every effort has been made to ensure the accuracy of the information contained in the product guide, the information is supplied without representation or warranty of any kind, is subject to change without notice and does not represent a commitment on the part of TrustCommerce. TrustCommerce, therefore, assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from this material or any part thereof, or any supplementary materials subsequently issued by TrustCommerce.

TrustCommerce Corporate Headquarters
9850 Irvine Center Drive
Irvine, CA 92618
USA

Toll Free: (800) 915-1680
Phone: (949) 387-3747
Fax: (949) 266-0359

www.trustcommerce.com

TC Link API and TC Vault are trademarks and TC Citadel, TC CrediGuard, and TC Trustee are registered trademarks of TrustCommerce.

All other trademarks are the property of their respective companies.

Contents

About TC Link	7
What is TC Link?	7
How TC Link Works	7
Intended Audience for This Guide	8
Where to Go for More Information	8
TC Link Functions	9
Overview	9
TC Link Data	10
Overview	10
Rules	10
Supported Payment Types	10
Overview	10
Supported Industry Segments	11
Overview	11
Getting Started with TC Link	12
Overview	12
Supported Platforms	12
Installing TC Link	12
Network Configuration	12
Testing Your Connection	14
Overview	14
Your First Transaction	16
Overview	16
Basic Transaction Processing	18
Overview	18
Transaction Requests	18
Transaction Responses	19
Testing Your Application	20
Test Data	20
Test Cards - Approved	21
Test Cards - Declined	22
ACH Test Numbers	22
Sale (Authorization and Host Capture)	23
Overview	23
Preauth (Authorization Only)	26
Overview	26
Postauth (Capture Only)	29
Overview	29



Offline Sale (Offline Authorization)	31
Overview	31
Verify (Card Verification)	33
Overview	33
Credit (Refund)	35
Overview	35
Credit2 (Standalone Refund)	37
Overview	37
Void (Void Transaction)	39
Overview	39
Reversal (Reversal Transaction)	41
Overview	41
Chargeback	43
Overview	43
About Transaction Responses	45
Overview	45
Transaction Status	46
Overview	46
Decline Type	47
Declinetype Fields	47
Error Response	48
Error Fields	48
Error Type	49
Errortype Fields	49
Response Field Groups	50
Overview	50
Platform	50
Card Present (CP)	53
Card Swipe Processing	53
Encrypted Card Swipe Processing	54
Card Not Present (CNP)	56
E-Commerce or Mail Order/Telephone Order	56
Debit	58
Overview	58
Signature Debit	58
PIN-based Debit	58
PIN-less Debit	61
Overview	61
Electronic Checks/ACH	63
Overview	63



Purchase Card	65
Overview	65
Purchase Level II	65
Purchase Level III	68
Inventory Information Approval System (IIAS)	73
Overview	73
IIAS Transaction Fields	73
Healthcare Eligible and Prescription Amount Fields	74
IIAS Sale Transaction Examples	74
IIAS Reversal Transaction Example	76
IIAS Partial Auth Response Field	77
Recurring/Installment Payments (Non TC Citadel)	78
Overview	78
Restaurant (Tip)	80
Overview	80
TC Citadel Fields	81
Overview	81
Creating a Billing ID	82
Overview	82
Verifying a Billing ID	84
Overview	84
Unstoring a Billing ID	85
Overview	85
Updating a Billing ID	86
Overview	86
Continuity Payments: Recurring, Installment, and Deferred	87
Overview	87
One-Time Subscription Transactions	89
Overview	89
One-Time Payment	91
Overview	91
Fraud Score	92
Overview	92
Blacklists	93
Overview	93
Velocity	94
Overview	94
Address Verification System (AVS)	96
Overview	96
Card Security Code (CSC)	98
Overview	98



Discretionary Fields.....	100
Ticket and Operator Fields.....	100
TC Transaction Custom Fields	100
TC Citadel Custom Fields.....	101
Query API	103
Overview.....	103
Best Practice.....	103
HTTPS/POST Query Types.....	104
HTTPS/POST Query Fields.....	105
TC Vault Example	107
TC Vault Query API Transaction ID Example	111
TC Batch.....	112
Overview.....	112
HTTPS/POST	118
Overview.....	118
TC Trustee API	119
Overview.....	119
Externally Integrated Applications.....	119
Overview.....	119
Appendix A — Fields List	120
Appendix B — AVS Response Codes.....	124
Address Verification System (AVS) Response Codes	124



About TC Link

What is TC Link?

The TC Link API™ (TC Link) is the fundamental interface for all payment processing with TrustCommerce. TC Link is a thin, open-source, messaging API that facilitates processing of a wide variety of payment types on a wide variety of platforms and languages. With TC Link, you get the following:

- Comprehensive dependability, security, and payment support
- Ease of integration with a full-featured test platform
- Integration, account, and implementation support
- High availability with client and server side fail-over and load-balancing
- Platform independence (Microsoft Windows, UNIX and derivatives, Linux and derivatives)
- Language/framework independence (C/C++, Java, COM/.NET, Perl, Python, and PHP)
- Code samples and test scripts

How TC Link Works

Specifics may vary on different languages and platforms; however, all TC Link implementations provide equal functionality to:

1. Initiate a secure connection.
2. Securely collect transaction details.
3. Securely transmit transaction details.
4. Securely receive a response.
5. Parse the response for easy handling.
6. Close the connection.

Higher languages, such as Java, Perl, Python, and PHP, provide the `Send()` method that combines steps 3 through 6 for easier integrations. As such, `Send()` accepts an associative array of input fields and returns an associative array of output fields.

Other implementations, such as C/C++, COM/.NET and CFX provide the `Create()`, `PushParam()`, `Send()`, and `GetResponse()` methods to accomplish the same.



Important: `Send()` is the only method which transmits and receives data from TrustCommerce servers. Generally, this method returns in no more than two seconds.



FYI: TrustCommerce recommends that Windows developers use either the COM or .NET implementation, as they facilitate many other languages such as ASP Classic, ASP.NET, CFX, Perl, Python, and PHP.

Intended Audience for This Guide

This guide assumes that you:

- Have a foundation as a software developer (standalone, Web, or otherwise)
- Have a basic understanding of electronic payment processing



Important: Developers should fully review the contents of this guide before attempting to install or implement TC Link or any other payment-related functionality. Failure to do so may result in a compromised implementation.

Where to Go for More Information

This Developer Guide provides a platform and language independent primer for integrating TrustCommerce payment processing and reconciliation into payment applications. Additional documentation is available in the TC Vault Downloads page. Other documents of interest are:

- TC Vault™ User Guide
- TC Trustee® API Developer Guide
- TC Link platform/language-specific README files



Important: Unless otherwise indicated, the examples used throughout this guide are presented in C/C++. The CustIDs, passwords, and cardholder primary account numbers (PANs) used throughout this guide are for example purposes only. Do not attempt to use them. The README files have language-specific information and should always be used as a first step in integration.



TC Link Functions

Overview

The following table summarizes the primary methods found in the C/C++ TC Link implementation. Subsequent sections define the specifics for the various transaction types.

TC Link Functions	
Function Name	Description
CreateLink()	Create a new transaction instance
PushParam()	Set a single input or request field: "custid", "action", "amount", and so on.
Send()	Process a transaction
GetResponse()	Retrieve a single output or response field
GetEntireResponse()	Retrieve all output or response fields
Destroy()	Perform garbage collection: close handles, free allocated memory, and so on.
GetVersion()	Get the TC Link version



Important: TC Link may vary from version to version. Consult the README file in the TC Link package you chose for your environment. That README file describes the specifics for that language/platform.

TC Link Data

Overview

TC Link processes inputs and outputs as key- or field-value pairs. Field ordering is arbitrary; input fields can be submitted in any order, and output fields should not be expected in any order.

Rules

TC Link requires that all data belong to a subset of the ASCII character set and:

- Non-printable characters, such as CR, LF, and other control characters, must not be submitted.
- The pipe ("|") character must not be submitted.
- Spaces at the beginning and end of fields are not preserved; spaces in the middle are preserved.
- Dates must be submitted as "YYYY-MM-DD" where "YYYY" is the four-digit year, "MM" is the two-digit month, and "DD" is the two-digit day. For example, February 1, 2015 is submitted as "2015-02-01".
- The payment card expiration date (exp), must be submitted as "MMYY", where "MM" is the two-digit month and "YY" is the last two digits of the year. For example, February 2015 is submitted as "0215".

Supported Payment Types

Overview

TrustCommerce is certified for processing with numerous major front-end processors, for all major payment types and all major card types. The following payment types are supported:

Supported Payment Types	
Payment Type	Description
Credit Cards	Visa, MasterCard, American Express, Discover, Diners Club, JCB
Debit Cards	PIN-based debit, signature debit, PIN-less debit
Purchase Cards	Purchase Level II and Level III
Checking/Savings	Electronic checks/Automated Clearing House (ACH)



Supported Industry Segments

Overview

The following industry segments are supported:

Supported Industry Segments	
Situation	Industry
Card Present (CP)	Retail and restaurant
Card Not Present (CNP)	Direct marketing, mail order/telephone order (MOTO), e-commerce (shopping carts, donation sites, etc.) and continuity payments (recurring, installment and deferred)



Important: Supported payment types and industry segments vary from processor to processor. Consult a TrustCommerce Solutions Consultant to discuss your specific needs: (800) 915-1680, Option 5.

Getting Started with TC Link

Overview

To get started with TC Link, first obtain the appropriate TC Link from the TC Vault Downloads page. A separate TrustCommerce login is required to login to the TC Vault and to begin testing your TC Link integration. If you do not have a TrustCommerce account, you may sign up for free trial from TrustCommerce Free Trial. Alternatively, call (800) 915-1680, Option 5 and speak with a Solutions Consultant.

Supported Platforms

TC Link is available for the following languages/platforms:

- C/C++
- Java
- COM/.NET
- Perl
- Python
- PHP
- CFX
- Ruby

Installing TC Link

TC Link installation instructions, as well as other language/platform specifics, are included in the README file contained within each TC Link package.



Important: If you cannot install TC Link in your environment (for example, if your application runs on a shared server), see Alternative Processing Interfaces.

Network Configuration

Your network must be configured as follows:

- DNS is required and must not be restricted, must honor TTLs, and must be otherwise RFC-compliant.
- ACLs must not restrict outbound traffic to, or related inbound traffic from any IP addresses in the TrustCommerce IP blocks.

TrustCommerce may at any time, add, change, or remove IP addresses within the published IP blocks. Please notify your Account Manager or a Client Services Representative if you want to be informed of these changes. TrustCommerce will provide 30 days advance notice before adding or removing blocks.



TrustCommerce currently uses the following IP blocks:

Network Configuration - IP Blocks	
CIDR Notation	Netmask
207.38.18.32/27	207.38.18.32/255.255.255.224
207.38.46.0/26	207.38.46.0/255.255.255.192
207.38.125.80/28	207.38.125.80/255.255.255.240
208.42.227.128/26	208.42.227.128/255.255.255.192



Important: TrustCommerce strongly recommends against the hard coding of IP addresses in firewalls, software, or other configurations as it may result in connectivity interruptions. The use of protocol filters and helpers, such as application-layer firewalls, is not recommended.



Testing Your Connection

Overview

Telnet can be used to test your environments ability to connect to TrustCommerce. Generally, TC Link connects to TrustCommerce via TCP to pgw1.trustcommerce.com, port 443. If you are able to connect successfully, then TC Link should have no problem connecting from the same computer.



Note: If you are unable to establish a connection, contact TrustCommerce Client Services at (800) 915-1680, Option 3.

```
Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'

Microsoft Telnet> status
Connected to pgw1.trustcommerce.com
Microsoft Telnet> quit
```

Windows XP

1. Click **Start**, **Run** and type telnet pgw1.trustcommerce.com 443.
2. Press **Ctrl** and **]** (simultaneously).
3. Type status at the prompt. The message "Connected to pgw1.trustcommerce.com" indicates a successful connection.
4. Type quit, and then press **Enter** to exit Telnet.

Windows Vista or Windows 7

1. Click **Start**, type cmd, and then press **Enter**. A **cmd.exe** window appears.
2. Type telnet pgw1.trustcommerce.com 443.
3. Press **Ctrl** and **]** (simultaneously).
4. Type status at the prompt. The message "Connected to pgw1.trustcommerce.com" indicates a successful connection.
5. Type quit then press **Enter** to exit Telnet.



UNIX/Linux

1. Open a console/shell, run `telnet pgw1.trustcommerce.com 443`. The message "Connected to pgw1.trustcommerce.com" indicates a successful connection.
2. Press **Ctrl** and **]** (simultaneously).
3. Type `quit` to exit Telnet.

```
you@yourhost:~$ telnet pgw1.trustcommerce.com 443
Trying 207.38.46.26...
Connected to pgw1.trustcommerce.com.
Escape character is '^]'.
^]

telnet> quit
Connection closed.
```



Your First Transaction

Overview

At this point, the correct TC Link package for your environment is downloaded and installed, the network is configured properly and you are ready to run your first test transaction. Keep the following tips in mind:

- Follow the README instructions and examples in your chosen package.
- Test the payment and transaction types you intend to use in your business model.
- You can use the test examples provided here and in the TC Link package, or build your own.
- Run test (demo=y) transaction(s) until you are satisfied with the results.
- Before you go live, run production test transactions to verify your account is set up properly with your processing platform.



Note: Contact Client Services with any questions: (800) 915-1680, Option 3.

Sample TC Link (Java)

```
tctest.java:

import java.security.GeneralSecurityException;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

import com.trustcommerce.TCLink;

public class tctest {
    public static void main( String[] argv ) {
        try { TCLink.init(); } catch (
GeneralSecurityException e ) { return; }
        TCLink tc = new TCLink();
        Map in = new HashMap();
        in.put("custid", "962100");
        in.put("password", "XXXXXXXX");
        in.put("action", "sale");
        in.put("amount", "100");
        in.put("cc", "4111111111111012");
        in.put("exp", "1012");
```




```
Map out = tc.submit(in);

Iterator it = out.keySet().iterator();
while ( it.hasNext() )
{
    Object key = it.next();
    System.out.println( key + "=" + out.get(key) );
}
}
```

Sample Compile and Response

```
# cd /usr/src
# unzip tclink-3.4.4-java.zip
# export CLASSPATH=$CLASSPATH:/usr/src/tclink-3.4.4-
java/trustcommerce.jar
# javac tctest.java
# java tctest
avs=0
authcode=1A64LQ
status=approved
transid=028-0029514119
#
```

Basic Transaction Processing

Overview

This section discusses the specific details of the basic transaction types supported in TC Link. Information in this section is language/platform independent. Please refer to the specific README file from your chosen TC Link language/platform as an early step in your integration.



Important: Unless otherwise indicated, the examples used throughout this guide are presented in C/C++. The CustIDs, passwords, cardholders primary and account numbers used throughout this guide are for example purposes only. Do not attempt to use them.

