SLIM CD, Inc

Transaction Gateway Developers Guide

Version 3.9

Date	Version	Description	Author
5/19/2005	1.0	Initial Design Document	Slim CD
6/29/2005	1.1	XML Revisions2	Mike, Slim CD, Inc.
7/15/2005	1.2	Revisions3	Mike, Slim CD, Inc.
7/17/2005	1.3	Revisions4	Ron, Slim CD, Inc.
7/27/2005	1.4	Revisions5	Mike, Slim CD, Inc.
9/22/2005	1.5	Revisions6	Mike, Slim CD, Inc.
11/18/2005	1.6	Added Blind Credit, Duplicates chapters	Frank, Slim CD, Inc
11/21/2005	1.7	Revisions 7	Mike, Slim CD, Inc.
1/4/2006	1.8	Revision 8	Frank, Slim CD, Inc
1/10/2006	2.0	Revision 2.0, with new license	Frank, Slim CD, Inc.
1/30.2006	2.1	Change "clerkid" to "username" for gift	Frank, Slim CD, inc
1/31/2006	2.2	Changed to add <corporatecardindicator>	Frank, Slim CD, Inc
3/9/2006	2.3	Defined duplicate transaction rules	Frank, Slim CD, Inc
3/27/06	2.4	Updated Appendix B	Frank, Slim CD, Inc
4/10/06	2.5	Updated Appendix F	Frank, Slim CD, Inc
6/20/05	2.6	Changed to identify COM object functions	Frank, SLIM CD, Inc
7/26/06	2.6b	Changed TIPEDIT documentation	Frank, SLIM CD, Inc
8/14/06	2.7	Updated Error Messages	Frank, SLIM CD, Inc
2/28/08	2.8	Enhanced VOID/FORCE processing	Frank, SLIM CD, Inc
9/29/2008	2.9	Updated name/value pair documentation	Frank, SLIM CD, Inc
12/9/2008	3.0	Support for IIAS and FSA / HSA Cards	Frank, SLIM CD, Inc
3/15/2010	3.1	Support for new flags & return values on transactions & improved void processing	Frank, SLIM CD, Inc
4/1/2010	3.2	Support for sending emails or sms text	Frank, SLIM CD, Inc

7/9/2010	3.3	New fields for TSYS autorental AMEX: vehicleclassid,rentaldistanceunit,rentallocationdba,rentaldistance	Z. Wu, SLIM CD, Inc
11/4/2010	3.4	Support for: BillPayment, DebtRepayment, Extra_credit, & Hardware Encryption	Frank, SLIM CD, Inc
07/27/2011	3.5	Updated Links, Parameter Lists and Parameter Descriptions for Web Services. Added GetBatchHistory2, GetBatchSummary and GetDailySummary WebService Methods and Documentation. Added Documentation for new ReverseOrder parameter in SearchTransactions Webservice Method. Updated COM Object	Sean, SLIM CD, Inc
08/01/2011	3.6	Added Documention for GetUserSettings, GetUserClients and GetUserClientSites web services. Updated parameter list for SearchTransactions webservice.	Sean, SLIM CD, Inc.
1/4/2012	3.7	Added ACH processing information in the CHECK section	Frank & Sean, SLIM CD, Inc
9/11/2012	3.8	Added information on gift card processing (seccode, cvv2)	Frank & Sean, SLIM CD, Inc
10/30/2012	3.9	Added information on processor_tokens, transient_tokens, loginname, etc. Enhanced appendix on TOKENIZATION.	Frank, SLIM CD, Inc

TABLE OF CONTENTS

Confidentiality, License & Usage

Licensee hereby agrees that it and its employees, agents, representatives and affiliates will comply with the provisions hereof:

The information herein is provided as CONFIDENTIAL and is not to be distributed without the written permission of SLIM CD, Inc. The Software Technology and Information contained herein will be used solely and exclusively in connection with SLIM CD, Inc.'s gateway systems as understood and contemplated by SLIM CD, Inc and not in any way directly or indirectly detrimental to SLIM CD, Inc.

Slim CD, Inc. hereby grants the Licensee a limited, non-exclusive and royalty-free license to use the sample code, components, and proprietary knowledge contained in this SDK for the express purpose of integrating software with the SLIM CD Gateway services. Integrations using these materials may be required to obtain a "license key" from Slim CD, Inc. for the components to properly function.

SLIM CD, INC. provides these components AS IS and does not guarantee the information contained herein from uninterrupted use or as to fitness for a particular purpose. User agrees to hold harmless and indemnify Slim CD, Inc. for any disputes that arise from use of this material contained herein in any form, or from use of any of the software components contained herein.

DISCLAIMER OF WARRANTY, UNLESS SPECIFIED IN THIS AGREEMENT, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLIDING ANY IMPLIED WARRANTY OR MERCHANTABILITY, FITNESS FOR A PARTICULAR PORPOSE OR NON-INFRINGREMENT ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SLIM CD OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMANGES, HOWEVER CAUSED AND REGARDLESS OF THE TEHORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SDK OR ANY PART THEREOF, EVEN IF SLIM CD HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event will SLIM CD's liability to you, the Point-of-Sale or Web Developer, whether in contract, tort (including

negligence), or otherwise, exceed the amount paid by you to SLIM CD, Inc. for Software under this Agreement. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose. Some states do not allow the exclusion of incidental or consequential damages, so some of the terms above may not be applicable to you.

Licensee recognizes that the Software Technology and Information are confidential information and proprietary to SLIM CD, Inc. Licensee acknowledges and agrees that the Software Technology and Information will consist of, and that Licensee will have access to, Trade Secrets and Confidential Information (as hereinafter defined) of SLIM CD, Inc., and that the disclosure or unauthorized use of such Trade Secrets or Confidential Information by Licensee will injure SLIM CD, Inc. Therefore, Licensee agrees that (other than as explicitly provided herein or agreed to in writing by SLIM CD, Inc.) it will not, at any time, use, reveal or divulge any Trade Secrets concerning SLIM CD, Inc. For purposes of this License, the term "Trade Secrets" shall mean: the whole or any portion or phase of any scientific or technical information, computer source code, technical documentation, design, process, procedure, formula, improvement, confidential business or financial information, listing of names, addresses, or telephone numbers, or other information (whether oral or written) relating to any business or profession which is secret and of value.

Licensee agrees that remedies at law for any actual or threatened breach by it of the covenants contained in this License would be inadequate and that SLIM CD, Inc. shall be entitled to equitable relief, including injunction and specific performance, in the event of any breach by Licensee of the provisions of this License, in addition to all other remedies available to SLIM CD, Inc at law or in equity.

It is the intention of the parties hereto that the provisions of this License shall be enforced to the fullest extent permissible under the laws and public policies of each state and jurisdiction in which such enforcement is sought, but that the unenforceability (or the modification to conform with such laws or public policies) of any provision hereof shall not render unenforceable or impair the remainder of this License, which shall be deemed amended to delete or modify, as necessary, the invalid or unenforceable provisions.

The parties further agree to alter the balance of this License in order to render the same valid and enforceable. The restrictions expressed in this License are in no way to supersede or eliminate any rights which the parties have pursuant to any applicable law pertaining to trade secrets. It is further understood and agreed that no failure or delay by either party in exercising any right, power or privilege hereunder will operate as a waiver thereof, nor will any single or partial exercise thereof preclude any other or further exercise thereof or the exercise of any right, power or privilege hereunder. This License is for the benefit of SLIM CD, Inc. and its assigns and will be governed by and construed in accordance with the laws of the State of Florida..

This Agreement shall be construed and enforced in accordance with the laws of the State of Florida and the venue for any action, dispute or proceeding with respect to this Agreement shall be in Broward County, Florida. Both developer and Slim CD consent to the jurisdiction of any competent court located in Broward County, Florida.

Use of the materials herein constitutes agreement to the terms described above.

For more information, or to contact SLIM CD directly, please go to: http://www.slimcd.com

Introduction

The Gateway

This POS Gateway is a card-present payment Gateway that can process credit card, PIN-Based debit, check cards, checks, EBT cards, EBT paper (vouchers), gift and loyalty card transactions. Unlike other Gateways that only process keyed e-commerce transactions, this system is capable of processing transactions for a number of industries. These industries include:

- Retail

- Restaurant

- Hotel/Lodging

- Auto rental

- Mail Order

- E-Commerce

The credit card processing Gateway accepts swiped (card-present) data. This means the track data can be sent and the transaction will be treated as a swiped transaction from a physical terminal. The Gateway can be used to power POS applications with lightening-fast speed and industry-specific features.