Transaction Requests

TC Link can be used to process any supported payment type in any supported industry segment. As such, TC Link accepts a wide variety of input parameters from which a transaction can be built. Keep the following in mind:

- Few input parameters are mandatory: such as CustID, password and action.
- Many input parameters are optional and are used either to qualify for preferred rates and/or for advanced reporting.
- Most transactions can be sent with only a few parameters.
- Some input parameters are specific to one transaction type.
- Input parameters are only minimally validated by TC Link.
- Most declines, rejections and processing errors are detailed within the response.



Note: TrustCommerce authenticates incoming TC Link requests using your TrustCommerce-issued CustID and password combination. Whereas your TC Vault and TC Link CustID will be the same, the passwords may differ. Unless otherwise specified, this document refers to your TC Link password.

Required Fields

The following fields are required for all transactions:

Required Fields for all Transactions	
Field Name	Description
custid	Your TrustCommerce-assigned, numeric CustID, for example, "962100"
password	The TC Link password associated with the aforementioned CustID
action	"sale", "preauth", "postauth", "credit", "void", "store", and so on.

Transaction Responses

Any transaction sent will return several parameters describing the success or failure of the transaction. Additional fields may be returned based on the type of action, and are described in more detail along with the examples throughout this document, but "status" and "transid" are returned for every transaction.

TransID

Properly formatted transactions will always return a transaction ID (TransID). The transaction ID is the unique identifier for this transaction, and can be used to retrieve the transaction from the TC Vault Web site, or otherwise access the transaction in the future. For example, in order to credit a previous transaction, you will need to send the "transid" of the original transaction.

Status

Transactions always return a status parameter, which describes the success or failure of the transaction. Status can be set to one of the following:

Required Fields for all Transactions	
Status Value	Description
approved	The transaction was successfully authorized.
accepted	The transaction has been successfully accepted into the system.
decline	The transaction was declined; see Decline Type for further details.
baddata	Invalid fields were passed; see Error Type for further details.
error	System Error when processing the transaction; see Error Type for further details



Note: For a comprehensive list, see [Transaction Responses](#).



Testing Your Application

The TrustCommerce TC Link test environment is fully functional and available 24/7. Your TC Link test account in our demo environment is available for as long as you want to use it. Many of our integrators continue to use the test account for future development projects, evaluation of new products, or for training new staff.

- If your account is in production (live), you can indicate whether the transaction is live by setting the "demo" field to "n"; or if the transaction is a demo transaction by setting the field to "y".
- By default, configured test accounts automatically send test transactions as (demo=y). As a result, you are not required to pass this field.

TC Link Test Environment Field	
Field Name	Description
demo	"y" or "n". Used to indicate the transaction request is a test or demo transaction regardless of account status.

The following example shows how this field can be used:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "store");  
TCLinkPushParam(handle, "demo", "y");  
TCLinkPushParam(handle, "cc", "4111111111111111");  
TCLinkPushParam(handle, "exp", "1215");  
TCLinkPushParam(handle, "name", "Test Person");  
TCLinkPushParam(handle, "amount", "455");
```

Test Data

While testing, you may want to experiment with the different responses that can be generated. The following table provides test card numbers will produce an approval and includes address data for testing the address verification system (AVS). If you want to test CVV, the code listed in the right-hand column is the correct CVV code. Other valid payment cards will work, but will produce a "U" AVS code.



Important: Test cards only work on transactions flagged as demo, or on accounts in test mode. If run as live transactions, they will all return a decline with a "declinetype" of "carderror".



Test Cards - Approved

Credit Card Approved Test Data				
Card Type	Card Number	Exp Date	Address1, City, State, and Zip	CVV
Visa	4111111111111111	04/19	123 Test St., Somewhere, CA, 90001	123
MasterCard	5411111111111115	04/19	4000 Main St., Anytown, AZ, 85001	777
American Express	3411111111111111	04/19	12 Colorado Blvd., Elsewhere, IL, 40000	4000
Discover	6011111111111117	04/19	6789 Green Ave., Nowhere, MA, 12345	N/A
Diners Club	364844444444446	04/19	7390 Del Mar Blvd, Anytown, NY, 01101	N/A
JCB	213122222222221	04/19	350 Madison Ave., Springfield, OH, 40000	N/A



Note: The test card numbers from the preceding table are duplicated in the following table **without spaces** for use with copy and paste commands.

```
Visa: 4111111111111111
MasterCard: 5411111111111115
American Express: 3411111111111111
Discover: 6011111111111117
Diner's Club: 364844444444446
JCB: 213122222222221
```

Test Cards - Declined

The following card numbers will generate a decline, with the “declinetype” as listed below. You may use this to test code which takes different paths dependent upon the type of decline.

Test Cards - Declined		
Card Number	Exp Date	Decline Type
4012 3456 7890 9	04/19	decline
5555 4444 3333 2226	04/19	call
4444 1111 4444 1111	04/19	carderror



Note: The test card numbers from the preceding table are duplicated in the following table **without spaces** for use with copy and paste commands.

```
decline: 4012345678909
call: 5555444433332226
carderror: 4444111144441111
```

ACH Test Numbers

ACH Test Numbers	
Routing Number	Account Number
789456124	55544433221

Sale (Authorization and Host Capture)

Overview

A **sale** transaction is the simplest method to process a complete payment. Upon submission, a sale transaction attempts an authorization and if approved, schedules capture, which initiates the process that results in funding.



Important: Only one payment type can be processed at a time.

The following basic payment types are supported:

- Card Present (CP) payments with "track1" and "track2"
- Encrypted Card Present payments with "encryptedtrack"
- Card Not Present (CNP) payments with "cc" and "exp"
- Checking/Savings payments with "routing" and "account"

Sale – Required Fields

Sale - Required Fields	
Field Name	Description
action	"sale"
cc	Cardholder primary account number (PAN) (for example, a credit or debit card number)
exp	Payment card expiration date in "MMYY" format
amount	Transaction amount in cents (for example, \$1.00 would be submitted as "100")

Sale – Suggested and/or Other Notable Optional Fields

Sale - Suggested and/or Other Notable Optional Fields	
Field Name	Description
address1 and zip	Address verification system fields; see Address Verification System (AVS) .
cvv	Card security code; see Card Security Code (CSC) .
ticket	This discretionary field can be used as a merchant-specified transaction identifier for searching and reporting. In some cases, the value of this field is passed through to the financial institutions.



Sale - Suggested and/or Other Notable Optional Fields	
Field Name	Description
store	"y" or "n." This function pushes credit card data to TC Citadel and returns a billingID. This enables merchants to charge the cardholder and keep a record on file for use at a later date. The store function only pushes data to citadel if the sale/preauth is approved. The BillingID is returned in the sale/preauth transaction response. The sale/preauth is associated with the BillingID record.

Sale – Common Responses

The following table demonstrates the common fields and values returned for a sale transaction request; for a comprehensive list, refer to [Transaction Responses](#).

Sale - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: approved - The transaction was successfully authorized. declined - The transaction was declined; see Decline Type for further details. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for further details.
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request; see Address Verification System (AVS) Response Codes for additional information.



Sale Transaction Example

The following example demonstrates a sale that attempts to charge \$1.25 against a payment card 4111111111111111 expiring in December of 2015:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "sale");  
TCLinkPushParam(handle, "amount", "125");  
TCLinkPushParam(handle, "cc", "4111111111111111");  
TCLinkPushParam(handle, "exp", "1215");
```

Preauth (Authorization Only)

Overview

A "preauth" is used to:

- Verify that sufficient funds are available from a valid payment card.
- Place a temporary hold on requested funds.

When the transaction can be finalized, a "postauth" is used. The authorization hold will remain in effect until either the transaction is funded (following a successful postauth) or the authorization is expired/canceled by the card issuer. Preauth transactions are often used when:

- The merchant cannot fulfill the purchase immediately (for example, if an out-of-stock item must be ordered or manufactured).
- The merchant does not yet know a final total (for example, pre-tip restaurant transactions, shipping costs, and manufacturing costs).

Preauth - Required Fields

Preauth - Required Fields	
Field Name	Description
action	"preauth"
cc	Cardholder primary account number (PAN) (e.g., credit or debit card number)
exp	Payment card expiration date in "MMYY" format
amount	Transaction amount in cents (e.g., \$1.00 would be submitted as "100")



Note: "preauth" and other authorization-only transactions must be captured ("postauth"-ed), reversed, or reauthorized within 72 hours for Card Not Present (CNP) processing, and within 24 hours for Card Present (CP) processing. Failure to comply may result in transaction downgrades and additional processing fees and penalties.



Preauth - Suggested and/or Other Notable Optional Fields

Preauth - Suggested and/or Other Notable Optional Fields	
Field Name	Description
address1 and zip	Address verification system fields (see Address Verification System (AVS))
cvv	Card security code (see Card Security Code (CSC))
ticket	This discretionary field can be used as a merchant-specified transaction identifier for searching and reporting. In some cases, the value of this field is passed through to the financial institutions.
store	"y" or "n." This function pushes credit card data to TC Citadel and returns a billingID. This enables merchants to charge the cardholder and keep a record on file for use at a later date. The store function only pushes data to citadel if the sale/preauth is approved. The BillingID is returned in the sale/preauth transaction response. The sale/preauth is associated with the BillingID record.

Preauth - Common Responses

The following table demonstrates the common fields and values returned for a preauth transaction request; for a comprehensive list, refer to [Transaction Responses](#).

Preauth - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: approved - The transaction was successfully authorized. declined - The transaction was declined; see Decline Type for further details. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for further details.
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request; see Address Verification System (AVS) Response Codes for additional information.



Preauth Transaction Example

The following example demonstrates a "preauth" that requests \$1.25 be reserved from a payment card 4111111111111111 expiring in December of 2015:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "preauth");  
TCLinkPushParam(handle, "cc", "4111111111111111");  
TCLinkPushParam(handle, "exp", "1215");  
TCLinkPushParam(handle, "amount", "125");
```

Postauth (Capture Only)

Overview

A "postauth" transaction is used to finalize a preauth, thereby initiating funding. In most cases, postauth transactions should be submitted for the same amount as the preauth. If the postauth amount differs from the preauth amount, a reversal should be issued and a new authorization obtained; see [Reversal \(Reversal Transaction\)](#).



Important: TrustCommerce allows you to postauth up to 14 days after a preauth.

Postauth - Required Fields

If an amount is not specified, the "postauth" will automatically be submitted in the same amount as the original "preauth". The following are the minimum required field value pairs for a postauth transaction:

Postauth - Required Fields	
Field Name	Description
action	"postauth"
transid	"transid" returned from the "preauth". "transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"

Postauth - Suggested and/or Other Notable Optional Fields

Postauth- Suggested and/or Other Notable Optional Fields	
Field Name	Description
amount	Transaction amount in cents (\$1.00 is submitted as "100"). In general, the postauth amount should match preauth amount.
ticket	This is a discretionary field that can be used as a merchant-specified transaction identifier for searching and reporting. In some cases, the value of this field is passed through to the financial institutions.



Postauth - Common Responses

The following table demonstrates the common fields and values returned for a postauth transaction request. For a comprehensive list, refer to [Transaction Responses](#).

Postauth - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: accepted - The transaction was successfully accepted into the system. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for further details.

Postauth Transaction Example

The following example demonstrates a "postauth" that attempts to capture \$3.66 based on a "preauth" with "transid" 012-1234567890:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "postauth");  
TCLinkPushParam(handle, "transid", "012-1234567890");  
TCLinkPushParam(handle, "amount", "366");
```

Offline Sale (Offline Authorization)

Overview

An "offline sale" transaction is used to initiate capture of an authorization outside of the TrustCommerce systems. Offline sales are generally used when an attempt to process a payment card transaction automatically is unsuccessful, for whatever reason (for example, a decline type of "call", unreadable track data). On these occasions, the operator may call the customer service number listed on the card to request an "offlineauthcode". The card issuer may provide a six-digit authorization code to enable offline processing.



FYI: If the "offlineauthcode" provided is more than six digits, the operator should ask for the six-digit code.

Offline Sale - Required Fields

Offline Sale - Required Fields	
Field Name	Description
offlineauthcode	Six-digit numeric code used to force the transaction
action	"preauth"
cc	Cardholder primary account number (PAN) (for example, a credit or debit card number)
exp	Payment card expiration date in "MMYY" format
amount	Transaction amount in cents (\$1.00 would be submitted as "100")

Offline Sale - Suggested and/or Other Notable Optional Fields

Postauth- Suggested and/or Other Notable Optional Fields	
Field Name	Description
ticket	This is a discretionary field that can be used as a merchant-specified transaction identifier for searching and reporting. In some cases, the value of this field is passed through to the financial institutions.



Offline Sale - Common Responses

The following table demonstrates the common fields and values returned for an offline authorization transaction request; for a comprehensive list, refer to [Transaction Responses](#).

Preauth - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>

Offline Authorization Example

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "5000");
TCLinkPushParam(handle, "offlineauthcode", "A12345");
```


Verify (Card Verification)

Overview

A "verify" transaction verifies that payment card (credit or debit) information is valid. Generally, verify is used to validate cardholder information before a merchant commits their inventory or resources to a product or service delivery. Additionally, verify is used by merchants when payment information on-file changes.



Important: Verify may apply to the cardholder primary account number (credit or debit card), expiration date, card security code and billing address. Verify does not check for available funds.

Verify - Required Fields

Verify - Required Fields	
Field Name	Description
action	"verify"
cc	Cardholder primary account number (PAN) (for example, a credit or debit card number)
exp	Payment card expiration date in "MMYY" format

Verify - Suggested and/or Other Notable Optional Fields

Verify - Suggested and/or Other Notable Optional Fields	
Field Name	Description
ticket	This is a discretionary field that can be used as a merchant-specified transaction identifier for searching and reporting. In some cases, the value of this field is passed through to the financial institutions.
address1 and zip	Address verification system fields; see Address Verification System (AVS) .
cvv	Card security code; see Card Security Code (CSC) .



Verify - Common Responses

The following table demonstrates the common fields and values returned for an offline authorization transaction request; for a comprehensive list, refer to [Transaction Responses](#).

Preauth - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: approved - The transaction was successfully authorized. declined - The transaction was declined; see Decline Type for further details. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for further details.
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request; see Appendix B - Address Verification System (AVS) Response Codes .

Verify Example

The following example demonstrates a verify transaction that attempts to validate the payment card "4111111111111111" expiring in December of 2015.

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "verify");  
TCLinkPushParam(handle, "cc", "4111111111111111");  
TCLinkPushParam(handle, "exp", "1215");
```



Credit (Refund)

Overview

A "credit" is a transaction request that reimburses funds for a previous transaction from the merchant's bank account back to the cardholder's payment card account. Credits are generally performed against a specific transaction ID. To perform a standalone refund (not in reference to a prior TrustCommerce transaction), refer to [Credit2 \(Standalone Refund\)](#).

Credit - Required Fields

Credit - Required Fields	
Field Name	Description
action	"credit"
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"

Credit - Suggested and/or Other Notable Fields

Credit - Suggested and/or Other Notable Fields	
Field Name	Description
amount	Transaction amount in cents (\$1.00 is submitted as "100")

Credit - Amount Field

The credit amount field is optional. If it is not included, the full amount from the previous transaction will be credited. If it is included, it must be some value greater than, equal to \$0.01 and less than, or equal to the amount collected on the previous transaction, and minus any credits that have already been run; otherwise the transaction will not be credited.

- As an example, the full amount of the previous transaction is \$25.12. A \$2.00 credit was already run, so the amount you want to credit is \$23.12. That amount would be submitted as "2312".
- If the amount is less than \$1.00, then the amount must be prepended with zeros until the length has three or more digits. For example, \$.75 cents would be submitted as "075".



Credit - Common Responses

The following table demonstrates the common fields and values returned for a credit transaction request. For a comprehensive list, refer to Transaction Responses.

Credit - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>accepted—The transaction has been successfully accepted into the system.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>

Credit Example

The following is an example of a "credit" (refund) transaction in the amount of \$3.66.

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "credit");  
TCLinkPushParam(handle, "transid", "012-1234567890");  
TCLinkPushParam(handle, "amount", "366");
```

Credit2 (Standalone Refund)

Overview

Another type of credit transaction is known as "credit2" or "solocredit". This is designed for new merchants who are navigating from one solution to another, and may need to issue credits for transactions that did not originate in the TrustCommerce system. A credit2 is a one-time transaction.



Important: TrustCommerce strongly recommends payment card number and cardholder's identity are verified prior to running a credit2 transaction. Consult a Solutions Consultant to discuss your specific needs: (800) 915-1680, Option 5.

Credit2 – Required Fields

Credit2 – Required Fields	
Field Name	Description
cc	Cardholder primary account number (PAN) (for example, a credit or debit card number).
exp	Payment card expiration date in "MMYY" format.
amount	Transaction amount to credit back in cents (\$1.00 is submitted as "100").

Credit2 – Suggested and/or Other Notable Fields

Credit2 – Suggested and/or Other Notable Fields	
Field Name	Description
ticket	This is a discretionary field that can be used as a merchant-specified transaction identifier for searching and reporting. In some cases, the value of this field is passed through to the financial institutions.



Credit2 – Common Responses

The following table demonstrates the common fields and values returned for a credit2 transaction request. For a comprehensive list, refer to Transaction Responses.

Credit2 – Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>accepted–The transaction has been successfully accepted into the system.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>

Credit2 Example

The following is an example of a "credit2" (Standalone Refund) where the payment card number and expiration date are provided.

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "credit2");  
TCLinkPushParam(handle, "cc", "4111111111111111");  
TCLinkPushParam(handle, "exp", "0412");  
TCLinkPushParam(handle, "amount", "500");
```



Void (Void Transaction)

Overview

A "void" transaction is used to cancel a postauth, sale, or credit or a credit2 before it is sent for settlement.

Void - Required Fields

Void - Required Fields	
Field Name	Description
action	"void"
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"

Void - Suggested and/or Other Notable Fields

Void - Suggested and/or Other Notable Fields	
Field Name	Description
amount	Transaction amount in cents (\$1.00 is submitted as "100"). The amount field is optional for voids. The full amount from the original transaction is generally used.

Void - Common Responses

The following table demonstrates the common fields and values returned for a void transaction request. For a comprehensive list, refer to Transaction Responses.

Void - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: accepted —The transaction has been successfully accepted into the system. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for further details.



Void Transaction Example

The following illustrates an attempt to void a previously approved sale transaction. For sale transactions, a reversal request is also submitted behind the scenes to free the funds from the authorization.

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "void");  
TCLinkPushParam(handle, "transid", "012-1234567890");
```


Reversal (Reversal Transaction)

Overview

A "reversal" is a transaction that allows the operator to reverse the credit hold on a previously approved authorization prior to settlement.

- This transaction is most often used in an e-commerce or mail order/telephone order (MOTO) environment when the original authorization amount changes or the cardholder cancels their order.
- Once a reversal is processed it cannot be undone.

TrustCommerce supports partial and full reversals for Visa branded cards and full reversals for MasterCard branded cards when this feature is available. Consult a TrustCommerce Solutions Consultant to discuss your specific needs: (800) 915-1680, Option 5.



Important: TrustCommerce recommends implementing the reversal transaction type within your processing model for authorizations still open after 72 hours. Failure to reverse or capture authorizations within 72 hours may result in a transaction downgrade to a different rate category and the merchant may be subject to additional fees/penalties.



Note: The reversal time limit for Card Present (CP) environments is 24 hours. Please contact your merchant bank for full details.

Reversal – Required Fields

Reversal - Required Fields	
Field Name	Description
action	"reversal"
transid	"transid" returned from the "preauth". "transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"

Reversal - Suggested and/or Other Notable Fields

Reversal - Suggested and/or Other Notable Fields	
Field Name	Description
amount	Transaction amount in cents (e.g., \$1.00 would be submitted as "100").



Note: The amount field is optional. If not included the full amount from the previous transaction will be reversed. If included, it must be some value greater than or equal to \$0.01 and less than or equal the amount collected on the previous transaction; otherwise, the transaction will not be reversed.

Reversal – Common Responses

The following table demonstrates the common fields and values returned for a reversal transaction request. For a comprehensive list, refer to Transaction Responses.

Reversal - Common Responses	
Field Name	Description
transid	"transid" returned from the "preauth". "transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: accepted —The transaction has been successfully accepted into the system. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for further details.

Reversal Transaction Example

The basic fields for "reversal" transactions are:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "reversal");  
TCLinkPushParam(handle, "transid", "012-1234567890");  
TCLinkPushParam(handle, "amount", "500");
```

Chargeback

Overview

A chargeback occurs when the cardholder disputes an item on their payment card statement because they:

- Never authorized the charge
- Are a victim of identity theft
- Received damaged or late goods
- Did not receive the services or merchandise

If the dispute is resolved in favor of the customer, the card issuer provides a refund to the cardholder. Operators can mark a transaction that has been disputed by the cardholder as a chargeback transaction through TC Link.



Important: Within TrustCommerce, the "chargeback" action is simply for reconciliation purposes (reporting) and does not return the funds back to the cardholder.

Chargeback – Required Fields

Chargeback - Required Fields	
Field Name	Description
action	"chargeback"
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"

Chargeback – Common Responses

The following table demonstrates the common fields and values returned for a chargeback request. For a comprehensive list, refer to Transaction Responses.

Chargeback - Common Responses	
Field Name	Description
transid	"transid" returned from the "preauth". "transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>accepted–The transaction has been successfully accepted into the system.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>

Chargeback Transaction Example

The following example illustrates an attempt to perform a "chargeback" for "transid" 012-1234567890.

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "chargeback");  
TCLinkPushParam(handle, "transid", "012-1234567890");
```

About Transaction Responses

Overview

Once a transaction is submitted, results are returned using an array of field/value pairs very similar to the transaction input (see [TC Link Functions](#) and [TC Link Data](#)).

The method used to retrieve these response fields depends on the Integration method used. Please see the examples for detail on how to retrieve the response values.

General Fields

The following response fields are common to any transaction.

General Response Fields	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Specific values that can be returned are documented in a later table.
declinetype	If the response field status is returned as "decline", the "declinetype" field will also be included, providing more detail regarding the reason for the decline.
error	If the response field status is returned as "baddata", the error field will also be included, providing more detail regarding the reason for the response.
offenders	If the error response field is included, the "offenders" response field will also be included. This will indicate the field or fields that are responsible for the error.
errortype	If the response field status is returned as "error", the "errortype" field will also be included, providing more detail regarding the reason for the error.
authcode	This is the numeric or alphanumeric code received from the processor in response to a transaction.

Transaction Status

Overview

Transactions always return a status field, which describes the success or failure of the transaction. Properly formatted transactions always return a transaction ID. TrustCommerce uses a 14-digit alphanumeric value formatted like this example:

```
"transid", "012-1234567890"
```

The transid is the unique identifier for each transaction. It can be used to retrieve a specific transaction from the TC Vault Web site, or access the transaction in the future. For example, to credit a previous transaction, you need to send the original transid.

Returned transaction statuses are one of the following:

Returned Transaction Statuses	
Status Value	Description
approved	The transaction was successfully authorized.
accepted	The transaction has been successfully accepted into the system.
decline	The transaction was declined; see Decline Type .
baddata	Invalid fields were passed; see Error Type .
error	System error when processing the transaction; see Error Type .

Approved or Accepted

The difference between accepted and approved is subtle, but important.

- An accepted response only means that the transaction was queued into the TrustCommerce system without errors, but may be rejected later.
- An approved response means that the transaction was an authorization of some sort, and has successfully cleared with the bank.

An example of the difference between approved and accepted is a sale versus a credit transaction.

- A sale is a real-time authorization, so it returns approved status on success.
- A credit is not real-time; there is no such thing as a credit authorization. A credit transaction is queued for processing by the bank and there is a possibility that it may be rejected later.

Decline Type

Declinetype Fields

When status is set to "decline", the "declinetype" field contains one of the following:

Declinetype Fields	
Declinetype Value	Description
decline	This is a true "decline"; it almost always is a result of insufficient funds on the card
avs	Address verification system (AVS) failed; the address entered does not match the billing address on file at the bank.
cvv	CVV failed; the number provided is not the correct verification number for the card. See Card Security Code (CSC) .
call	The card must be authorized manually over the phone. You may choose to call the customer service number listed on the card and ask for an offline "authcode", which can be passed in the "offlineauthcode" field.
expiredcard	The card has expired. Request an updated expiration date from cardholder.
carderror	Card number is invalid, which could be a typo, or a card reported stolen.
authexpired	Attempt to "postauth" an expired (more than 14 days old) "preauth".
fraud	TC CrediGuard fraud score was below requested threshold. See Fraud Score .
blacklist	TC CrediGuard blacklist value was triggered. See Blacklists .
velocity	TC CrediGuard velocity control was triggered. See Velocity .

Error Response

Error Fields

A status of "baddata" indicates that no transaction was attempted because one or more fields were invalid. In this case, the field error will indicate the problem, and the offender field lists the offending input fields.


The error field may be returned as one of the following:

Error Fields	
Error Value	Description
missingfields	One or more fields required for this transaction type were not sent.
extrafields	Fields not allowed for this transaction type were sent.
badformat	A field was improperly formatted, such as non-digit characters in a number field.
badlength	A field was longer or shorter than the server allows.
merchantcantaccept	The merchant cannot accept data passed in this field. If the offender is "cc", for example, it usually means that you tried to run a card type (such as American Express or Discover) that is not supported by your account. If it was "currency", you tried to run a currency type not supported by your account.
mismatch	Data in one of the offending fields did not crosscheck with the other offending field.

Error Type

Errortype Fields

A status of "error" indicates an error occurred while processing the transaction. If the status is error, then the "errortype" field will be returned as one of the following:

Errortype Fields	
Errortype Value	Description
cantconnect	Could not connect to TrustCommerce. Check your Internet connection to make sure it is up.
dnsvalue	<p>The TC Link software was unable to resolve DNS hostnames. Make sure you have name resolving ability on the machine.</p> <hr/> <p> Note: The "dnsfailure" response will only be given with a test transaction. Live transactions will never receive this response. It is used to identify DNS connectivity issues prior to the merchant going live.</p> <hr/>
linkfailure	The connection was established, but was severed before the transaction could complete. To avoid duplication, check the transaction independently to determine if the transaction was successfully processed.
failtoprocess	Transmission errors were encountered in TrustCommerce's connection to the payment processor.
notallowed	Certain actions are only permitted if a CustID is configured to perform that action. If a transaction is attempted that the CustID is not authorized to perform, this "errortype" will be returned.



Important: "linkfailure" and "failtoprocess" errors indicate a non-specific communication breakdown between the payment processor and TrustCommerce. Resubmission may result in a duplicate authorization reservation though it will not result in a duplicate capture. To avoid duplication, check the transaction independently to determine if the transaction was successfully processed.



Note: Other fields (such as AVS or BillingID) may be returned by the transaction depending on the action. See the sections covering the transaction type you are running for detailed information on the specialized return values.

Response Field Groups

Overview

The response codes sent to TrustCommerce from some front-end processing platforms can be passed directly to the integrator as additional response fields. This feature must be requested for the CustID and will be activated only by TrustCommerce.



FYI: Please refer to the individual processing platform messaging documentation for possible return values and meanings.

If configured to return these fields, they are as follows:

Response Field Groups	
Response Field	Description
responsecode	The response code as sent to TrustCommerce from the processing platform.
responsecodedescriptor	The textual description of what this code indicates, as documented in the processing platform's messaging documentation.

Platform

A "baddata" error may occur when a transaction fails to meet TC Link internal validation checks. This may be the result of malformed data being sent to TrustCommerce or that the functionality required for processing the network failover component correctly is missing. These errors can usually be corrected by modifying the environment or your application to send the correct data through TC Link.



Important: Try to solve these errors in your test environment prior to moving to the production environment. This helps your business minimize interruptions when processing live transactions.

Baddata

When the status value is "baddata", the following errors may be returned with "comma-delimited list of input fields" named as "offenders".

Baddata Errors	
Error	Description
missingfields	You are missing fields required for this transaction to be processed. Provide the fields as specified in offenders—the field names in the offenders value are comma delimited.
extrafields	You are providing fields that are not allowed for this transaction type. Do not provide the fields as specified in offenders.
badlength	The input field has minimum and maximum length restrictions. Review the field description and provide a corrected version that meets the field length requirements.
badformat	The field characteristics do not match to the value that you provided. Review the field description and re-submit the correct accepted format.
merchantcantaccept	Your CustID is not setup to process the given card type or currency type. Contact your Solutions Consultant to determine if you should be processing this type of transaction.
mismatch	The field provided contains conflicting data with another field. For example, if the "cc" field does not match the data provided in the "track1" data, you would encounter an error like this. Remove one of the fields and try again.

Server

Server-generated error conditions may be due to unavailability, unreliable communication to and from the card processing networks, or an unrecognized response code from the actual card processing networks.

In these cases, a transaction may have actually processed all the way to the card issuer. If this happens, have the cardholder contact their bank, and if the authorization is present have them reverse it. The following table contains details of server generated error conditions:

Server Generated Errors		
Error	Description	Offenders
linkfailure	This occurs when there are issues receiving the response data from TrustCommerce. To avoid duplication, check the transaction independently to determine if the application was successfully processed.	N/A
unsupported	The action that you provided is not supported by the processing platform.	Action
unsupported	The use of the "tip" field is not supported on this industry or platform.	Tip



Server Generated Errors		
Error	Description	Offenders
unsupported	The use of the "shipdate" field is not supported on this industry or platform.	Shipdate
failtoprocess	Transmission errors were encountered in TrustCommerce's connection to the payment processor.	N/A



Important: "linkfailure" and "failtoprocess" errors indicate a non-specific communication breakdown between the payment processor and TrustCommerce. Resubmission may result in a duplicate authorization reservation though it will not result in a duplicate capture. To avoid duplication, check the transaction independently to determine if the transaction was successfully processed.

Client

Errors may occur that indicate network connectivity issues, regardless of the transaction type submitted to TrustCommerce. Fields that are not present are indicated as N/A and will return the appropriate type as determined by the TC Link implementation. Try to check the transid field as it may allow you to differentiate between various error types. The following table describes the two errors you might see if there was trouble connecting to or receiving data from TrustCommerce:

Client Errors	
Error	Description
cantconnect	This occurs when a connection cannot be made to any transaction-processing server.
linkfailure	This occurs when there are issues receiving the response data from TrustCommerce. To avoid duplication, check the transaction independently to determine if the application was successfully processed.

Card Present (CP)

Card Swipe Processing

Track Data

Track data is the information encoded within the magnetic strip on the back of a credit card. The electronic reader/swipe device at the point-of-sale (POS) location reads this data. Retail environments with card swipe devices should pass the magnetic stripe data along with the other payment card information on a preauth or sale transaction. The two fields used to send this data are track1 and track2.



Note: The data on track2 is preferred by most platforms over the data on track1. If the contents of track2 can be read successfully, please submit that as part of the transaction. You only need to submit the track1 data if track2 cannot be read.

All data read from the tracks may be included; however, only data between the start sentinel (a percent character "%" for track1 and a semicolon ";" for track2), and the end sentinel (a question mark "?") will be used. Everything outside the start and end sentinels (such as the trailing LRC character) will be discarded.



Important: For PCI compliance and security reasons, never store or save track data to temporary files.



FYI: Track1 and track2 data are used as substitutes for the cc and exp fields.

Card Swipe Fields

Field Name	Description
track1	The "track1" data from a card that is swiped with a magnetic stripe reader. Available for merchants who conduct face-to-face transactions. It is against many card issuer regulations to store or capture this data beyond the time needed to perform the transaction— do not store this .
track2	The "track2" data from a card that is swiped with a magnetic stripe reader. Available for merchants who conduct face-to-face transactions. It is against many card issuer regulations to store or capture this data beyond the time needed to perform the transaction— do not store this .

Card Swipe – Common Responses

The following table demonstrates the common fields and values returned for a card swipe transaction request. For a comprehensive list, refer to Transaction Responses.

Card Swipe Fields - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS). See Appendix B - Address Verification System (AVS) Response Codes .

Card Swipe Transaction Example

The following is an example of a card swipe transaction with "track2" data submitted.