This Gateway accepts PIN-Based Debit data. Just send the encrypted pin block along with the track 2 data. The Gateway will route this data to your processor as a PIN debit transaction. Support has also been added for accepting checks, both paper guarantee services and image truncation services.

All this can be done via a simple HTTP POST to our system. In an effort to make using HTTP POST communications simple, we have provided sample code in a number of different programming languages. In addition, we have created COM objects that can be used to completely insulate the end user (developer) from the implementation of the HTTP POST.

Security

The Gateway systems are current on all security audits, and have been approved by Visa to process credit cards in a safe and secure fashion. Cardholder data is encrypted per Visa's CISP/PCI-DSS requirements. Applications that send data to the Gateway do not retain cardholder information and can easily pass security questionnaires, since no cardholder data need be retained to use this Gateway.

Background:

Slim CD, Inc. develops credit card processing software to replace hardware terminals. The Slim CD solutions include a range of desktop and web-based end-user products, including PC software, virtual terminals and shopping carts. These products send transactions through the

Internet to the Slim CD Gateway system, where they are routed to the major credit card processors. You can learn more about Slim CD by visiting our website at: http://www.slimcd.com.

Purpose

This document provides technical guidelines for interfacing to the Slim CD web services. This document is intended for software developers with a background in the following technologies: XML, SOAP, or HTTPS posting of name/value pairs. Any programming language or platform capable of using XML, SOAP, or HTTPS posting of name/value pairs can be used in with Slim CD web services.

Philosophy

The overall philosophy of Slim CD's transaction processing systems can be broken down into two distinct types: WEB and Point-of-Sale terminals.

WEB Processing

Using the WEB for transaction processing includes:

- Shopping Carts

- Gateway Only Processing

- Custom Integrations

Point-of-Sale

Custom "POS" systems are constructed utilizing:

- Custom "Drop-in" tools (as provided in this SDK)

- Gateway Only "Front-end"

- Hardware Integration/Support (also provided in this SDK)

The purpose of this SDK is to simplify the programmer's job in interfacing with the transaction Gateway utilizing the tools and methods described herein.

Overview

Slim CD Web Services provide access to all our payment processors. All communications take place over secure HTTPS links on the Internet. This transaction type does not require distribution of source code or libraries (DLLs, OCXs, etc). Any device or platform able to originate an HTTPS connection can perform payment processing through Slim CD Web Services. To accommodate various skill sets, platforms, and programming languages, developers will choose the interface of their choice among the three available options.

- **Name/Value Pairs**

    Requests can be sent to Slim CD via an HTTPS POST/GET containing name/value pairs. To use these with our web services, the NAME/VALUE pairs must be URL-ENCODED!

- **Extensible Markup Language (XML)**

    An HTTPS POST of XML data defining the request can successfully use the Slim CD Web Services.

- **Simple Object Access Protocol (SOAP)**

    Slim CD supports version 1.1 of the W3C SOAP specification. Using SOAP, developers may remotely invoke payment transaction types that are defined on the Slim CD web services.

Card Processing

There are several types of transactions. The different types are referenced in this documentation by the value contained in the field **transtype**. It forms the basis of all credit cards, debit cards, corporate purchase cards, gift cards, loyalty cards and check processing transactions.   All transactions during a business day are held in the Gateway in something referred to as the **Batch**. The Batch is "open" during the day as transactions are entered. The Batch is then "closed" at the end of each business day indicating that the transactions held in the Batch have been sent to be processed by the various card processors or banks.

Credit Cards
If the card is a credit card, processing a purchase involves either "swiping" the card or entering the card number and expiration date into the credit card terminal using a keypad. Pressing the "ENTER" button on the card reader (or hitting "RETURN" on the terminal) sends the transaction data (which includes the card number and expiration date) to the Gateway. At this point the card owner's "open to buy" balance (or, available credit) is reduced by the amount of the sale. Note: No money actually flows from the account until the Batch is processed or closed later that day, typically after the business closes.

Debit Cards
If the card is a debit card, an additional piece of data is required: the card owner's PIN number. The PIN number is typically entered by the customer using a keypad and is sent encrypted in a field named **pinblockdata** along with the other card information (such as the card number) to the Gateway to be processed.   The gateway also supports sending the pin data as two separate fields, **ksn** (key serial number) and **pinblock.**

Corporate Purchase Cards
Corporate purchase cards are similar to credit cards, but they require more information. When a transaction involving a corporate purchase card is sent, the response from Slim CD's Gateway will expect extra data, which includes a purchase order number, whether or not tax is to be paid and if so, what kind of tax and how much.   This extra information is sent using a transaction (**transtype**) called an "UPDATE". Please note that sending the Purchase Order and , Sales Tax Type and Sales Tax with the original transaction will avoid having to perform an UPDATE transaction for purchase cards.

A purchase by a customer involving any of the above cards is entered into the Gateway using a transaction (**transtype**) called a "SALE". The steps involved in a "SALE" are shown in the next diagram.

"Credit Card Sale" FLOW DIAGRAM

Credit card transactions

Credit Card Processor

Credit Card "SALE" in Hotel, Restaurant, etc.

"BATCH" Holds the Day's "Credit Card" Sales

Money transfers to merchant's account!

"Close Batch" command at the end of the business day sends "Credit Card" sales held in "BATCH" to processor
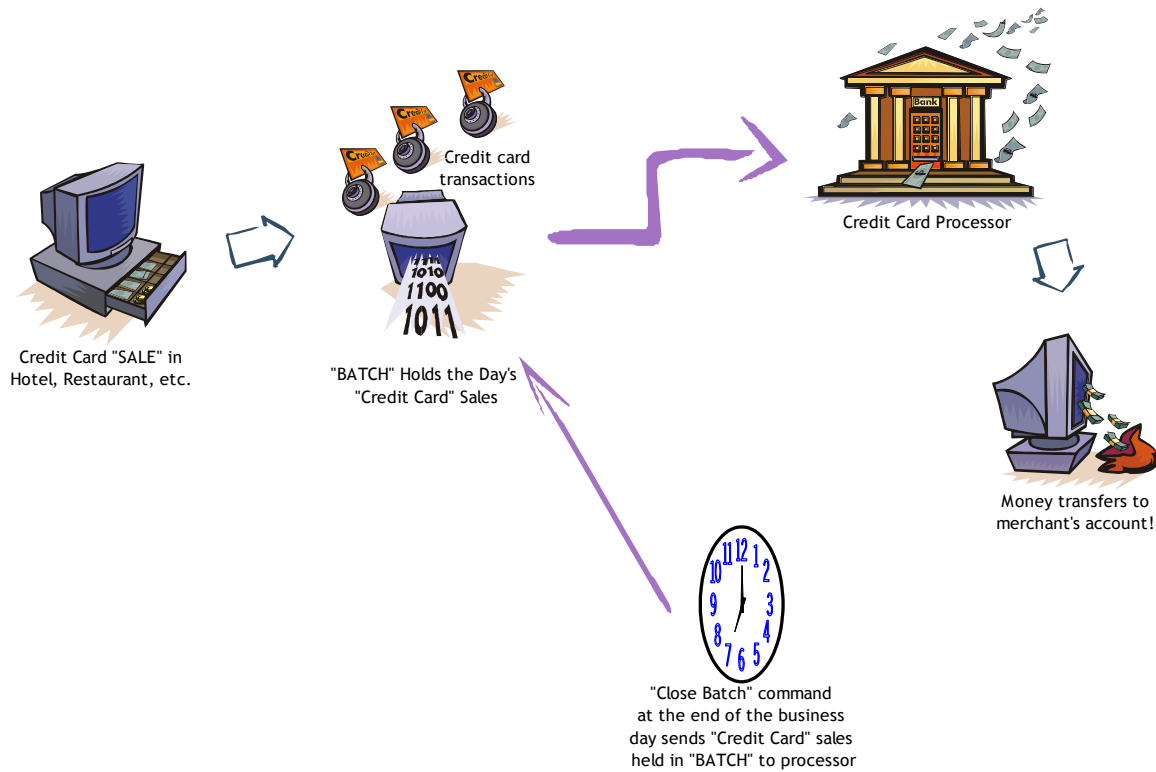
SALE - This transaction reduces the cardholder's "open to buy" balance (or, available credit), and places the transaction in the open batch. This action is commonly used in retail environments.