```

TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle,
  "track2", ";4111111111111111=07120000000000000000?");
TCLinkPushParam(handle, "amount", "1500");
  
```

Encrypted Card Swipe Processing

TrustCommerce key-injected point-of-sale (POS) swipe devices return the track data in an encrypted format for added security. TrustCommerce systems then securely decrypt and process the track data. In this scenario, instead of sending the track1 or track2 fields, use the encryptedtrack field.



Important: Not all POS swipe devices have this functionality. Consult a TrustCommerce Solutions Consultant to discuss your specific needs: (800) 915-1680, Option 5.

Encrypted Card Swipe Fields

Encrypted Card Swipe Fields	
Field Name	Description
encryptedtrack	track1 or track2 data encrypted at the swipe device

Encrypted Card Swipe - Common Responses

The following table demonstrates the common fields and values returned for an encrypted card swipe transaction request; for a comprehensive list please refer to Transaction Responses.

Encrypted Card Swipe - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: approved - The transaction was successfully authorized. declined - The transaction was declined; see Decline Type for further details. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for further details.
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS). See Appendix B - Address Verification System (AVS) Response Codes .



Encrypted Card Swipe Transaction Example

The following is an example of an encrypted card swipe transaction:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "sale");  
TCLinkPushParam(handle, "encryptedtrack", "4111111111111111");  
TCLinkPushParam(handle, "amount", "1500");
```

Card Not Present (CNP)

E-Commerce or Mail Order/Telephone Order

Merchants who process payments through the telephone or mail are considered direct marketing mail order/ telephone order (MOTO) merchants. Because these transactions are not face-to-face, acquirers often charge higher fees to mitigate the higher risk.

If you ship products to customers and use the "postauth" transaction, you can specify the "shipdate" field. By default, the "shipdate" is the day the "postauth" is initiated, as this is consistent with industry clearing requirements. The ship date is in the form "MM-DD-YYYY" where "MM" represents the two-digit month, "DD" represents the two-digit day, and "YYYY" represents the four-digit year.

E-Commerce and MOTO Fields

Merchants are set up as direct marketing by default, which encompasses MOTO, and e-commerce. In this setup, they process within the application as if the transactions are e-commerce. If a merchant wants to process the transactions as occurring via telephone or mail order, they would submit "type=moto" for a "preauth", "postauth", or "sale" transaction.

E-Commerce and MOTO Fields	
Field Name	Description
shipdate	The date the products are shipped.
type	"moto"

E-Commerce and MOTO – Common Responses

The following table demonstrates the common fields and values returned for an encrypted card swipe transaction request. For a comprehensive list, refer to Transaction Responses.

E-Commerce and MOTO – Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>

E-Commerce and MOTO Example

The following is an example of a MOTO transaction:

```

TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "postauth");
TCLinkPushParam(handle, "transid", "012-1234567890");
TCLinkPushParam(handle, "ticket", "011112222");
TCLinkPushParam(handle, "shipdate", "12-01-2010");
TCLinkPushParam(handle, "type", "moto");
  
```

Debit

Overview

Debit cards enable a direct withdrawal from the user's savings or checking account. Since debit cards represent actual money being transferred, a failure to process, a failure to communicate, or simple unavailability may produce undesirable results such as duplicate charges. To mitigate this, you may want to consider any response other than an "approval" or "decline" as requiring follow up with TrustCommerce or your merchant bank.

The three primary types of debit card transactions are PIN-based debit, signature debit and [PIN-less debit](#).



Important: The ability to process PIN-based debit, PIN-less debit transactions or restaurant (tip support) functionality depends on your processing platform and industry segment. If you have any questions, call (800) 915-1680, Option 5 and speak with a TrustCommerce Solutions Consultant.

Signature Debit

Often called an "offline debit", this type of payment can be made from any check/debit card displaying a Visa or MasterCard logo. Signature debit transactions are accepted with a signature in lieu of a PIN and are processed similarly to credit cards.

PIN-based Debit

A PIN-based debit transaction occurs at point of sale where a cardholder enters their Personal Identification Number (PIN) instead of signing a receipt. Not available everywhere, this type of transaction is common in retail environments. It offers merchants lower processing fees.

PIN-based Debit Sale Fields

The fields described in the table below are generally required to process a debit sale transaction:

PIN-based Debit Sale Fields		
Field Name	Required	Description
pin	Y	This field includes the DUKPT encrypted pin block (16 hex characters) followed by the KSN (16 or 20 hex characters).
surcharge	N	This is the amount in cents of the surcharge assessed on the transaction. This is commonly used by merchants wanting to charge some arbitrary amount of money for using their device. It is the portion of the "amount" field indicating the surcharge. Merchants who wish to charge a surcharge must provide this information as required by the acquiring bank. This amount may be "0" and if this field is omitted altogether, the "surcharge" amount is considered to be "0".



PIN-based Debit Sale Fields		
Field Name	Required	Description
cashback	N	This is the amount of money, in cents, the merchant will return to the cardholder. It is the portion of the "amount" field indicating the cash back. Merchants who want to provide cash back to the cardholder must provide this information as required by the acquiring bank. This amount may be "0" and if this field is omitted altogether, the "cashback" amount is considered to be "0".

- The "amount" is the cost of the goods or services plus the merchant's surcharge and the cash back amount requested by the cardholder.
- The debit "sale" has additional constraints on the fields that may be submitted for processing with respect to the section on "preauth" or "Sale Required Fields". Specifically, "track2" and "amount" must be submitted, or the transaction will not process.
- It is recommended that the integrator not populate the additional fields in that section as the TrustCommerce system may reject the transaction because of other validation criteria (e.g., lack of availability, unsuitability, etc.).

PIN-based Debit – Common Responses

The following table demonstrates the common fields and values returned for a PIN-based debit transaction request. For a comprehensive list, refer to Transaction Responses.

PIN-based Debit - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: approved - The transaction was successfully authorized. declined - The transaction was declined; see Decline Type for further details. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for details.
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS). See Appendix B - Address Verification System (AVS) Response Codes .



PIN-based Debit Sale Example

This example performs a "sale" for a debit card. The track data and PIN information are extracted from the card track data. The cost of the goods being purchased is \$4.00. A surcharge of \$1.00 is included as well as a request for \$20.00 cash back. The net total transaction, therefore, is \$25.00.

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "sale");  
TCLinkPushParam(handle,  
"track2", ",4111111111111111=1215?");  
TCLinkPushParam(handle,  
"pin", "BECD3379A717D40E4B00010000200000001A");  
TCLinkPushParam(handle, "amount", "2500");  
TCLinkPushParam(handle, "surcharge", "100");  
TCLinkPushParam(handle, "cashback", "2000");
```



Important: PIN-based debit refunds are generally processed as (signature debit) credit card refunds. Please see [Credit \(Refund\)](#) for more information.

PIN-less Debit

Overview

PIN-less debit transactions are Card Not Present (CNP) transactions where a customer can use their debit card to make payments online or over the phone without entering their **PIN**. PIN-less debit processing allows certain industry segments to process payments while taking advantage of lower processing fees than traditional credit card transactions. PIN-less debit transactions may be credited as any other debit transaction.

The use of PIN-less debit cards is supported for e-commerce/MOTO integrations with some platforms. Your CustID must be configured to support this transaction type. Contact your TrustCommerce Account Manager or Solutions Consultant for details regarding the support and setup of this functionality.

When integrating PIN-less debit, it is important to be able to distinguish between cards that do or do not support the concept. Functionality has been added to TC Link to assist in that determination.

PIN-less Debit Fields

PIN-less Debit Fields		
Field Name	Required	Description
pinless_debit	Y	Indicates the transaction should be processed as PIN-less debit.
pinless_lookup	N	If set to "y", a BIN lookup on the "cc" number provided is performed first. Depending on the results of the lookup, the transaction is submitted once as a "pinless_debit" transaction or as a standard "cc" transaction.

PIN-less Debit – Common Responses

The following table demonstrates the common fields and values returned for a PIN-less debit transaction request. For a comprehensive list, see Transaction Responses.

PIN-less Debit - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>



PIN-less Debit - Common Responses	
Field Name	Description
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS). See Appendix B - Address Verification System (AVS) Response Codes .

Submitting a PIN-less Debit Sale

If the "pinless_lookup" parameter is not submitted with the transaction, the integrator should do a bank identification number (BIN) lookup prior to submitting the transaction. A BIN is the first six-digits of an account number used by a payment card issuer to identify their institution and can be used to properly identify the card as PIN-less eligible. An example of a simple PIN-less debit transaction might look like the following:

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle,
"cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "1000");
TCLinkPushParam(handle, "pinless _ debit", "y");
```

Using PIN-less Debit BIN Lookup

In the following example, a BIN lookup is first performed locally on the card number provided. A single transaction is attempted based on the result of that lookup, either a PIN-less debit transaction, or a standard credit card "sale" transaction. Integrators should use this field if they do not have the ability to perform a BIN lookup prior to submitting the transaction.

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle,
"cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "1000");
TCLinkPushParam(handle, "pinless _ debit", "y");
TCLinkPushParam(handle, "pinless _ lookup", "y");
```

Electronic Checks/ACH

Overview

Automated Clearing House (ACH) is the back-end process through which electronic checks may be processed. ACH is used to move money to and from checking and savings accounts. The concept of "authorization" does not apply to ACH transactions; it is purely a money transfer. For this reason, the only transaction types available for ACH are "sale" and "credit".

- ACH "credits" are identical to all other types of credits in the TrustCommerce system. Please refer to [Credit2 \(Standalone Refund\)](#) for more details on issuing credits.
- ACH "sales" use the same fields as payment card sales, with the exception of the "cc" and "exp" fields. Instead, they use the following fields: "account", "routing".
- The address verification system (AVS) is not available for ACH transactions, so the AVS setting is ignored.

Electronic Check - Required Fields	
Field Name	Description
routing	checking or savings account routing number
account	checking or savings account number
media	media=ach



Note: ACH transactions can sometimes be associated with savings accounts instead of checking accounts. Not all processing platforms support the "savings" field. Consult a TrustCommerce Solutions Consultant to discuss your specific needs: (800) 915-1680, Option 5.

Electronic Check -Optional Field	
Field Name	Description
savings	This field can be set to "y" to indicate that the referenced routing/account field values are for a savings account. Otherwise, it is assumed to be a checking account.



The following table demonstrates the common fields and values returned for an electronic check transaction request. For a comprehensive list, see Transaction Responses.

Electronic Check - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>accepted - The transaction was successfully accepted into the system.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>

Electronic Check Transaction Example

The following example includes the data necessary to perform an ACH transaction:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "sale");  
TCLinkPushParam(handle, "media", "ach");  
TCLinkPushParam(handle, "account", "123456789123");  
TCLinkPushParam(handle, "routing", "789456124");  
TCLinkPushParam(handle, "amount", "5000");
```


Purchase Card

Overview

Purchase card transaction processing allows government agencies, universities, medium- to large-sized businesses and other qualified entities to process transactions with varying levels of information. This is done primarily for improved reconciliation and reporting and may also result in a preferred discount rate.

Purchase Level II

Purchase Level II cards are used for business to business (B2B) or business to government (B2G) transactions. If you are selling products directly to consumers, you do not need to use Purchase Level II. Additionally, Purchase Level II can only be used if the cardholder is using one of the three types of commercial cards: a corporate card, a business card, or a purchase card.



Note: Depending on the platform, you will receive an indicator specifying whether the card is a business card or a purchase card. A business card will return a value of "B", while a purchase card will return a value of "S" and a corporate card, a value of "R". Submit the additional data based on these indicators and your application should prompt accordingly. If the processor does not support the identifier return and TrustCommerce does not return an identifier, submit the data on every transaction to avoid non-compliance fees and potential discount rate downgrades.

For Purchase Level II, TrustCommerce supports the use of the "preauth", "postauth", and "sale" transaction sets. TrustCommerce supports the American Express Corporate/Purchase Card, but none of the optional charge descriptors.



Important: TrustCommerce supports Purchase Level II and Level III transactions for a variety of processing platforms; however, the specific fields that are available or required is highly dependent on the processing platform you are using. Please consult your TrustCommerce Account Manager or Solutions Consultant prior to attempting to integrate Level II transactions.

Purchase Level II Fields	
Field Name	Description
purchaselevel	Must be specified as "2".
purchaseordernum	Purchase order number "purchaseordernum" is provided by the customer.
tax	The amount of tax, in cents, charged for this order. If the order is tax



Purchase Level II Fields	
Field Name	Description
	exempt, you may omit this field or set the field to "0".
taxidentifier	Can be set to "localsales" or "taxexempt" to indicate the specific tax status of this order.
shipto_zip	The zip code of the destination where the goods are being shipped. If this field is not provided, TrustCommerce will send blanks.
duty	The duty charged (in cents) if the goods were imported. If this field is not provided, TrustCommerce will send zeros.
shippinghandling	The total freight amount of the order in cents (including handling). This field must be populated with numbers only. Any other input results in a badformat error. If this field is not provided, TrustCommerce will send zeros.