"Credit Card Offline" FLOW DIAGRAM

Merchant calls Credit Card Processor for "Voice Authorization" for "SALE", "AUTH" or "FORCE"

"Authorization"

Credit Card Processor

Transaction information is entered into terminal when available.

Credit card transactions

"BATCH" Holds the Day's "Credit Card" Sales

Money transfers to merchant's account!

"Close Batch" command at the end of the business day sends "Credit Card" sales held in "BATCH" to processor

OFFLINE – An "OFFLINE" is similar to a "SALE", except that it is not an online action. An "OFFLINE" is a way for a merchant to receive an approval and an authorization code from their processing company via a telephone operator or an automated system. This type of authorization is necessary if the processing company's internet interface, network, or modems are down, or if the merchant has experienced a power failure, computer crash, or other issues that do not allow them to process electronically. An "OFFLINE" may also need to occur if a credit card transaction is flagged (for a variety of reasons) by the credit card issuer, and the credit card issuer requests you call them to obtain the authorization. An OFFLINE transaction is placed in the open batch so that the merchant can be paid for the transaction when the batch closes.

"Credit Card Void" FLOW DIAGRAM

Removes pending sale from BATCH BEFORE it is processed!

Credit card transactions

Credit Card Processor

"BATCH" Holds the Day's "Credit Card" Sales

VOID

Remaining sale money transfers to merchant's account!

"Close Batch" command at the end of the business day sends "Credit Card" sales held in "BATCH" to processor
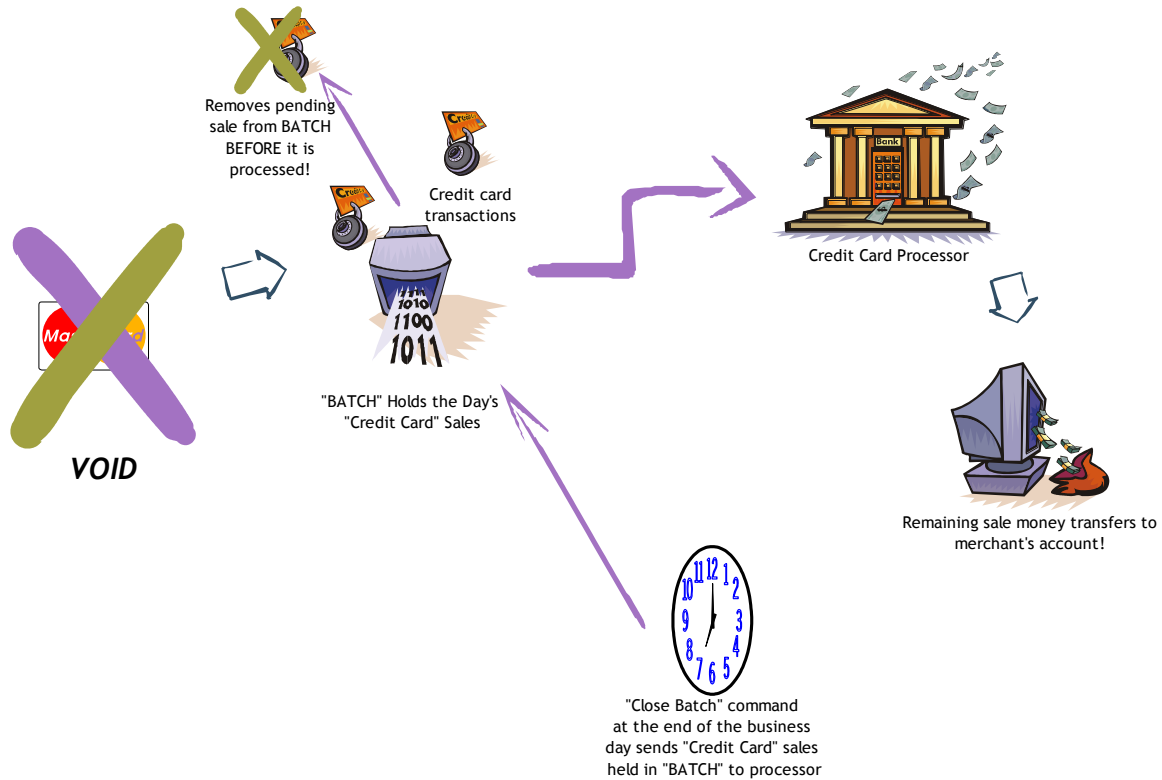
VOID – This action removes a sale, force, or credit transaction from the open batch. No funds will be deposited into the merchant's bank account at settlement/close. The Void Sale action is typically used for same day returns or to correct cashier mistakes. This action can only be performed before the batch is settled/closed (this usually means the action has to be performed on the same day as the sale). Note that AUTH transactions can also be voided, but this simply prevents them from being FORCED (placed in the batch) later and removes them from the OPEN AUTH reports. On some processors or for some cards, VOIDing an AUTH or SALE does not actually release the hold the cardholder's bank has placed on the funds.

E-COMMERCE "Credit Card Credit" FLOW DIAGRAM

Credit card transactions

Credit Card Processor

"CREDIT" Requested

"BATCH" Holds the Day's "Credit Card" Sales & "Credits"

Remaining sale money transfers to merchant's account!

Customer receives CREDIT!

"Close Batch" command at the end of the business day sends sales & credits held in "BATCH" to processor
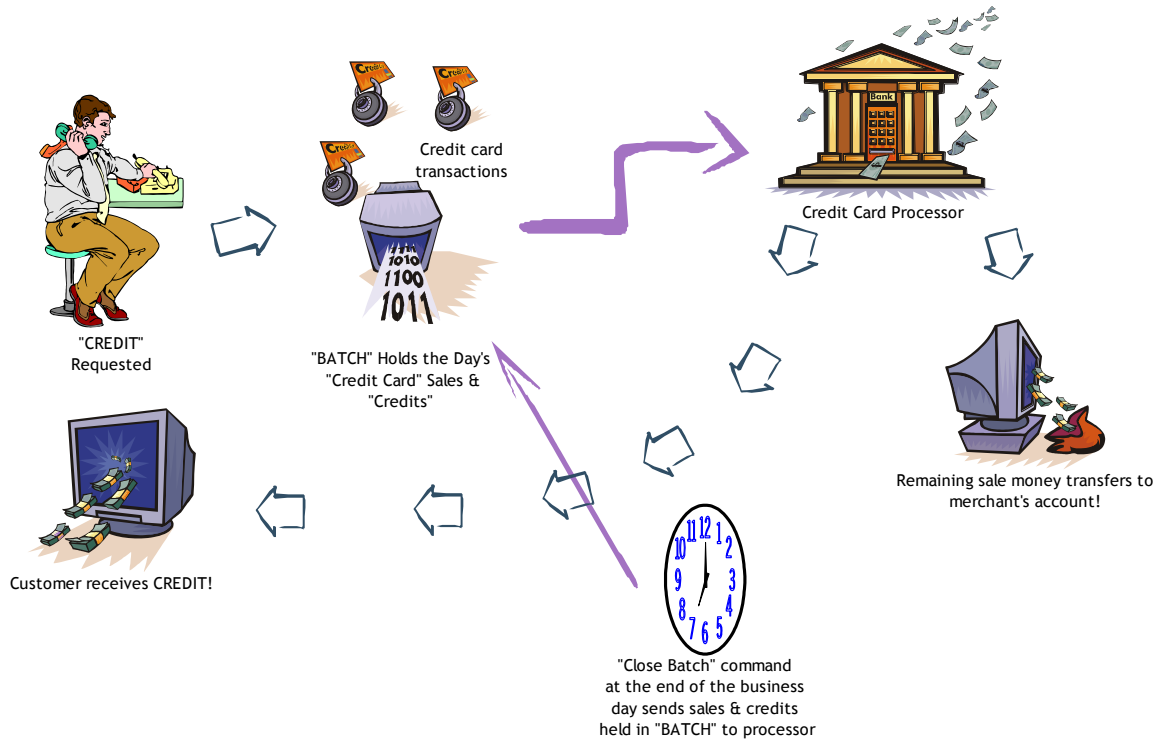
CREDIT – This action is used to refund money to the cardholder. This action is typically used after the batch that contains the Sale (or "FORCE") transaction has been settled or closed. This action will place funds back on the cardholder's card once the batch containing the credit has been settled. Usually, the funds appear back on the cardholder's card within 3 to 5 days.

"Credit Card "Auth" & "Force" FLOW DIAGRAM

Credit "AUTH" (or "Reversal")
(modify or remove previous "Auth")

Pending "Auth" record
is modified or removed
entirely!

Credit card
transactions

Credit Card Processor

"AUTH(s)" held until a "FORCE"
is received. Then, transaction is
moved to "BATCH", with other "SALES"

"BATCH" Holds the Day's
"Credit Card" Sales &
"Credits"

Money transfers to
merchant's account!

"FORCE" command!
(SALE is made)

"Close Batch" command
at the end of the business
day sends sales & credits
held in "BATCH" to processor

AUTH – This action only reduces the cardholder's "open to buy" balance (or, available credit). It does not place a transaction in the open batch. Auth can be considered the first half of a sale. An Auth reduces the "open to buy" balance (or, available credit) for only a predetermined amount of time, usually 3-30 days. The credit card's issuing bank determines the amount of time. To place the transaction in the open batch, a follow-on transaction (called a "FORCE") must occur. This action is commonly used in **MOTO** (Mail Order / Telephone Order), Restaurant, Hotel, Auto-rental and e-commerce environments.

INCREMENTAL – An "INCREMENTAL" action increases the authorization amount on a previous "AUTH" (or "OFFLINEAUTH") This transactions is industry-specific and can be used by Hotels and Mail Order merchants.

REVERSAL – A credit card "REVERSAL" reverses part or all of a previous "AUTH" transaction, and frees up that portion of the "hold" placed on the cardholder's funds. This transactions is industry-specific and can be used by Hotels and Mail Order merchants.

FORCE – This action places an approved "AUTH" transaction into the open batch. This action can be considered the second half of a sale. This follow-on transaction must occur before an "AUTH" can be settled/closed.

TIPEDIT – This action allows changes to the gratuity amount anytime before the batch is closed ("processed"). You would use TIPEDIT to correct mistakes in the tip or total amount or both.

> **IMPORTANT**
>
> BE CAREFUL! The **total amount** you send with TIPEDIT *WILL* become the total of the credit charge, unless you use the COM object, in which case the tip will be added to the total to produce the amount to charge to the card!

OFFLINEAUTH – An "OFFLINEAUTH" is similar to an "AUTH", except that it is not an online action. An "OFFLINEAUTH" is a way for a merchant to receive an approval and an authorization code from their processing company via a telephone operator or an automated system. This type of authorization is necessary if the processing company's internet interface, network, or modems are down, or if the merchant has experienced a power failure, computer crash, or other issues that do not allow them to process electronically. An "OFFLINEAUTH" may also need to occur if a credit card transaction is flagged by the credit card issuer. In order to place an "OFFLINEAUTH" in the open batch, a follow-up transaction, called an OFFLINEFORCE must occur.

OFFLINEFORCE – This action is similar to a "FORCE" with the exception that it is to be matched with an OFFLINEAUTH. This action, like the regular "FORCE", can be considered the second half of a sale. This follow-on transaction must occur before an "AUTH" (or "OFFLINEAUTH") can be settled/closed.

"Credit Card Update" FLOW DIAGRAM

Credit card transactions

Credit Card Processor

Record in "BATCH" gets modified.

Previous Credit Card "SALE" is "updated" with tax and "Purchase Order" information.

"BATCH" Holds the Day's "Credit Card" Sales

Money transfers to merchant's account!

"Close Batch" command at the end of the business day sends "Credit Card" sales held in "BATCH" to processor
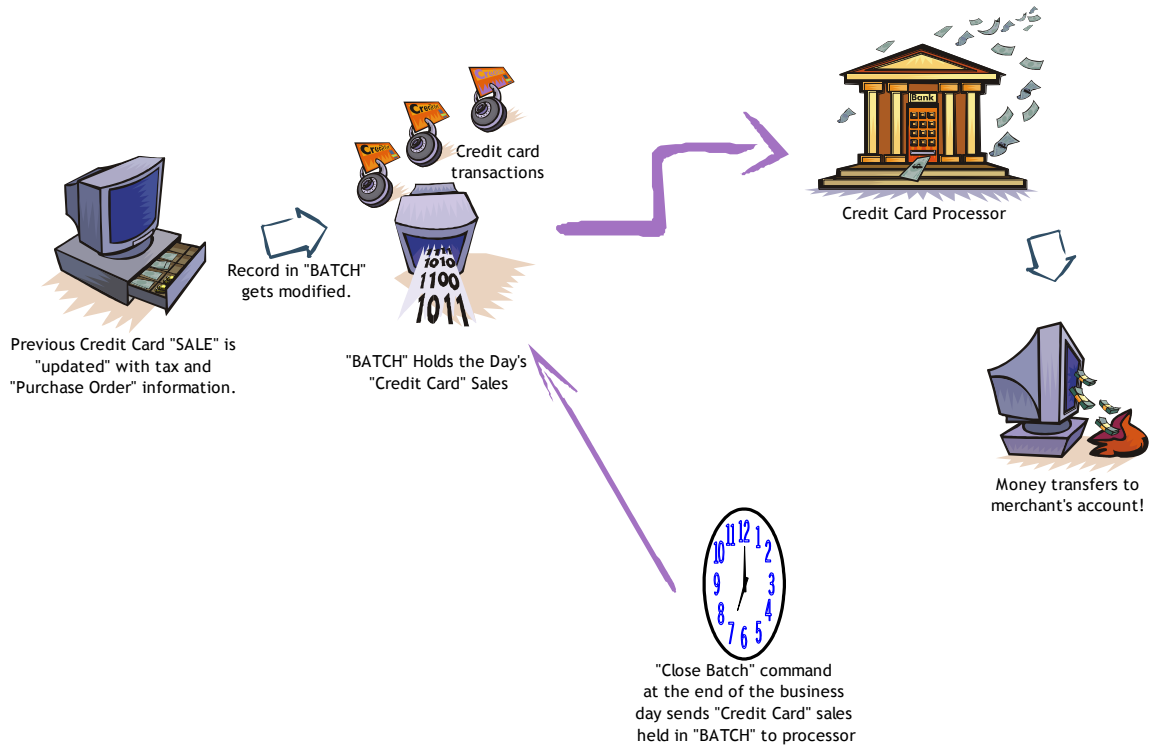
UPDATE – An "UPDATE" action adds tax and purchase order information to an existing "SALE". This is typically used with corporate or government purchasing cards. Please note that you can avoid the need to perform the UPDATE secondary transaction by sending the Purchase Order, Sales Tax Type and Sales Tax with each original transaction.

There are two basic types of check services performed by the system, verification/guarantee and truncation/conversion. These services differ in what they offer to the merchant.

Verification / Guarantee

The "verification" service will verify that the check writer is not in the "known bad check writer" database. The "guarantee" service guarantees that the check service provider will reimburse the merchant for any returned checks and perform collection on the merchant's behalf.

Electronic Check Conversion (ECC)

A conversion operation will scan a paper check and extract the bank account information for processing, thereby "converting" the check to an electronic transaction. This conversion changes the check from a paper check to an ACH Debit from the check writer's account. The check service provider may also guarantee the truncation/conversion if the merchant's account was set up to provide this service as part of the truncation/conversion. Please note that not all checks can be converted. Those that cannot be converted may still require physical bank deposits. The messages returned from the services will indicate which checks must be retained and deposited normally in the bank.

Business Rules for Processing

Accepting Payments in a Retail Environment

Retail environments traditionally process real-time payments for the purchase of goods and services. A retailer can accept credit cards, debit cards, checks, and gift/loyalty cards. Most transactions can be swiped or keyed, although swiped is usually preferred. Keyed transactions, when allowed, usually result in higher processing fees to the retailer.

Retail transactions

Retail transactions are usually performed during a single interaction with the processing system, using the "SALE" transaction type. Transactions processed for certain types of business or commercial credit cards are often accompanied by additional data. When requested by the system, an additional transaction can be performed to "UPDATE" this information if it was not originally provided as part of the "SALE".