The following table demonstrates the common fields and values returned for a Purchase Level II transaction request. For a comprehensive list, see [Transaction Responses](#).



Note: If the processor does not support the "commercialcard" identifier return and TrustCommerce does not return the "commercialcard" identifier, please submit the data on every transaction to avoid non-compliance fees and potential discount rate downgrades.

Purchase Level II - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen, formatted as follows: "123-1234567890"
status	This is a status indicator of your transaction request. Possible values include: approved - The transaction was successfully authorized. declined - The transaction was declined; see Decline Type for further details. baddata - Invalid fields were passed; see Error Type for further details. error - System Error when processing the transaction; see Error Type for further details.
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system



Purchase Level II - Common Responses	
Field Name	Description
	(AVS) for the transaction when the verify parameter (verify=y) is included in the request; see Appendix B - Address Verification System (AVS) Response Codes.
commercialcard	This is indicator specifies whether the card is a purchase, corporate or business card; returned as "S" if it is a purchasing card, "R" if it is a corporate card and "B" if it is a business card.

Purchase Level II Transaction Example - Sale

The following is an example of a Purchase Level II "sale" transaction:

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle,
"cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "0412");
TCLinkPushParam(handle, "amount", "1000");
TCLinkPushParam(handle, "name", "John Smith");
TCLinkPushParam(handle, "purchaselevel", "2");
TCLinkPushParam(handle, "purchaseordernum",
"12345678901234567");
TCLinkPushParam(handle, "taxidentifier", "localsales");
TCLinkPushParam(handle, "tax", "100");
```

Purchase Level II Transaction Example - PostAuth

When applicable, send the "tax", "purchaselevel", "taxidentifier", "purchaseordernum", and "shipto_zip" information on the "postauth" transaction.

The following example is for a "postauth" transaction, including the Purchase Level II fields:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "postauth");  
TCLinkPushParam(handle, "transid", "012-1234567890");  
TCLinkPushParam(handle, "purchaselevel", "2");  
TCLinkPushParam(handle, "taxidentifier", "localsales");  
TCLinkPushParam(handle, "tax", "100");  
TCLinkPushParam(handle, "shipto _ zip", "92606");
```

Purchase Level III

Purchase Level III transactions are essentially Level II transactions with additional data elements. These data elements help the cardholder's business identify items that were purchased and provide information about the order as a whole. TrustCommerce supports Purchase Level III transactions for a variety of processing platforms; however, the specific fields that are available or required is highly dependent on the processing platform you are using. If the appropriate data is not submitted, the transaction may not clear.



FYI: While all issuers support the same fields, the data elements may change from merchant bank to merchant bank. Check with your merchant bank for the correct data-element mapping.

TrustCommerce supports the following items with Purchase Level III transactions:

- Non-travel and entertainment and non-fleet transactions on specific providers
- Data elements for Level II and Level III (II + Line Item Data) transactions.
- Limited "preauth"/"postauth" configuration and full support for the use of the "sale" transaction sets
- TrustCommerce supports the American Express Corporate/Purchase Card but none of the optional charge descriptors.

Purchase Level III Integration



Important: Please consult your TrustCommerce Account Manager or Solutions Consultant before attempting to integrate Level III transactions.



- Additional information about your business is required.
- Additional fields may be defined as needed to process orders outside of the U.S.
- Your acquiring bank may have additional requirements for Level III transactions.

Purchase Level III Fields

Visa Level III transactions may contain the following information about the order as whole:

Purchase Level III Fields	
Field Name	Description
purchaselevel	Must be specified as "3".
vatnum	The VAT Registration number associated with the customer. If this field is not provided, TrustCommerce will send blank spaces.
commoditycode	The commodity code assigned by the acquiring bank. If this field is not provided, TrustCommerce will send "7472".
discount	The total discount amount of the order (in cents). If this field is not provided, TrustCommerce will send zeros. This is the direct sum of the "discount#" items if they are provided individually.
shippinghandling	The total freight amount of the order in cents (including handling). If this field is not provided, TrustCommerce will send zeros.
duty	The duty charged (in cents) if the goods were imported. If this field is not provided, TrustCommerce will send zeros.
shipto_zip	The zip code of the destination where the goods are being shipped. If this field is not provided, TrustCommerce will send blanks.
numitems	This field is used to indicate the number of line item details associated with the order. This is mandatory if line item data is submitted because it will be used to determine how many are available. Up to 99 line items are supported.
amount	For Level III transactions, this represents the sum total of the entire transaction. This value should equal the "tax" + "shippinghandling" + sum("price#" * "quantity#")
tax	The total tax amount of the order (in cents). If this field is not provided, TrustCommerce will send zeros. This is the direct sum of the "tax#" items if they are provided individually.
purchaseordernum	The purchase order number from their Visa purchasing card. This is either a 16- or 17-digit number. If a non-Visa card is being used, do not pass this field.
taxidentifier	Can be set to "localsales" or "taxexempt" to indicate the specific tax status of this order.



Line Item Data

Additional input fields are included to define the individual line items of the order. The hash mark or pound sign (#) in the following field names are replaced with the list item number. For example, if you are referring to the second item, then you would replace "#" with "2" and provide that data accordingly. TC Link supports 999 distinct line items, differentiated by the line item number scheme above.

Line item data may be submitted for Visa Level III transactions as follows:

Purchase Level III - Line Item Data	
Field Name	Description
productcode#	The product code or SKU of the item in question.
price#	The unit price of the SKU or item in question.
discount#	The discount amount in cents. This discount is applicable to that single SKU or product code, regardless of the number of items.
quantity#	The number of units ordered.
productdescription#	<p>A description of the product in question. The product description must be keyed in as follows:</p> <ul style="list-style-type: none">• The first 12 characters of the product description represent the item commodity code. The item commodity code is a Visa specific field with an ISO value assigned to the given product. Contact your acquiring bank for more information on what this field should contain if the commodity list codes referenced in "Commodity Code Listing" is insufficient.• If the number of characters is less than 12, space-fill the remaining characters.• The remaining 35 characters of the product description represent the item descriptor. This must be submitted with non-blank data.
unitofmeasures#	The type of measurement used with respect to the product code. For example, some products might be sold in cases of six, or individually. Hence, the merchant might define this as "CASE" or "EACH" respectively as the unit of measure. This field should be capitalized.
tax#	The tax on this line item in cents. This tax is applicable to that single SKU or product code, regardless of the number of items.
shippinghandling#	Shipping and handling costs associated with this specific line item.
taxamount#	Tax amount applied to this specific line item.
taxidentifier	Can be set to "localsales" or "taxexempt" to indicate the specific tax status of this line item.

Purchase Level III – Common Responses

The following table demonstrates the common fields and values returned for a Purchase Level III transaction request. For a comprehensive list, refer to [Transaction Responses](#).

Purchase Level III - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request; see Appendix B - Address Verification System (AVS) Response Codes.
commercialcard	This is indicator specifies whether the card is a purchase, corporate or business card; returned as "S" if it is a purchasing card, "R" if it is a corporate card and "B" if it is a business card.



Note: If the processor does not support the "commercialcard" identifier return and TrustCommerce does not return the "commercialcard" identifier, please submit the data on every transaction to avoid non-compliance fees and potential discount rate downgrades.



Purchase Level III Transaction Example

The following example will send a Purchase Level III "sale" transaction:

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "2000");
TCLinkPushParam(handle, "purchaselevel", "3");
TCLinkPushParam(handle, "tax", "120");
TCLinkPushParam(handle, "discount", "200");
TCLinkPushParam(handle, "zip", "90000");
TCLinkPushParam(handle, "vatnum", "123");
TCLinkPushParam(handle, "cvv", "999");
TCLinkPushParam(handle, "shippinghandling", "300");
TCLinkPushParam(handle, "numitems", "2");
TCLinkPushParam(handle, "purchaseordernum", "1234567");
TCLinkPushParam(handle, "productcode1", "P003");
TCLinkPushParam(handle, "price1", "300");
TCLinkPushParam(handle, "discount1", "050");
TCLinkPushParam(handle, "quantity1", "2");
TCLinkPushParam(handle, "productdescription1", "20966 PINPAD");
TCLinkPushParam(handle, "unitofmeasure1", "each");
TCLinkPushParam(handle, "productcode2", "T300010");
TCLinkPushParam(handle, "price2", "455");
TCLinkPushParam(handle, "discount2", "045");
TCLinkPushParam(handle, "quantity2", "1");
TCLinkPushParam(handle, "productdescription2", "20693 Terminal");
TCLinkPushParam(handle, "unitofmeasure2", "each");
```


Inventory Information Approval System (IIAS)

Overview

The TrustCommerce processing platform supports Inventory Information Approval System (IIAS) transactions for healthcare applications.

- IIAS is only available to merchants registered with Special Interest Group for IIAS Standards (SIGIS) and using a SIGIS-approved inventory management system.
- Merchants registered with SIGIS under the 90% Rule program are not required to submit these additional fields.
- Not all processing platforms support this functionality. Please check with your Account Manager or a TrustCommerce Solutions Consultant for more information.



Note: Please refer to the IIAS transaction rule set (available for download in the TC Vault) for specific code samples and developer notes.

IIAS Transaction Fields

The following fields are available when submitting IIAS transactions:

IIAS Transaction Fields		
Field Name	Required	Description
iias_transaction	Y	Indicates you are submitting an IIAS eligible transaction.
iias_lookup	Y	Indicates you want TrustCommerce to qualify the eligibility of the card number provided.
partialauth	N	By default, TrustCommerce submits the transaction as if your software supports "partialauth" transactions. If your software cannot accept partial payments, submit this flag as "n".
marketindicator	Y	Submit a value of "m" to identify the transaction as a healthcare specific.
additionalamount1	Y	Indicates what portion of the amount total is Healthcare Eligible. See the section below for formatting details.
additionalamount2	Y	Indicates what portion of the amount total is Prescription. See the section below for formatting details.



Healthcare Eligible and Prescription Amount Fields

The Healthcare and Prescription Amounts field values must be formatted and used in a specific way for each transaction. These are 20 character, fixed-length strings with a prefix indicating the value type, followed by the amount in cents. The Healthcare Amount prefix is "004S840C", followed by a 12-digit, zero-padded integer value representing the amount in cents.

- A Healthcare Amount field for \$10.00 should be submitted as: 004S840C000000001000

The Prescription Amount prefix is "004U840C", followed by a 12-digit, zero-padded integer value representing the amount in cents.

- A Prescription Amount field for \$10.00 should be submitted as: 004U840C000000001000

IIAS Sale Transaction Examples

Example I

The following is an example of an IIAS "sale" transaction where IIAS data is to be included and the integrator is performing its own BIN lookup prior to submitting the transaction.

- The total transaction "amount" is \$25.67 ("amount")
- \$18.23 is healthcare eligible ("additionalamount1",)
- \$15.99 is designated as for prescriptions ("additionalamount2")

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "2567");
TCLinkPushParam(handle, "iias _ transaction", "y");
TCLinkPushParam(handle, "iias _ lookup", "n");
TCLinkPushParam(handle, "marketindicator", "m");
TCLinkPushParam(handle, "partialauth", "n");
TCLinkPushParam(handle, "additionalamount1", "004S840C000000001823");
TCLinkPushParam(handle, "additionalamount2", "004U840C000000001599");
```



Example II

In this example, the total healthcare eligible amount is authorized, but the other amounts are not. This could happen when the card is restricted to healthcare eligible costs only.

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxxx");
TCLinkPushParam(handle, "action", "preauth");
TCLinkPushParam(handle,
"cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "22789");
TCLinkPushParam(handle, "iias _ transaction", "y");
TCLinkPushParam(handle, "iias _ lookup", "y");
TCLinkPushParam(handle, "marketindicator", "m");
TCLinkPushParam(handle, "partialauth", "y");
TCLinkPushParam(handle,
"additionalamount1", "004S840C000000019753");
TCLinkPushParam(handle,
"additionalamount2", "004U840C000000004784");
```

Example III

The following example encompasses several actual transactions demonstrating the "partialauth" capability. In this example:

- It is requested that the application perform the BIN lookup as part of the transaction processing.
- "partialauth" is allowed.
- The total transaction is for \$227.89.
- Of that amount, \$197.53 is healthcare eligible.
- Of that amount, \$47.84 is directly for prescriptions.

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxxx");
TCLinkPushParam(handle, "action", "preauth");
TCLinkPushParam(handle,
"cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "22789");
TCLinkPushParam(handle, "iias _ transaction", "y");
TCLinkPushParam(handle, "iias _ lookup", "y");
TCLinkPushParam(handle, "marketindicator", "m");
```



```
TCLinkPushParam(handle, "partialauth", "y");  
TCLinkPushParam(handle,  
"additionalamount1", "004S840C000000019753");  
TCLinkPushParam(handle,  
"additionalamount1", "004U840C000000004784");
```

Example IV

In this example, the total healthcare eligible amount (\$197.53) is authorized, but the other amounts are not. This could happen if the card is restricted to healthcare eligible costs only. The response fields would look like:

IIAS Sale Transaction Examples	
Response Field	Response Field Value Examples
status	approved
authcode	123456
transid	011-1234567890
partialamount	19753

IIAS Reversal Transaction Example

At this point, depending on the merchant's business rules, you may choose either of the following:

- Reverse "transid" 011-1234567890 and cancel the entire transaction, or
- Get information for another card that can accept the non-healthcare charges and submit another transaction for the difference.