The "SALE" transaction type can be used to accept credit cards, debit cards, gift/loyalty cards, and to perform certain operations with paper checks. In most cases, this single transaction type will complete most of the processing in a single step. In certain situations (such as with corporate cards or check images) additional data may need to be sent in a secondary transaction or process.

Restaurant Environments

Restaurant environments traditionally process real-time payments in two phases so that a tip may be added to the purchase. A restaurant can accept credit cards, debit cards, checks, and gift/loyalty cards. Most transactions can be swiped or keyed, although swiped is usually preferred. Keyed transactions, when allowed, usually result in higher processing fees to the restaurateur.

Restaurant transactions are usually performed as a pair of transactions, one that authorizes the card and holds the funds ("AUTH") and the second that adds the tip and schedules the transaction for payment ("FORCE"). Tip entries and transaction totals may be adjusted for "FORCE" or "SALE" transactions as long as the batch has not been closed using the "TIPEDIT" command.

TIP HANDLING AND TOTAL AMOUNTS

The amount set to the gateway is always the amount to charge the card. Tip amounts can be specified on SALE, FORCE and TIPEDIT commands, but the AMOUNT value should also be sent and should represent the TOTAL AMOUNT TO CHARGE THE CARD.

Here are some examples:

    AUTH   **Amount = 10.00**

    FORCE   **Amount = 12.00, Tip = 2.00 (Total sale charged to card = 12.00)**

SALE   Amount = 12.00, Tip = 2.00 (Total sale charged to card = 12.00)

TIPEDIT   Amount = 12.00, Tip = 2.00 (Total sale charged to card = 12.00)

AUTH   Amount = 10.00

FORCE   Amount = 10.00, no tip specified

TIPEDIT   Amount = 12.00, tip = 2.00 (NEW total sale charged to card = 12.00)

Transactions processed for certain types of business or commercial credit cards are often accompanied by additional data. When requested by the system, an additional transaction can be performed to "UPDATE" this information if it was not originally provided as part of the "SALE", "AUTH" or "FORCE".

The "AUTH" transaction type can be used to accept credit card transactions, but a "SALE" transaction is required to accept debit cards, gift/loyalty cards, and to perform certain operations with paper checks.

There are specific requirements for each type of payment.
- Credit cards are usually the easiest to implement, as they may be swiped or keyed.
- Debit transactions need to be accompanied by a "PINBLOCK" and "KSN" that contain the encrypted pin information from a physical hardware pinpad device.
- Check processing usually requires hardware to read the check's MICR data or to scan the entire check as an image.
  - Check verification / guarantee services are done in a single step, often using a MICR reader to acquire the check's bank information
  - Check truncation / conversion services require an image of the check to be uploaded after the transaction is approved
- Gift/Loyalty cards may impose additional receipt requirements and additional balance inquiry functions may need to be supported for the retailer to participate in the gift card programs.

Swiped vs. Keyed
In the retail environment, transactions are usually processed by swiping the credit card through a magnetic card reader. When a card is swiped only the fields "trackdata" and the "amount" needs to be supplied. When a card is manually entered or "keyed", additional data may be sent as a fraud reduction tool and to avoid processing penalty fees. This additional data includes items such as the cardholder's zip code or the CVV2 value from the back of the card.

Important! Cost of Transactions!

The merchant's "cost of transactions" can vary greatly depending on whether or not the cards are "swiped" through a magnetic reader (lower fees!) or hand entered from a keypad (higher fees!) To make sure your customers get the benefit of lower transaction costs, always set the fields "cardpresent" and "readerpresent" to "**Yes**" for "swiped" cards (that's when track two data is present in the field "trackdata").

Corporate Card Processing

The credit card system may return a response that indicates additional data is requested for the transaction. This occurs whenever a "purchase/business/commercial card" was used, and has been approved. If you have already supplied the "po", "salestax", and "salestaxtype" values, you may ignore this value and treat the response as a normal approval. Otherwise, simply supply this information, along with the "gateid" returned in the response, as part of an "UPDATE" command. Note that even if you do not perform the "UPDATE" command, the transaction has already been approved so that the merchant will be paid and the charge will appear on the cardholder's statement. The value "corporatecardindicator" will be included in the response to indicate the specific processor's letter for each of the different types of corporate cards (for example, B, R, S, etc)

Voice Authorizations:

This occurs when a swiped or keyed transaction returns a message to call the processing center. When the retailer calls the center as instructed, they may be given a "voice authorization code". This must be processed into the system so the merchant will be paid for the authorization. An "OFFLINE" transaction is used for this purpose. The "OFFLINE" transaction functions the same way a "SALE" transaction does, except the voice authorization "authcode" is supplied as part of the transaction.

Batching out

The system automatically BATCHs transactions every day. This functionality can be overridden so that only manual Batching occurs. There are also Batch close commands available within the web-based user interface and through the programmer's API toolkit.

Batching is done with a simple processing request to the host, which closes the Batch. Any Batch errors that may occur result in a declined Batch close, and a notification will be sent to the administrator of the processing system so the Batch can be successfully closed on the merchant's behalf. Once a Batch is closed, certain transactions cannot be performed on items in the Batch. This includes the "VOID", "UPDATE", and "TIPEDIT" transactions.

Duplicates

Duplicate transactions usually occur inadvertently when a transaction is accidentally submitted twice. To prevent this, the gateway looks for and rejects duplicate transactions (if you actually WANT to submit duplicate transactions, this behavior may be overridden by setting the "allow_duplicates" field to "Yes".)

Duplicate transactions are defined as where the card number, amount, transtype, etc. is the same in two or more transactions. Transactions are identified as a duplicate as follows:

1. The same clientid/siteid combination is used
2. The same cardnumber is used
3. The same transaction type is used
4. The same amount is specified
5. No invoice number (client_transref field) is sent, or the same invoice number is used
6. The transactions occurs within a short time interval of each other(30 minutes, but this value can be changed using the "duplicates_window" variable)

Submitting a duplicate transaction will result in your receiving one of the following two error messages:

DUPLICATE_TRANSACTION_ALREADY_APPROVED
DUPLICATE_TRANSACTION_PENDING

The "already_approved" message means that a similar transaction has been approved within the past 30 minutes or so (or, based on the value of "duplicates_window").

The "pending" message means a transaction has been posted to the gateway for processing but has not yet received an approval or decline from the issuer (two transactions must be sent within a second or so to get this condition).

As a developer, you should write your code to:
1. Direct the merchant to the WEB reporting URL to check for the duplicate and re-print a receipt, or…
2. Use the WEB services to pull the original transaction data and print a receipt programmatically.

Reports

Reporting tools are available online from the web-based user interface. The programmer's API also includes tools that download the data necessary to produce a variety of reports.

There are specific requirements for each type of payment.
- Credit cards are usually the easiest to implement, as they may be swiped or keyed and can be "pre-authorized" with a tip added later.
- Debit transactions need to be accompanied by a "PINBLOCK" that contains the encrypted pin information from a physical hardware pinpad device.   The tip may need to be pre-determined, as a "pre-author
- Check processing usually requires hardware to read the check's MICR data or to scan the entire check as an image.
    - Check verification / guarantee services are done in a single step, often using a MICR reader to acquire the check's bank information
    - Check truncation / conversion services require an image of the check to be uploaded after the transaction is approved

- Gift/Loyalty cards may impose additional receipt requirements and additional balance inquiry functions may need to be supported for the retailer to participate in the gift card programs.

The gateway software system automatically detects duplicate transactions. This is the default configuration. You can override this functionality by specifying a value for the "allow_duplicates" variable. This variable can be set to the value of "yes" or "no", where "no" is the default. Setting this value to "yes" will prevent the software from automatically detecting duplicate transactions and declining them. (see the preceding chapter on Business Rules for additional information).

When the software identifies a duplicate transaction, one of the following messages is returned:

```
DUPLICATE_TRANSACTION_ALREADY_APPROVED
DUPLICATE_TRANSACTION_PENDING
```

When your software receives this message, you can override the duplicate message by setting "allow_duplicates" to "yes" if you really expect this duplicate transaction. If you were not expecting this transaction to be a duplicate, this message indicates the transaction has been previously submitted in the past few minutes for the exact same amount on the exact same cardnumber, and that transaction was approved. **You may want to use one of the "SearchTransaction" web services to retrieve that previous approval data, or have some other method of handling duplicates in your software.**

Transactions are identified as a duplicate as follows:

1. The same clientid/siteid combination is used
2. The same cardnumber is used
3. The same transaction type is used
4. The same amount is specified
5. No invoice number (client_transref field) is sent, or the same invoice number is used
6. The transactions occurs within a short time interval of each other(approximately 30 minutes)

See the section above on "Business Rules" to learn more about handling duplicate transactions.