The "reversal" transaction would simply be:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "reversal");  
TCLinkPushParam(handle, "transid", "012-1234567890");
```

IIAS Partial Auth Response Field

IIAS transactions can return partial authorizations if your account is configured to support them. For that reason, the “partialamount” field is returned with IIAS transactions to indicate the partial amount of the authorization.

IIAS Transaction - Partial Authorization Response Field Name	
Response Field Name	Description
partialamount	This represents the amount, in cents, of the authorized amount. This will only be returned if the “partialauth” field is set to “y” in the transaction definition. This will equal the “amount” field if the transaction was fully authorized.

The response to an IIAS transaction with the “partialamount” field returned would look like:

IIAS Transaction - Partial Authorization Response Fields	
Response Field	Response Field Value Examples
status	approved
authcode	123456
transid	011-1234567890
partialamount	19753

Recurring/Installment Payments (Non TC Citadel)

Overview

Merchants who initiate recurring or installment payments outside of the TC Citadel system must send the fields below. These represent transactions that are part of a previously established recurring/installment payment plan.

- Transactions with only the "paymentnumber" field are marked as recurring.
- Transactions with both fields are sent as installment payments.
- Only sale transactions support this indicator.
- In addition to the fields below, it is recommend you submit address verification system (AVS) data on the first transaction and whenever the cardholder changes their address in your records.

Recurring/Installment Fields

Recurring/Installment Fields	
Field Name	Description
paymentnumber	This field is the payment number associated with the transaction. It is required for both recurring and installment payment transactions and is required for a transaction to be flagged as recurring.
totalpayments	This field indicates the number of installments. It is required for installment payment transactions.

Recurring/Installment – Common Responses

The following table demonstrates the common fields and values returned for a recurring/installment transaction request. For a comprehensive list, see [Transaction Responses](#).

Recurring/Installment – Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>



Recurring/Installment – Common Responses	
Field Name	Description
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The “avs” response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request; see Appendix B - Address Verification System (AVS) Response Codes.

Submitting a Recurring/Installment Payment

The following is an example of an installment payment transaction with the first of three payments for \$15.00:

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "1500");
TCLinkPushParam(handle, "address1", "123 Anywhere St");
TCLinkPushParam(handle, "zip", "90210");
TCLinkPushParam(handle, "paymentnumber", "1");
TCLinkPushParam(handle, "totalpayments", "3");
```



Restaurant (Tip)

Overview

Transactions processed in a restaurant environment may want to use the optional "tip" (gratuity) field. The tip field allows the integrator to designate how much of the transaction went towards gratuity. Tip may also be supplied with a "sale".

Restaurant Fields

Restaurant Fields	
Field Name	Description
tip	This is the amount of the tip (in cents) to be included in the "postauth" request (e.g., \$1.00 would be submitted as "100").

Restaurant Transaction Example

In a restaurant environment where tips are provided, the "tip" parameter will be submitted on the "postauth" transaction. The following is an example of a transaction with a tip. This shows an authorization for \$15.00 and added tip from the customer in the amount of \$2.00, for a total dollar amount of \$17.00.

Initial Preauth

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "preauth");
TCLinkPushParam(handle,
"track2", ",411111111111111=07120000000000000000?");
TCLinkPushParam(handle, "amount", "1500");
```

Postauth with Tip

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "postauth");
TCLinkPushParam(handle, "transid", "012-1234567890");
TCLinkPushParam(handle, "tip", "200");
```




TC Citadel Fields

Overview

The following fields are unique to TC Citadel. They can be used when creating BillingIDs for one-time and subscription payments.

TC Citadel Fields	
Field Name	Description
action	"store"
verify	"y" or "n". Applicable to non-ACH transactions. When this field is set to "y", a "verify" transaction is submitted using the card information provided to determine if the card is valid. If the card verification transaction is not supported by the acquiring bank, or if the return is unspecified, a \$1.00 "preauth" will be run on the card to determine validity.
cycle	"d/w/m/y". The time interval which transactions should be processed. <ul style="list-style-type: none">• Days = #d• Weeks = #w• Months = #m• Years = #y TrustCommerce supports numbers from 1-99 and do not require a leading zero.
start	"YYYY-MM-DD". This is the date or offset from the current date in which transactions should start occurring. If a start date is NOT provided, then the start date is presumed to be the day on which the "billingid" record was created. A valid offset has the same constraints as the "cycle" field.
payments	The number of payments to make against the "billingid" record. By default, this field is set to "0" to indicate that the "billingid" requested is either recurring or a store only.
authnow	"y" or "n". This field determines whether TrustCommerce should attempt an authorization immediately for the "amount" specified. TrustCommerce will perform an authorization for the amount specified and attempt to capture (postauth) it on the day specified by the "start" field.
lastpaymentunstore	"y" or "n". This field determines whether TrustCommerce should "unstore" for an installment "billingid" record once all payments have been made. By default, this field is set to "y", which tells us to perform the "unstore".



TC Citadel Fields	
Field Name	Description
store	"y" or "n." This function pushes credit card data to TC Citadel and returns a billingID. This enables merchants to charge the cardholder and keep a record on file for use at a later date. The store function only pushes data to citadel if the sale/preauth is approved. The BillingID is returned in the sale/preauth transaction response. The sale/preauth is associated with the BillingID record.
firstamount	Used if the first payment for a recurring or installment payment is to be different from the regular payment.
lastamount	Used if the final payment for an installment payment is to be different from the regular payment.

Creating a Billing ID

Overview

The following is an example of how to use the TC Citadel for the purpose of storing customer payment card information:

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "store");
TCLinkPushParam(handle, "name", "John Smith");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "zip", "90000");
```

TC Citadel – Responses

The following table demonstrates the common fields and values returned for a store transaction request. For a comprehensive list, see Transaction Responses.

TC Citadel – Responses	
Field Name	Description
transid	“transid” is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: “123-1234567890”
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>
billingid	This is a six-character alphanumeric code used to retrieve customer payment card and ACH information for future, one-time and recurring/installment transactions.

Verifying a Billing ID

Overview

Integrators are able to verify if a card stored in the TC Citadel is valid. To do this, include the "verify" parameter in the initial "store" request.

The following is an example of a "store" request with "verify". This performs a "store" for card number 4111111111111111 with an expiration date of December 2015. A verify transaction is also run to validate the card information.

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "store");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "verify", "y");
```

Verify - Common Responses

The following table demonstrates the common fields and values returned for a verify request. For a comprehensive list, see Transaction Responses.

Verify - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>
avs	The "avs" response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request. Please see Appendix B - Address Verification System (AVS) Response Codes .
billingid	This is a six-character alphanumeric code used to retrieve customer payment card and ACH information for future, one-time and recurring/installment transactions.



Note: If the response field status is returned as "decline", the "declinetype" field will also be included. See Decline Type Field for further details.

Unstoring a Billing ID

Overview

An action type of "unstore" removes access to an existing BillingID. To maintain the integrity of existing transactions the data is not deleted; however, the use of the BillingID for further transactions is permanently disabled. The following are the minimum required fields to "unstore" a BillingID:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "unstore");  
TCLinkPushParam(handle, "billingid", "b12345");
```

Unstore - Common Responses

The following table demonstrates the common fields and values returned for an "unstore" request. For a comprehensive list, see Transaction Responses.

Unstore - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>



Updating a Billing ID

Overview

There may be times that updates or additions to a stored "billingid" are necessary. Updating an existing BillingID is similar to creating a new BillingID record; the difference is that the "billingid" parameter and any fields that you want to update or add to the record are included. The following example demonstrates updating the expiration date for an existing "billingid".

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "store");  
TCLinkPushParam(handle, "billingid", "B12345");  
TCLinkPushParam(handle, "name", "John Smith");  
TCLinkPushParam(handle, "address1", "321 Main St.");
```

BillingID Updates - Common Responses

The following table demonstrates the common fields and values returned for a "billingid" update request.

BillingID Updates - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>



Continuity Payments: Recurring, Installment, and Deferred

Overview

Integrators who want to implement the TC Citadel recurring billing engine extension use the continuity payments feature (recurring, installment and deferred). This feature provides the ability to set up repetitive billing cycles with and without a known end date. In addition, it relieves the integrator of the responsibility of managing recurring transactions through their application. The TC Citadel provides three classifications of continuity payments:

- Recurring Payments - there is no end date and payments continue to run until the card is no longer valid or the "billingid" deactivated.
- Installment Payments - used for regularly scheduled payments when there is a known end date or a known number of payments to be processed.
- Deferred Payments – generally structured by commencing installment payments that will begin on a specified future date and continue until the payment(s) are completed.

The following is an example of a recurring payment transaction. The absence of the payments field indicates that this is a recurring payment transaction and will run indefinitely.

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "store");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "500");
TCLinkPushParam(handle, "name", "John Smith");
TCLinkPushParam(handle, "address1", "321 Main St.");
TCLinkPushParam(handle, "verify", "y");
TCLinkPushParam(handle, "cycle", "1m");
TCLinkPushParam(handle, "start", "3d");
```

Continuity Payments - Common Responses

The following table demonstrates the common fields and values returned for a continuity transaction request. For a comprehensive list, refer to Transaction Responses.

Continuity Payments - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request. Please see Appendix B - Address Verification System (AVS) Response Codes .



One-Time Subscription Transactions

Overview

Occasionally, the TC Citadel recurring billing engine may be unable to process a "billingid" due to a decline response. If this occurs, a one-time subscription transaction may be run against the "billingid" to collect on the outstanding subscription amount.

The following is an example of a one-time subscription transaction using a "billingid":

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "sale");  
TCLinkPushParam(handle, "billingid", "B12345");  
TCLinkPushParam(handle, "amount", "5000");  
TCLinkPushParam(handle, "name", "John Smith");  
TCLinkPushParam(handle, "cyclepayment", "y");  
TCLinkPushParam(handle, "paymentnumber", "3");
```

One-time Subscription Payments - Common Responses

The following table demonstrates the common fields and values returned for a one-time subscription transaction request. For a comprehensive list, refer to Transaction Responses.

One-time Subscription Payments - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request. Please see Appendix B - Address Verification System (AVS) Response Codes .

One-Time Payment

Overview

As mentioned, TC Citadel allows a "billingid" to be used to process a one-time transaction later. This is ideal for integrators who want to manage the recurring payments through their own application. In these cases, the "billingid" field is included as a substitute for the payment card or ACH information.

The following is an example of a one-time sale transaction using a "billingid":

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "billingid", "B12345");
TCLinkPushParam(handle, "amount", "1000");
```

One-time Payments - Common Responses

The following table demonstrates the common fields and values returned for a one-time transaction request. For a comprehensive list, refer to Transaction Responses.

One-time Payments - Common Responses	
Field Name	Description
transid	"transid" is the 14-character unique identifier assigned to each transaction regardless of the status. It is composed of 13 numbers and 1 hyphen formatted as follows: "123-1234567890"
status	<p>This is a status indicator of your transaction request. Possible values include:</p> <p>approved - The transaction was successfully authorized.</p> <p>declined - The transaction was declined; see Decline Type for further details.</p> <p>baddata - Invalid fields were passed; see Error Type for further details.</p> <p>error - System Error when processing the transaction; see Error Type for further details.</p>
authcode	The numeric or alphanumeric code received from the processor in response to a transaction. A value will be provided if the sale transaction had been approved and if the merchant is configured for this response.
avs	The "avs" response code returned by the address verification system (AVS) for the transaction when the verify parameter (verify=y) is included in the request. Please see Appendix B - Address Verification System (AVS) Response Codes .

Fraud Score

Overview

The fraud score represents an evaluation of how legitimate or fraudulent the transaction appears to be. A fraud score close to "0" means that the transaction is likely to be fraudulent, while a fraud score close to "100" means the transaction is more likely to be legitimate. Please see the TC Vault User Guide for more details to assist you in fraud prevention.

With TC CrediGuard enabled, the return fields for each transaction will contain an additional field for the calculated fraud score:

Fraud Score Response Fields	
Response Field Name	Description
fraudscore	The actual numeric score (0 to 100) assigned by TC CrediGuard to the transaction.

Fraud Threshold

When TC CrediGuard is enabled, a fraud threshold can be assigned to each transaction based on its appearance of fraud or legitimacy.

Fraud Threshold Fields	
Field Name	Description
fraudthreshold	This is the fraud threshold for the current transaction. If the transaction receives a fraud score below this threshold, it will be declined. A default value is defined in the TC Vault TC CrediGuard settings. It is submitted via TC Link only if there is a significant business reason to override the specified default value for the account. This value is optional.

Using Fraudthreshold

The following is an example of a sale transaction using the "fraudthreshold" field:

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "1299");
TCLinkPushParam(handle, "fraudthreshold", "45");
```

You may want to handle the score result yourself. You can implement logic that treats a certain range of low scores as outright declines, a high range as approvals, and a middle range as accepted pending further processing. The last category of transactions could be set aside in your local database and trigger an email to a designated person or department to review the transaction and follow up as necessary.

Fraudthreshold - Returned Values

Fraudthreshold - Returned Values	
Field Name	Returned Value
status	decline
declinetype	fraud
fraudscore	32

Blacklists

Overview

TC CrediGuard blacklists are used to detect and block specific cardholder information that you know to be fraudulent.

- No blacklist input fields are submitted via TC Link.
- Blacklist rules are configured from within the TC Vault web interface and apply to all transactions.
- When an account is configured to use TC CrediGuard blacklists, the only change TC Link integrators need to be aware of is the possible additional return fields.
- Refer to the TC Vault Users Guide for more details on the use and configuration of blacklists.

Handling Blacklist Responses

If a transaction is declined because it matches a blacklist rule, the reason for this decline is returned with the following fields:

Blacklist Response Fields	
Field Name	Description
blacklistfield	The name of the input field that contained the blacklisted value; this can be one of the following: "name", "cc", "ach", "zip", "address1", "state", "country", "email", "phone", or "ip".
blacklistvalue	The input value that triggered the blacklist decline; for instance, if the decline was due to a blacklisted IP address, the offending IP address would be returned here.

Velocity

Overview

Velocity rules are used to restrict transactions with certain repeated values, indicating that they may be fraudulent.

- No velocity input fields are submitted via TC Link.
- Velocity rules are configured from within the TC Vault web interface and apply to all transactions.
- When an account is configured to use TC CrediGuard velocity rules, the only change TC Link integrators need to be aware of are the possible additional return fields.
- Please refer to the TC Vault User Guide for more details on the use and configuration of velocity rules.

Velocity Fields

If a transaction is declined because it violates a velocity rule, the reason is returned through the fields described in the following table:

Velocity Fields	
Field Name	Description
velocitytype	This is a brief string that contains the precise reason for the velocity decline: <ul style="list-style-type: none">• The first part of the string contains a velocity type, either "global", "zip", "ip", or "cc".• The second part of the string contains a time period, either "1day", "3day", "15day", or "30day".• The third part of the string contains an indicator of the type of violation, e.g., "count" if the transaction count for the velocity type and time period was exceeded, or "total" if the dollar amount sum was exceeded.
velocityvalue	This is the value that exceeded the velocity rule; for example, if the rule were a limit on CC# usage, this would be set to the last four digits of the credit card number.



Velocity Transaction Example

The following example demonstrates a sale transaction that exceeds the velocity rule regarding the count of transactions on a single card, for a single day:

```
TCLinkPushParam(handle, "custid", "962100");  
TCLinkPushParam(handle, "password", "xxxxxx");  
TCLinkPushParam(handle, "action", "sale");  
TCLinkPushParam(handle, "cc", "4111111111111111");  
TCLinkPushParam(handle, "exp", "1215");  
TCLinkPushParam(handle, "amount", "500");
```

Velocity - Returned Values

Velocity - Returned Values	
Field Name	Returned Value
status	decline
declinetype	velocity
velocitytype	cc1daycount



Address Verification System (AVS)

Overview

The Address Verification System is used to verify the address of the customer presenting the payment card. The AVS system checks the billing address, numeric part of street address and zip code of the payment card provided by the user, against the address on file for that card. This fraud tool is supported for "preauth", "sale", "store", and "verify" card transactions.

The "avs" parameter tells TrustCommerce whether or not to issue "soft declines" based on the AVS result code from the card issuing bank. In some cases, card issuing banks will give an approval even though the AVS information provided did not match the values on file. In these cases TrustCommerce will optionally provide a response that indicates "decline" for AVS. Cardholders may receive false negatives for AVS from e-commerce verification systems, which may require manual overrides, voice authorization, or reprogramming of the AVS entries by the card issuing bank.



Note: TrustCommerce strongly recommends that integrators who encounter card swipe problems (e.g., unable to read the track data) or process mail order/telephone order (MOTO), or e-commerce transactions, submit AVS data.

AVS Fields	
Field Name	Description
avs	"y" or "n". If not included, "n" is assumed.

Using AVS

The following example sends the address "123 Main St" and zip code "926148561" to the card issuer for validation as part of a sale transaction. For addresses that have a number as part of the street name, the numeric form should be sent. For example, send "One Hundred Twenty-Three Main St" as "123 Main St".

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "avs", "y");
TCLinkPushParam(handle, "amount", "5000");
TCLinkPushParam(handle, "address1", "123 Main St");
TCLinkPushParam(handle, "zip", "926148561");
```


Handling AVS Return Codes

The payment card issuer, not TrustCommerce, determines AVS response codes. AVS responses should serve only as a guideline for the integrator to make a business decision as to whether to process a card or not.

TrustCommerce will only decline on “NO MATCH” response; the integrator is responsible for decisions based on other return values. You should use the response code to determine the appropriate business logic for your particular business. For example, if you are shipping physical product and the “avs” response code is “N”, then you might choose not to ship the product until you investigate further. Please see [Appendix B - Address Verification System \(AVS\) Response Codes](#) for a list of AVS response codes.



Card Security Code (CSC)

Overview

Visa, MasterCard, Discover, and American Express implement an additional security feature known as Card Security Code (CSC). The CSC is a three- or four-digit value printed on the card or signature strip, but not encoded on the magnetic stripe. It is a generally accepted best practice to send this data when the card is not present.

The codes have different names unique to each brand, as follows:

- Visa - CVV2 (Card Verification Value)
- MasterCard - CVC2 (Card Validation Code)
- Discover - CID (Card Identification Number)
- American Express - CID or unique card code

This data (or an appropriate indicator if it is unavailable) should be sent for non-swiped transactions. In some cases non-compliance fees may be assessed if no information is submitted. TrustCommerce will attempt to send a response if available. In some cases, this response may be a space (which doesn't indicate anything at all) in the "cvv" field.



Important: For PCI compliance and security reasons, never store or save the CSC value to temporary files.

Card Security Code Fields

The following table describes fields associated with Card Security Code (CSC):

Card Security Code Fields	
Field Name	Description
cvv	The three- or four-character field located in the front or back of the card. For Visa, MasterCard, and Discover, this is a three-character field located on the back of the card to the far right. For American Express, it is a four-character value located on the front of the card above the primary account number.
cvvstatus	<p>This field should be assigned if the "cvv" field is not submitted. Possible values include "present", "notpresent", and "illegible". By default, it will be treated as "notpresent" although some processors may treat it otherwise if this field is not submitted by the integrator.</p> <ul style="list-style-type: none">• The "present" value states that the integrator did not provide the card code stated above even though it was present.• The "notpresent" value states that the integrator did not see a card code.• The "illegible" value states that the integrator could not read the card code. <p>In all cases, this field is used when the CSC is not submitted. By</p>



Card Security Code Fields	
Field Name	Description
	submitting a CSC, the merchant is stating that the CSC is present and the value is the "cvv" indicated above. Only used if "cvv" is NOT provided.
checkcvv	"y" or "n". By default, this is set to "y". This determines whether TrustCommerce should decline the transaction on behalf of the integrator if the CSC does not match as reported by the card issuing bank. Only used if "cvv" is provided.

Using Card Security Codes (CSC)

The following example sends a sale transaction, including the CVV value of "123".

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "1299");
TCLinkPushParam(handle, "cvv", "123");
```

The following example sends a sale transaction where the CVV was not present.

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "amount", "1299");
TCLinkPushParam(handle, "cvvstatus", "notpresent");
```

Handling CVV Response Codes

The “cvv” response field result code is included in the transaction response if the “cvv” input field was included. The following table details possible response codes:

Card Security Code Fields		
CVV Result Code	Meaning	Implementation Comments
M	CVV2/CVC2/Discover CID Match.	None.
N	CVV2/CVC2/Discover CID No Match.	None.
P	Not Processed.	This data was not forwarded to the issuer.
S	Acknowledgement that CVV2/CVC2/CID was not sent.	This would happen because the “cvvstatus” was set to “notpresent”.
U	Issuer doesn’t support this field.	None.

Discretionary Fields

Ticket and Operator Fields

The “ticket” and “operator” fields, described in the following table, are optional fields that can be associated with each transaction. They are built into TC Link and do not use the internal custom field number format.

Ticket and Operator Fields	
Field Name	Description
ticket	“ticket” is a discretionary field that can be used as a merchant-specified transaction identifier for searching and reporting. In some cases, the value of this field is passed through to the financial institutions.
operator	A field the integrator may choose to use if there are multiple operators that are potentially entering transactions. The operator field is only used for reporting within TrustCommerce systems.

TC Transaction Custom Fields

The TC Vault allows merchants to create custom fields. These fields can be assigned names and data types specific to your business operation. As part of that definition, the custom field is associated with an internal custom field number. To enter values for these custom fields using TC Link, the integrator must use the internal custom field number rather than the name displayed in TC Vault, for example, “customfield#”.



TC Transaction Custom Fields	
Field Name	Description
customfield#	As defined in the TC Vault for custom field number.

An example of a sale transaction that includes custom fields might look like the following:

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "sale");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
TCLinkPushParam(handle, "name", "Test Person");
TCLinkPushParam(handle, "customfield5", "N123456");
```

TC Citadel Custom Fields

Similar to custom fields, the TC Citadel offers BillingID custom fields. The interface in TC Link is similar. Instead of referencing these as "customfield#", for TC Citadel, TrustCommerce refers to these as "bcustomfield#".

- To enter values for these BillingID custom fields using T CLink, the integrator must use the internal custom field number rather than the name displayed in TC Citadel.
- One additional field is required for the TC Citadel custom field entry via TC Link. This is value will indicate the specific list of TC Citadel custom field definitions that will follow.

TC Citadel Custom Fields	
Field Name	Description
bcustomfieldlist	This is a comma-separated list of TC Citadel custom field IDs that are included in the transaction.
bcustomfield#	As defined in TC Citadel for BillingID custom field #1.

The following is an example of a store action for a new BillingID that includes specification of TC Citadel custom fields:

```
TCLinkPushParam(handle, "custid", "962100");
TCLinkPushParam(handle, "password", "xxxxxx");
TCLinkPushParam(handle, "action", "store");
TCLinkPushParam(handle, "cc", "4111111111111111");
TCLinkPushParam(handle, "exp", "1215");
```



```
TCLinkPushParam(handle, "name", "Test Person");  
TCLinkPushParam(handle, "bcustomfieldlist", "1,3,5");  
TCLinkPushParam(handle, "bcustomfield1", "N123456");  
TCLinkPushParam(handle, "bcustomfield3", "ABCDEF");  
TCLinkPushParam(handle, "bcustomfield5", "1275");
```

The specific TC Citadel custom field definitions must be created in the TC Vault. The TC Citadel custom field references in this example demonstrate the use of previously created/existing TC Citadel Custom Fields.

Query API

Overview

The Query API provides integrators a way to request transaction data from TrustCommerce. This data may include a single transaction or large quantities of data.

- The input is field/value pairs, similar to the definition of transactions from the previous methods. However, in the Query API, these are search filters instead of transaction definitions.
- There are four types of queries that can be performed: “chain”, “summary”, “transaction” and “billingid” (see Query Types). Each is similar in behavior to reports that can be generated in the TC Vault.
- The Query API returns data in three formats: CSV, HTML, or XML. The type is specified using the “format” field.
- You can only run one report at a time. To obtain both a summary and a chain report, you will need to execute the query twice; once for the summary and once for the chain report.



Important: In all other cases in this document, the TC Link password is used; however, the Query API uses the TC Vault password.

Best Practice

Depending on system conditions, Query API requests may not pick up transactions immediately. For best results, TrustCommerce recommends that you not schedule an automated process to pull transactions immediately at the end of a time window.

For example, if you are creating a tool to pull all transactions on a daily basis for 10:00 p.m. one day through 10:00 p.m. the next day, you should schedule the job to run at 10:15 p.m. or later. This will ensure that all transactions have fully processed before the report runs.



HTTPS/POST Query Types

HTTPS/POST Query Types	
Querytype	Comparable TC Vault Reports
chain	In TC Vault, the user can request a chain report. This report contains a single record summarizing all transactions that are related to one transaction (e.g., "postauth" with the associated "preauth", "credit" with the associated "sale"). If any individual transaction in a chain meets the search criteria, the chain will be included in the chain report.
transaction	In TC Vault, the user can request a transaction report. This report contains specific details of each transaction that meets the search criteria.
summary	In TC Vault, the user can request a summary report. This report summarizes all transactions meeting the search criteria.
billingid	In TC Citadel, the user can request a BillingID report. This report displays information on "billingids" meeting the search criteria.

HTTPS/POST Query Fields

The following table demonstrates the required and optional fields used to build a query:

HTTPS/POST Query Fields			
Field	Possible Values or Format	Required	Description
custid		Y	TrustCommerce customer ID number.
password		Y	Your TC Vault password.
querytype	chain, transaction, summary, billingid	Y	The response corresponding to the equivalent reports on the TC Vault web site.
format	text, html, xml	N	Set to "text" (default) for CSV text output, "html" for HTML format, or "xml" for XML format.
media	cc, ach	N	Filters results based on the media type.
begindate	MM-DD-YYYY HH:MM:SS	N	Filters results based on the beginning date to be included.
enddate	MM-DD-YYYY HH:MM:SS	N	Filters results based on the ending date to be included.
chain		N	Filters results to the specified Chain ID.
transid	012-0123456778	N	Filters results to the specified Transaction ID.
billingid	abc123	N	Filters results to include only transactions referencing the specified "billingid".
pastdue	y,n	N	For querytype=billingid reports, filters results to include only "billingids" that have been flagged as "pastdue".
action	preauth,sale, credit, etc.	N	Filters results to transactions referencing the specified action.
status	approved, accepted, decline,etc.	N	Filters results to transaction reporting the specified status.



HTTPS/POST Query Fields			
Field	Possible Values or Format	Required	Description
name	Any value	N	Filters results to transactions referencing the specified cardholder name; includes partial matches.
cc	Cardholder primary account number (PAN)	N	Filters results to transactions referencing the specified cc; this can be either the full credit card number or just the last four digits.
ticket	Any value	N	Filters results to transactions referencing the specified ticket field value.
limit	Number	N	This field specifies the maximum number of records to return. This works in conjunction with the offset field to allow the integrator to page through results instead of returning large quantities of data.
offset	Number	N	This field specifies the starting point of the records to be returned. This works in conjunction with the offset field to allow the integrator to page through results instead of returning large quantities of data.
showcount	Yes, No	N	Request that the total record count be included on an additional record at the end of the results set.



TC Vault Example

The sample HTML code shown below illustrates the functionality of the Query API. It can be pasted into a file and loaded on your web browser, if you want to try out some queries before you begin to write your code.

```
<html>
  <head> <title> TrustCommerce Query Interface </title> </head>
  <body>
    <h2> TrustCommerce Query Interface </h2>
    <form action=https://vault.trustcommerce.com/query/>
      <table align=center border=1>
        <tr>
          <td> custid </td>
          <td> <input type=text name=custid> </td>
          <td> This is your TrustCommerce custid (required)
</td>
        </tr>
        <tr>
          <td> password </td>
          <td> <input type=password name=password> </td>
          <td> This is your TrustCommerce password
(required) </td>
        </tr>
        <tr>
          <td> format </td>
          <td> <select name=format> <option
value=text>text</option>
                                <option
value=html>html</option></select></td>
          <td> Human readable (html) or computer readable
(text) results</td>
        </tr>
        <tr>
          <td> Query type </td>
          <td> <select name=querytype>
<option value=chain>chain</option>
<option value=transaction>transaction</option>
<option value=summary>summary</option>
<option value=billingid>billingid</option>
</select></td>
```



```

        <td> Type of query </td>
    </tr>
    <tr>
        <td> media </td>
        <td> <input type=text name=media value=cc> </td>
        <td> For now this must be cc </td>
    </tr>
    <tr>
        <td> begindate MM-DD-YYYY HH:MM:SS </td>
        <td> <input type=text name=begindate> </td>
        <td> Query begins at this date</td>
    </tr>
    <tr>
        <td> enddate MM-DD-YYYY HH:MM:SS </td>
        <td> <input type=text name=enddate> </td>
        <td> Query ends at this date</td>
    </tr>
    <tr>
        <td> chain </td>
        <td> <input type=text name=chain> </td>
        <td> Narrow search to a single chain of
transactions </td>
    </tr>
    <tr>
        <td> transid </td>
        <td> <input type=text name=transid> </td>
        <td> Narrow search to a single transactions </td>
    </tr>
    <tr>
        <td> billingid </td>
        <td> <input type=text name=billingid> </td>
        <td> Narrow search to a single billingid </td>
    </tr>
    <tr>
        <td> pastdue </td>
        <td> <select name=pastdue>
        <option value=y>y</option>
        <option value=n>n</option>
        </select></td>
        <td> Use "y" or "n". For BillingID search. Show
only recurring BillingIDs
that have been unable to capture the

```



```
requested funds. </td>
</tr>
<tr>
    <td> action </td>
    <td> <input type=text name=action> </td>
    <td> Narrow search by action. (example:
preauth,postauth) </td>
</tr>
<tr>
    <td> status </td>
    <td> <input type=text name=status> </td>
    <td> Narrow search by status. (example:
approved,accepted) </td>
</tr>
<tr>
    <td> name </td>
    <td> <input type=text name=name> </td>
    <td> Narrow search by name. </td>
</tr>
<tr>
    <td> cc </td>
    <td> <input type=text name=cc> </td>
    <td> Narrow search by credit card field. </td>
</tr>
<tr>
    <td> limit </td>
    <td> <input type=text name=limit value=20> </td>
    <td> Limit results to this number of fields (not
used for summary)</td>
</tr>
<tr>
    <td> offset </td>
    <td> <input type=text name=offset value=0> </td>
    <td> Report results at this offset (used with
limit to page through results)</td>
</tr>
<tr>
    <td> showcount </td>
    <td> <select name=showcount>
    <option value=y>yes</option>
    <option value=n>no</option>
    </select></td>
```



```
        <td> Show the number of not-limited rows on the  
last line of the result</td>  
    </tr>  
    <tr>  
        <td colspan=3> <input type=submit> </td>  
    </tr>  
</table>  
</form>  
</body>  
</html>
```

TC Vault Query API Transaction ID Example

The following example demonstrates what to send to receive chain report for a specified transaction:

```
http://vault.trustcommerce.com/query/?custid=911111&password=XXXXXX&querytype=chain&transid=024-0106430891
```

Enterprise Example

If you have an enterprise account with a Controller ID and sub CustIDs, you can pull records from multiple CustIDs by sending a single query from your Controller ID. The same setup as in the previous TC Vault Example could also be specified.