Control window for duplicate checking. We have added finer control over the automated features that search for duplicate transactions. To give the developer better control, a new (and OPTIONAL) variable, **"duplicates_window"** has been added. This value can be sent to the gateway, along with the normal payload (as XML or name/value pairs).

The default for this variable is 30, which represents the current configuration of 30 minutes to search backwards for a duplicate.

Some developers requested to specify this window themselves. The value is specified in minutes, with a maximum value is 20,000. We recommend using the default, but the duplicate checking window can be expanded or tightened using this setting.

Note that setting "allow_duplicates" to "yes" will disable all duplicate checking and avoid using this setting altogether.

The gateway software system can be forced to automatically match each CREDIT (refund) transaction with a previous SALE or FORCE. The default is enabled for this functionality, and you may not submit CREDIT transactions without specifying a gateid. You can override this default behavior by using the "blind_credit" variable. This variable can be set to the value of "yes" or "no", where "no" is the default. Setting this value to "no" will cause the software to require a "gateid" from a past transaction in order to allow a CREDIT to be performed, whereas setting this to "yes" avoids the need to supply a gateid.

By default, SLIM CD restricts the amount of a refund so that the total of all refunds does not exceed the total amount of the original sale. This feature can be disabled by setting "extra_credit" to "yes", indicating you wish to allow a refund for more than the amount of the original sale. Note that when using an old gateid (ticket number) for tokenization purposes, we recommend using extra_credit=yes so that all future refunds are permitted.

1. **Enhanced VOID processing**. The current method of performing a VOID on the SLIM CD system requires the developer to submit a "gateid" that specifies the record to void. However, if there is a communication issue, the developer may not have a gateid available. Or, the developer may have a legacy application that is completely based on credit card numbers and cannot support sending the "gateid".

SLIM CD has enhanced its VOID verb as follows. Specifying an exact gateid is the preferred method if identifying the specific transaction to void. However, if a "gateid" is NOT sent, then the system will attempt to find a gateid based on a variety of developer-supplied search parameters. These parameters are:

- clientid/siteid
- cardnumber or processor_token
- amount
- client_transref (also known as "invoice number" in various reports)
- match_window (optional, in minutes, with a default of 2 days… or 2880 minutes)

The VOID verb will search for a transaction using the above user-supplied information along with additional requirements as follows:

a. The original transaction to be voided must have been processed using the same "clientid/siteid" as the current transaction.

b. If the amount is positive, the original transaction to be found must be a "SALE", "FORCE", or "OFFLINEFORCE" transaction.   If the amount is negative, a "CREDIT" transaction will be voided.

c. The transaction must not have already been VOIDed. (In other words, previously voided transactions are ignored when searching for the transaction to VOID)

d. The "client_transref" value will match that sent with the original transaction. If no value was sent with the original transaction, no value should be sent when attempting to void the transaction. If no value is sent on the VOID, then the client_transref value will not be used to identify a transaction to void.

e. The transaction must have occurred within the "match_window" time frame. This is an optional parameter, which defaults to 2880 minutes, or 2 days. This value is specifed in minutes, with a maximum value of 100,000 minutes. Values above 100,000 will be ignored, and the default will be used.

f. Only "approved" transactions will be located when attempting to VOID. This means that if a transaction is still processing on our side (approved=U for UNKNOWN), it will NOT be found using this search criteria. This means you should wait a few seconds before auto-voiding transactions on connection

interruptions as they might not be finished approving. This also means if you attempt to auto-void a DECLINED transaction, no harm will be caused as long as there is not an earlier approved transaction.

Please note that the transactions will be located NEWEST ONE FIRST, which means that to void an OLDER transaction of the exact same cardnumber, amount, and client_transref, you must first void the NEWER transaction. To be more technical, the search orders by ID in descending order so that the newest non-voided transaction (with the largest gateid) is found. The reason for this is to allow the most recent transaction to be voided, since it is probably the duplicate.

Finally, if a match cannot be found, the Error message will be logged to our database and returned to the caller with the error message text :

"*--MATCH_NOT_FOUND".

Note that the asterisk implies the standard meaning that you can re-try the transaction by changing data (such as the match window).

2 (A). Enhanced FORCE/OFFLINEFORCE processing. The enhanced FORCE processing works similarly to the above VOID processing. The search criteria are the same, as are the "match_window". There are some important differences:

a. The "match_window" defaults to 15 days instead of two days.

b. The search ignores the amount, as amount does not need to match exactly.

c. The search function will only look for either AUTH or OFFLINEAUTH transactions that match the cardnumber.

d. The search **WILL** return AUTH transactions that <u>have already been FORCED</u>. Therefore, if you AUTH a card and then do it again, you can only find the newest auth using the automated feature. You cannot use this feature to get each AUTH in reverse age, the way the VOID operates.

   NOTE: You can refine the search to better identify a previous AUTH transaction by supplying the original AUTH transactions' AUTHCODE when performing the FORCE transaction. If the AUTHCODE is supplied, it will be used in addition to the cardnumber when searching for an earlier AUTH transaction.

e. AUTHs that have been VOIDED (which must be done by VOID with a gateid) will not be returned using this search feature.

2 (B). Enhanced FORCE/OFFLINEFORCE processing. The next enhancement has to do with automatic conversion of FORCE and OFFLINEFORCE. You can now send either FORCE or OFFLINEFORCE to the gateway interchangeably. The gateway will

identify the original transaction as an AUTH or OFFLINEAUTH, and will set the current transaction type to match the original.

Example: If a FORCE transaction is attempted on an OFFLINEAUTH transaction, the FORCE will be converted into an OFFLINEFORCE before attempting to complete the transaction.

This means you can now just use one FORCE verb for either AUTH or OFFLINEAUTH transactions and you are no longer required to store the type of the original AUTH (or OFFLINEAUTH). Our system will perform that for you.

3. VISA/MC/Discover Enhancements that effect VOID processing. The card associations now allow an authorization (as performed by an AUTH or SALE) to be reversed. This is known as a FULL REVERSAL. Rather than make this it's own command, SLIM CD has elected to implement automatic FULL REVERSAL functionality as part of the VOID processing. If the original transaction caused an online authorization (AUTH or SALE), then performing a VOID on that original transaction will now automatically perform an online FULL REVERSAL to release that VOID. If a VOID is performed on a transaction that does not perform ONLINE authorizations (OFFLINE, OFFLINEAUTH, FORCE, CREDIT, etc, then the VOID will not perform a FULL REVERSAL and will remain a batch database operation.

Note that support for FULL REVERSALS to release held authorization amounts is in a per-processor basis and may or may not be supported by the card issuers.

AUTOMATIC RECEIPTS VIA EMAIL or SMS

The SLIM CD Gateway can automatically format and send receipts to customers via EMAIL or SMS-Text. This feature is available for all credit card transactions. To utilize this feature, simply set the "send_email", "send_cc" or "send_sms" fields to "yes".

send_email – instructs the gateway to send an email-based receipt to the email address provided in the email field.

send_cc – instructs the gateway to send a receipt to the merchant based on either the email address on file for the merchant's profile, or the email address specified in the "cc_email" field.

send_sms – instructs the gateway to send a receipt to a mobile phone via SMS-Text, using the value of the "phone" field.

cc_email – this optional field overrides the email address stored in the merchant's profile when sending emails to cardholers and as the destination address when requesting a carbon-copy of an email.

The above flags cause the SLIM CD servers and systems to be used to send emails or sms-texts. Using these flags will cause additional billable transaction(s) to be created within the SLIM CD gateway. You can use one, two or all three send_xxx flags on a single credit card transaction, however a maximum of only one EMAIL and one SMS transaction will be created.

You may send the merchant an email using the "send_cc" value without having to send the cardholder an original receipt. This allows you to notify merchants of purchases easily. Alternatively, you can just send the cardholder a receipt. Or, you can send a receipt to both the cardholder and the merchant. In any of the above cases, a single EMAIL transaction will be created in the gateway.