Important: The referenced controller CustID must control all CustIDs in the specified range. The password defined must be for the controller CustID.

Format I: Comma-Delimited List

In the first format, a comma-delimited list of CustIDs is specified in the "custid" field.

- The Controller CustID must be the first CustID in the list.
- An unlimited list of sub CustIDs can be included after that.
- All sub CustIDs must be defined as being controlled by the referenced controller ID.

If CustID 12340 is the controller, and CustIDs 12341, 12342, and 12343 are defined as being controlled by that CustID, a request for transactions in all of those CustIDs could be obtained by specifying the CustID field as:

```
custid=12340,12341,12342,12343
```

Format II: Range of CustIDs

The second format allows for the specification of a range. In this format, the Controller CustID is specified first, followed by a range of CustIDs. The range is specified by a start CustID followed by "-" followed by the end CustID.

```
custid=12340,12341-12343
```

TC Batch

Overview

The TrustCommerce Batch Processing (TC Batch) service allows submission of a large number of transactions in a single file for offline processing. The file can contain a mixture of transaction types (for example, preauths and postauths, credits, sales, stores and unstores) and is able to use TC Citadel BillingIDs, ACH routing numbers or payment card information to identify the account to be charged.



Note: Multiple CustIDs may be used in a batch file, as long as each transaction contains the CustID and password for the account in which the transaction should be processed.

Batch File Format

The format for submitting batch files is a simple CSV file. The first row of the file contains the various field names as described previously for the TC Link transactions. The order of the columns is arbitrary. You only need to include the columns in which you will be providing data. Subsequent rows each represent individual transactions. A blank value on a transaction row is equivalent to not submitting the field at all.

If positive identifiers for each record are required (such as an invoice number), include that identifier in the "ticket" field of your input file. The value will be echoed back in the "ticket" field of the results file.

A sample batch upload file might look like this:

```
ticket,action,cc,exp,amount,name,email
01-1224,sale,4111111111111111,0404,999,Test
Person,person@example.com
00-1175,preauth,4111111111111111,0404,333,Test
Person,person@example.com
```



Note: Any field found in [Appendix A - Fields List](#) of this guide can be included for batch processing.



Naming Conventions

While there is no required naming convention for the transaction file, TrustCommerce recommends the following:

Batch File Naming Conventions	
Recommended Syntax	Description
batch –[custid]-[mmddyy] [x].txt	This is the input file containing the list of transactions to be processed: [custid] is your numeric TrustCommerce CustID. [mmddyy] is the numeric date associated with this particular batch. [x] is an optional alpha identifier to distinguish if multiple batches are sent in one day.
result-[your file name].txt	This file contains a response record for each transaction record sent in a batch.

Batch Submission

Batches of transactions can be uploaded automatically using HTTPS/POST. Posting a file to TrustCommerce Batch Processing System in this method is only a POST call and does not enable root or any direct access to the servers. Simply send the batch file as a standard HTTPS/POST, with CGI parameters, URL encoded.

HTTPS/POST processing is done through an encrypted URL:

```
https://batch.trustcommerce.com/submit.php
```

The form data posted to TrustCommerce is URL encoded:

```
custid=XXXXXX&password=PPPPP&file=[batch file]
```



An example of how this might work in cURL is as follows:

```
$ curl -F "custid=X" -F "password=P" -F "file=@batchfile.txt"
https://batch.trustcommerce.com/submit.php
The batch file was uploaded successfully.<p><a
href="https://batch.trustcommerce.com/list.php">View Batch
Results</a>
```

Batch Upload Interface Example

The sample HTML code below illustrates the functionality of the Batch Processing service invocation of TC Link. Paste the sample into a file and load on your web browser to experiment with batch processing.



Note: If a client requires batch responses in a certain format, it is best to query the fields from the source response document. Do not assume that the header information will always stay the same on a response file.

```
<h2> TrustCommerce Batch Upload Interface </h2>
<form action=https://batch.trustcommerce.com/submit.php
method=post
enctype=multipart/form-data>
  <table align=center border=1>
    <tr>
      <td> custid </td>
      <td> <input type=text name=custid> </td>
      <td> This is your TrustCommerce custid (required)
    </td>
  </tr>
  <tr>
      <td> password </td>
      <td> <input type=password name=password> </td>
      <td> This is your TrustCommerce password (required)
    </td>
  </tr>
  <tr>
      <td> batch file </td>
      <td> <input name=file type=file> </td>
      <td> This is the batch file to upload (required)
    </td>
  </tr>
</table>
```



```
</tr>
<tr>
    <td colspan=3> <input type=submit> </td>
</tr>
</table>
</form>
</body>
</html>
```

Batch Response

The response file contains one response record for each uploaded transaction record. The results file can be downloaded from the TC Vault or by invoking your language's HTTPS POST function pointed at:

```
https://batch.trustcommerce.com/download.php
```

A sample response file might look like this:

```
ticket,status,transid,billingid,declinetype,errortype,offenders,avs
01-1224,approved,010-0001364786,,,,,N
00-1175,approved,010-0001364785,,,,,N
```



Batch Results Download Interface Example

```
<html>
<head> <title> TrustCommerce Batch Results Download Interface
</title> </head>
<body>
<h2> TrustCommerce Batch Results Download Interface </h2>

<form action=https://batch.trustcommerce.com/download.php
method=post>
  <table align=center border=1>
    <tr>
      <td> custid </td>
      <td> <input type=text name=custid> </td>
      <td> This is your TrustCommerce custid
(required) </td>
    </tr>
    <tr>
      <td> password </td>
      <td> <input type=password name=password> </td>
      <td> This is your TrustCommerce password
(required) </td>
    </tr>
    <tr>
      <td> result file name </td>
      <td> <input name=file type=text> </td>
      <td> This is the result file name to download
(required) </td>
    </tr>
    <tr>
      <td colspan=3> <input type=submit> </td>
    </tr>
  </table>
</form>
</body>
</html>
```

Batch Status

Email is the primary means used by the Batch Processing service to provide status updates. Please contact your Account Manager or TrustCommerce Client Services (800) 915-1680, Option 3 to request your email address be added to the distribution list.

Once your file is uploaded, the system responds immediately with an “acknowledged” message. This is followed by an email providing processing statistics, including the number of records found in the file. The email subject will look like:

```
Subject: Batch File [your file name] Picked Up
```

Once processing is completed, a second e-mail is sent. The body of this email contains relevant information including the number of approvals, declines, accepts and settlements. The email subject this time will look like:

```
Subject: Batch File [your file name] Complete
```



HTTPS/POST

Overview

If you are unable to install TC Link software, you can still process through the TrustCommerce processing platform using HTTPS/POST. This method is only a POST call and does not enable root or any direct access to TrustCommerce servers.



Important: This method should only be used if a TC Link API installation is not an option. It does not have the TC Link failover capabilities, reducing the guaranteed processing uptime. It also lacks some of the enhanced security features of TC Link, and transactions may experience a performance hit of an extra half second.

This transaction should be sent as a standard HTTPS/POST, with CGI parameters, URL encoded. The fields are otherwise identical to those used through the TC Link API. Processing a HTTPS/POST is done via an encrypted URL at: <https://vault.trustcommerce.com/trans.php>

The form data you will post to TrustCommerce is URL encoded.

- The text before the equal sign is the name of the field.
- The text after the equal sign is the value of the field.
- Each pair of field name and field value is separated by the ampersand (&) symbol.

```
custid=XXXXXX&password=PPPPP&action=sale&amount=100&cc=4111111111111111
```

Most people use some sort of software or library to perform the actual transaction (e.g., cURL, .NET WebRequest). You will have to integrate your application to the library or be able to call the software externally in order to use this.

- In the following cURL example, the text before the equal sign is the name of the field.
- The text after the equal sign is the value of the field.
- Each pair of field name and field value is separated by a UNIX newline.

```
$ curl -d "custid=X" -d "action=sale"  
https://vault.trustcommerce.com/trans/  
offenders=password,amount,cc,exp  
error=missingfields  
status=baddata
```

TC Trustee API

Overview

TC Trustee API is a transparent Web application pass through that allows merchants to process payments directly from their Web site without ever handling sensitive cardholder data. TC Trustee API seamlessly integrates into the checkout process of a merchant's shopping cart, payment page, or other online payment system. Customers enter their payment card data on a form on the merchant's Web site, submit the payment form, and the data is seamlessly redirected to the TrustCommerce payment gateway. TrustCommerce stores and transmits the sensitive cardholder data. This solution gives the merchant full branding and customization capabilities.

TC Trustee API is easy to integrate and supports any language that handles web processing, such as ASP.NET, PHP, and Python. In most cases, installation requires fewer than 30 lines of code.



Note: For more information about this, or any of the TrustCommerce products, you may refer to the documentation available in the Downloads section of the TC Vault interface, or call (800) 915-1680, Option 5 and speak with a TrustCommerce Solutions Consultant.

Externally Integrated Applications

Overview

TC Link API is open to developers and e-commerce firms who would like to integrate TrustCommerce services into their front end products. Several third-party products, including shopping carts, patient management systems, and event management systems, have successfully integrated TC Link API within their environments.

To accept payment through these applications, you must first acquire TC Link credentials from your Solutions Consultant and then reference them in your application integration. Please see the documentation from the third-party provider for more information on integrating your application with their product.



Appendix A — Fields List

The following table provides basic validation parameters for the various input fields that can be included in TC Link API transaction requests. Please see the content of this document to understand the context in which these fields can be used and examples for how parameters may be customized.

Fields List		
Field Name	Data Type (Length)	Service
custid	String (20)	All
password	String (20)	All
action	String (10)	All
media	String (10)	Many
amount	Number(3-8)	Many
cc	Number (13-16)	Many
exp	Number (4)	Many
cvv	Number (3-4)	CVV
routing	Number (9)	ACH
account	Number (3-17)	ACH
savings	y/n (1)	ACH
cvvstatus	String (20)	CVV
checkcvv	y/n (1)	CVV
billingid	String (6)	Many
verify	y/n (1)	TC Citadel
transid	String (14)	Many
avs	y/n (1)	AVS
name	String (60)	Personal Info
address1	String (80)	Personal Info
address2	String (80)	Personal Info
zip	String (20)	Personal Info
city	String (40)	Personal Info



Fields List		
Field Name	Data Type (Length)	Service
state	String (20)	Personal Info
country	String (20)	Personal Info
phone	String (30)	Personal Info
email	String (50)	Personal Info
ip	String (7-15)	Personal Info
track1	String (79)	Card Swipe
track2	String (40)	Card Swipe
encryptedtrack	String (512)	Card Swipe
ticket	String (30)	Many
operator	String (20)	Many
shiptosame	y/n (1)	Ship Info
shipto_name	String (60)	Ship Info
shipto_address1	String (80)	Ship Info
shipto_address2	String (80)	Ship Info
shipto_city	String (20)	Ship Info
shipto_state	String (2)	Ship Info
shipto_zip	String (20)	Ship Info
shipto_country	String (20)	Ship Info
purchaselevel	Number (1)	Level II/III
purchaseordernum	String (17)	Level II/III
tax	Number (12)	Level II/III
numitems	Number (3)	Level III
taxidentifier	String (12)	Level II/III
vatnum	String (13)	Level III
commoditycode	String (4)	Level III
discount	Number (12)	Level III



Fields List		
Field Name	Data Type (Length)	Service
duty	Number (12)	Level II/III
productcode#	String (20)	Level III
quantity#	Number (3)	Level III
price#	Number (6)	Level III
tax#	Number (6)	Level III
discount#	Number (12)	Level III
productdescription#	String (47)	Level III
unitofmeasure#	String (12)	Level III
taxamount#	Number (12)	Level III
taxidentifier#	String (15)	Level III
shippinghandling#	Number (3-8)	Level III
pin	String (16-36)	Debit
surcharge	Number (5)	Debit
cashback	Number (12)	Debit
debitdata	String (52)	Debit
originalsaledate	String (14)	Debit
pinless_debit	y/n (1)	Pinless Debit
pinless_lookup	y/n (1)	Pinless Debit
tip	Number (3-8)	Restaurant
shipdate	String (10)	ECom/MOTO
type	String (20)	ECom/MOTO
paymentnumber	Number (6)	TC Citadel
totalpayments	Number (3)	TC Citadel
cyclepayment	y/n (1)	TC Citadel
verify	y/n (1)	TC Citadel
start	String (2-10)	TC Citadel



Fields List		
Field Name	Data Type (Length)	Service
cycle	String (2-4)	TC Citadel
payments	Number (4)	TC Citadel
authnow	y/n (1)	TC Citadel
lastpaymentunstore	y/n (1)	TC Citadel
firstamount	Number (3-8)	TC Citadel
lastamount	Number (3-8)	TC Citadel
customfield#	String (1024)	Cust. Fields
bcustomfieldlist	String (1024)	TC Citadel
bcustomfield#	String (1024)	TC Citadel
iias_transaction	y/n (1)	IIAS
iias_lookup	y/n (1)	IIAS
partialauth	y/n (1)	IIAS
marketindicator	String (1)	IIAS
additionalamount#	String (20)	IIAS
fraudthreshold	Number (2)	TC CrediGuard
demo	y/n (1)	General
offlineauthcode	String (6)	Offline Auth

Appendix B — AVS Response Codes

Address Verification System (AVS) Response Codes

Use the information below to determine the meaning of each AVS response code.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

AVS Response Codes	
AVS Code	Description
A	Street address matches, but five-digit and nine-digit postal code do not match.
B	Street address matches, but postal code not verified.
C	Street address and postal code do not match.
D	Street address and postal code match. Code "M" is equivalent.
E	AVS data is invalid or AVS is not allowed for this card type.
F	Card member's name does not match, but billing postal code matches.
G	Non-U.S. issuing bank does not support AVS.
H	Card member's name does not match. Street address and postal code match.
I	Address not verified.
J	Card member's name, billing address, and postal code match.
K	Card member's name matches but billing address and billing postal code do not match.
L	Card member's name and billing postal code match, but billing address does not match.
M	Street address and postal code match. Code "D" is equivalent.
N	Street address and postal code do not match.
O	Card member's name and billing address match, but billing postal code does not match.
P	Postal code matches, but street address not verified.
Q	Card member's name, billing address, and postal code match.
R	System unavailable.
S	Bank does not support AVS.
T	Card member's name does not match, but street address matches.



AVS Response Codes	
AVS Code	Description
U	Address information unavailable.
V	Card member's name, billing address, and billing postal code match.
W	Street address does not match, but nine-digit postal code matches.
X	Street address and nine-digit postal code match.
Y	Street address and five-digit postal code match.
Z	Street address does not match, but five-digit postal code matches.