SLIM CD uses various SMS text gateways and partners to perform the SMS-Text function. The rules for sending SMS-Text from computers are changing. As a result, SLIM CD cannot guarantee that text messages will be delivered in a timely manner. At times, SLIM CD may need to substitute email-to-sms functionality for true SMS-Texting. If this is done, please note that not all wireless carriers deliver email-to-sms messages in a timely manner and SLIM CD provides no guarantees to the timeliness or reliability of SMS services.

WEB SERVICE FOR RECEIPTS VIA EMAIL or SMS

A web function is also available so that a receipt can be emailed independently of a transaction by performing a POST to the following URL: http://stats.slimcd.com/soft/email_sms.asp

This routine requires clientid, password, and gateid. The optional control fields are send_email, send_cc, and send_sms. To use these, you can also supply email, phone and cc_email. These fields operate as defined above.

The ASP page returns an XML document in the standard format, with a <datablock> containing 3 values for <email_status>

Sample post data: (this sample just sends an email)

Clientid=1032&password=289075&gateid=18421874&send_email=yes&email=you@test.com

Sample reply:

```
<reply>
 <response>Success</response>
 <responsecode>0</responsecode>
 <description></description>
 <responseurl>/soft/email_sms.asp</responseurl>
 <datablock>
          <email_result>
                      <status>SENT </status>
                      <info>you@test.com</info>
          </email_result>

          <cc_result>
                      <status>NONE</status>
                      <info></info>
          </cc_result>

          <sms_result>
                      <status>NONE</status>
                      <info></info>
          </sms_result>
   </datablock>
 </reply>
```

The <status> fields can contain either "SENT",. "FAIL", or "NONE". The <info> fields echo back the destination upon success, are empty if NONE was requested, or contain descriptive error text if a FAIL status is returned.

Note that a single call to this service can send out an email, a cc, and an SMS-text, based on the options selected.

CVV2 - Card Code Value (also called CVC2 & CID)

The CVV2 name is named from Visa-branded credit cards. This data is called **CVC2** by MasterCard, and the four digits on the front of an American Express card are called the **CID**. This Gateway accepts the data, regardless of card brand, in the field named CVV2. For American Express values to be processed successfully the merchant must specifically enroll with American Express and the processor must be on American Express's "approved processor" list. Check with your processor to find out if they are certified with American Express to process CID. To enroll for American Express CID processing, the merchant must send an email to American express at: es.fraud.prevention@aexp.com and must undergo a security review.
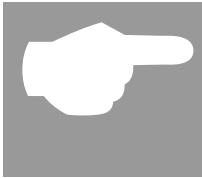
Please note that newer American Express cards now contain an additional set of 3 digits on the signature line of the card, but at this time these digits are not processed by the gateway software and can not be entered as a CVV2 value.

Processing of the CVV2 data only occurs for manually keyed (not swiped) transactions at this time.

The CVV2 field can contain the following data:

- **999** - where 999 is the 3 digits from the back of a Visa or MasterCard

- **9999** - where 9999 is the 4 digits from the front of an American Express card

- **N/A** - the 3 characters "N/A" indicate that the CVV2 value on the back of the Visa/MasterCard does exist, but is illegible

- **NONE** - the 4 characters "NONE" indicate there is no CVV2 value on the back of the Visa/MasterCard. This can only occur on older credit cards, most of which have been removed from circulation at this time.

To process the value of "NONE" successfully, the card must not have the CVV2 digits associated with it, or a CVV2 mismatch will occur. In other words, "NONE" will match only if no actual digits are on the back of the card. If the digits are there, or were there at one time, use the "N/A" value to indicate the CVV2 is illegible.

Test CVV2 Numbers for VITAL

For testing purposes when using VITAL as a processor, use the following CVV2 values:

- Visa Cards: 999

- Mastercards: 998

- Discover and Diner: 996

- Amex: 9997

Programming with the gateway can be a complex undertaking. In working with programmers, we've found the following information is misunderstood often enough to earn this section called programming "GOTCHAs"!

CONFIGURABLE SETTINGS

It is recommended that your software include the following configurable user settings:

- Domain (Gateway URL)

- ClientID

- SiteID

- PriceID

- Password

- Key (see next item)

KEYS

A separate "KEY" is required for EACH CLientID your software uses! (Developer's may obtain KEYs by:

1. First registering as a developer at the Slim CD WEB Site:

   http://www.slimcd.com/developers/(click on the "To signup as a developer…" link)

    …and then by calling Slim CD Toll-free at 1-877-475-4623

GATEID

The "gateid" is a UNIQUE number that is returned to you every time you send a transaction to the Gateway. Whenever you need to refer to a previous transaction (such as in a VOID, FORCE, OFFLINEFORCE, etc.), you simply use the "gateid" you received when you sent the first transaction (you don't have to "repeat" all the fields you previously sent.)

GATEID is:

1. Preferred for VOID, FORCE, OFFLINEFORCE

2. Optional on CREDIT, based on the merchant's profile

3. Can be sent for a "recurring charge" when performing a SALE, AUTH or OFFLINEAUTH with the "gateid" specified and a "recurring=yes". Just send the new "amount" to perform a new transaction on an old (existing) card.

Used for tokenization on a SALE transaction, which recalls all of the original cardholder data (cardnumber, name, address, invoice number, po, etc) and overrides any new data provided as part of this transaction (except cardnumber).

4. Returned by LOAD transactions that allow data to be stored on the SLIM CD servers without actually charging the card.

COMMON ERRORS

AUTH_LOOKUP_FAIL: One (or more) of your ID fields (clientid, siteid, priceid, password, key) is incorrect!

DUPLICATE_TRANSACTION_ALREADY_APPROVED:VITAL1:1241322:YM:
You sent the same transaction more than once!

- Where VITAL1 is the 6-character AUTHCODE

- 1241322 is the gateid

- Y is the AVS

- M is the CVV2

DUPLICATE_TRANSACTION_PENDING: You sent the same transaction more than once VERY quickly!

Servers and Interfaces

Server Domains

Slim CD Admin Web Services are accessible from two sets of web servers, test and production. A pair of servers exists within each set of servers. The pair of servers work together to provide online and reporting services. The online server is used to board merchants, change configuration, and process transactions. The reporting servers are primarily used for data-intensive report generation, but can be used as a source for pulling merchant information with the caveat that the data on the reporting server may not be as up to date as the data on the transaction server.

- **Production Domain and Protocol**

  The production WEB Services are located in the subdirectories of each domain as follows:

  Transactions: https://trans.slimcd.com/wswebservices/transact.asmx

  Reports: https://stats.slimcd.com/wswebservices/reports.asmx

  Images: https://stats.slimcd.com/wswebservices/images.asmx

- **Test Domain and Protocol**

"Test" transactions are done using the production servers (above) with test accounts (See appendix "C")

Note that these servers **must** be accessed using SSL (**https**), as the normal http protocol is not recommended for use with any of these systems for PCI Compliance reasons..

All information contained in this document is designed for the production servers. It is expected that all transaction or merchant boarding/updating will be performed on the "stats" servers. Do not update information on the reporting servers, as it will not be effective for real-time processing. In addition, do not expect the information on the reporting servers to be updated in real-time. Get all merchant setup or configuration information directly from the transaction servers.

To use the test servers, simply change the examples contained in this document to point to the test environment systems instead of the production environment systems.

The following interfaces are supported by the Slim CD web services:

Name/Value Pairs

To use this interface a name/value pair query string is created and sent via POST/GET to the Slim CD web services.

The name/value pairs are posted to:

For testing and production:

https://trans.slimcd.com/wswebservices/transact.asmx/PostHTTP

The **entire name/value pair string** should be posted in a field called PostData. You must first build your normal string for HTTP as below: (Note that normal URL-Encoding would apply for the data, which is shown with the plus-signs for spaces.)

This data below is name/value pairs with url-encoding:

clientid=1035&siteid=655877458&priceid=77&password=234980&key=SVD-491-QMFQ-WU89&product=MAL&version=1.0&transtype=SALE&first_name=first&last_name=last&address=123+somewhere+street&city=Coral+Springs&state=FL&zip=33076&amount=1.00&cardnumber=4444555566669999&expmonth=09&expyear=08

Then, you would URL-Encode the entire string again and place it in the "postdata" value. This way, we can have a Microsoft web service work to accept a growing number of name-value pairs without breaking the web service. An example of a credit card charge using a GET would look like the following:

REQUEST (GET):

https://trans.slimcd.com/wswebservices/transact.asmx/PostHTTP?postdata=clientid%3D1035%26siteid%3D655877458%26priceid%3D77%26password%3D234980%26key%3DSVD-491-QMFQ-WU89%26product%3DMAL%26version%3D1.0%26transtype%3DSALE%26first_name%3Dfirst%26last_name%3Dlast%26address%3D123+somewhere+street%26city%3DCoral+Springs%26state%3DFL%26zip%3D33076%26amount%3D1.00%26cardnumber%3D4444555566669999%26expmonth%3D09%26expyear%3D08

REPLY **(for field details, see appendix "B"):**

response=Success&responsecode=0&description=OK&responseurl=OCC_DEMO/wswebserv
ices/transact.aspx&authcode=VITAL9&approved=Y&cvv2reply=&avsreply=N&gateid=44173

NOTE:

The name/value pair interface (POST/GET) requires that the input parameters be
URL-encoded per RFC 17381738 (Note the PLUS signs in the address fields, above
instead of spaces!) The output returned from the transaction Gateway will also be
URL-encoded. See Appendix F for more on URL encoding.

Extensible Markup Language (XML)

This interface requires the developer to create an XML document. This XML document will contain all the appropriate elements and values needed to represent the transaction they wish to process. Once the XML document is created it is POSTED via HTTPS to the Slim CD web services.

XML data is posted to

> For testing & production: https://trans.slimcd.com/wswebservices/transact.asmx

If posted to the XMLPost webservice then the entire xml document should be posted as a string in a field called XMLData. An example of a gift card add funds transaction would look like the following:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Password, Ver, Product, Key, XMLData)
```

WHERE:

> Ver = the version of the software sending the transaction
> Product = the name of the software sending the transaction
> Key = the Slim CD "key" is the "developer's key" obtained from Slim CD, Inc. at 1-877-475-4623

The field "XMLData" is formatted as follows:

```
<request>
        <transtype>ADD</transtype>
        <cardnumber>1111222233334444555</cardnumber>
        <cardtype>GC</cardtype>
        <client_transref>1234</client_transref>
        <amount>50.00</amount>
        <first_name>Joe</first_name>
        <last_name>Tester</last_name>
        <address>123 somewhere street</address>
        <city>Some City</city>
        <state>FL</state>
        <zip>33076</zip>
        <country>US</country>
        <phone>800-555-1212</phone>
</request>
```

REPLY: (for field details, see appendix "B")

```
<reply>
    <responsecode>0</responsecode>
    <response>Success</response>
    <description>OK</description>
    <responseurl>https://trans.slimcd.com/wswebservices/transact.asmx/PostXML</responseurl>
    <datablock>
        <authcode>Y</authcode>
        <gateid>46256</gateid>
        <bal>850.00</bal>
    </datablock>
</reply>
```

Simple Object Access Protocol (SOAP)

To use this interface, a SOAP object is first created on a local machine. Pointing it to the Slim CD web services then initializes the SOAP object. Then, the appropriate transaction type is invoked and the return values are read from the output xml. All parameters must be sent when the transaction type is called, due to a limitation in the SOAP specification not allowing optional parameters.

The Slim CD web services can be found at:

For testing & production: https://trans.slimcd.com/wswebservices/transact.asmx

> **NOTE:**
>
> If you are developing using Microsoft .NET, simply add a "WEB Reference" to your project using https://trans.slimcd.com/wswebservices/transact.asmx for the URL of the service transaction service and https//trans.slimcd.com/wswebservices/reports.asmx for the reports.. The creation of the SOAP and XML are done automatically.

An example of a SOAP call to the GiftCardAdd transaction type using Microsoft Visual Basic .net would look like the following:

Add a web reference to the Slim CD webservice.

Instantiate the web service:

```
Dim objSlimCD As WebReference.Transact
Dim ReturnXMLNode As New Xml.XmlNode
```

Call the web service:

```
ReturnXMLNode = objSlimCD.PostXML(ClientID, SiteID, PriceID, Password, Ver, Product, Key,
XMLData)

MsgBox("Approved = " &
ReturnXMLNode.SelectSingleNode("/reply/datablock/authcode").Value.ToString)
```

All SOAP calls return an XML document node containing the server's response to your request:

RETURN STRUCTURE

All methods return responses in the following XML data structure format:

```
<reply>
      <responsecode>[contains 0 for success, non-zero for failure]</responsecode>
      <response>[text result]</response>
      <description>text description or message about the result]</description>
      <responseurl>[complete calling path of the method returning this reply]</responseurl>
```

```
        <datablock>
              [block containing returned data, if any and in the format of each method]
        </datablock>
</reply>
```
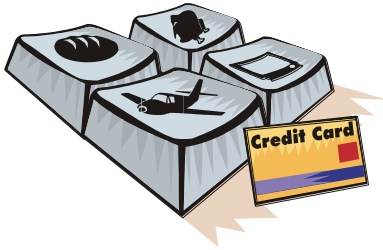
Transaction Types

The following transaction types are available through all 3 interfaces to the Slim CD web service:

SALE – Charge a card and place it in the batch for settlement.

CREDIT – Refunds a cardholder by returning funds from a merchant.

AUTH – Authorize an amount from the card (placing a hold on the funds), but not transferring the funds to the merchant. Note that $0.00 auths are permitted on many processors as a way of validating the CVV2 and Address information.

FORCE – Takes a previous AUTH and places it into the current open batch so the merchant is funded for the transaction.

VOID – On a SALE, prevents the transaction from being funded to the merchant and releases the hold on the funds (when possible). On an AUTH, prevents the transaction from being FORCED, removed it from the OPEN BATCH report and releases the hold on the funds (when possible), on a CREDIT, removes it from the current batch so the cardholder does not receive the refund.

LOAD – Accepts data from the merchant and stores it on the SLIM CD servers for tokenization purposes.

UPDATE – Updates an existing transaction's SALESTAX information so the merchant gets a better rate from VISA/MC

TIPEDIT – Adjusts the amount of the sale and/or the amount of the tip.

OFFLINE – Adds a transaction to the open batch using a supplied authorization code (does not go online and place an additional authorization on the card)

OFFLINEAUTH – Places an AUTH in the SLIM CD system without going online and authorizing the card (similar to an OFFLINE, without placing the transaction in the open batch)

OFFLINEFORCE – Places an OFFLINEAUTH in the open batch.

INCREMENTAL – Increments a previous AUTH before performing a FORCE so the authorized amount can match the final amount to be FORCEd.

REVERSAL – Reverses a previous AUTH before performing a FORCE so the authorized amount can match the final amount to be FORCEd . Note that reversal for the full amount can be performed on any card type if the developer wishes to release the hold on funds from a previous AUTH or SALE, but this is functionality is performed automatically when a VOID is used.

ACTIVATE – Activates a gift card

ADD VALUE – Adds funds to a gift card

DEACTIVATE – Deactivates a gift card (with and without a cash-back refund)

BALANCE – Returns the balance on a gift card

The Credit Card Sale transaction type authorizes a credit card transaction and places it in the current open BATCH. Funds are transferred to the merchant when the current open BATCH is "closed" (typically after the close of the business day.)

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

NOTE

You may see (in various places in the documentation) a field referred to as "gateid", "BatchID", "slimid" or "ticketnumber". These terms *ALL* refer to the same field (generally known as "gateid"), a unique identifying number the Gateway applies to all transaction records, identifying each record.

```
EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Password, Ver, Product, Key, XMLData)

WHERE field "XMLData" is formatted similar to (for a swiped card):

<request>
     <transtype>SALE</transtype>
     <trackdata>%B5454545454545454^LAST/FIRST
       ^05121015432112345678?;5454545454545454=05121015432112345678?</trac
       kdata>³
     <cardpresent>yes</cardpresent>³
     <readerpresent>yes</readerpresent>³
     <amount>50.00</amount>
</request>

OR…( for a manually keyed-in card):

<request>
     <transtype>SALE</transtype>
     <cardnumber>1111222233334444555</cardnumber>
     <expmonth>08</expmonth>
     <expyear>2006</expyear>
     <amount>50.00</amount>
```

---

[1] Slim CD requires cardholder data. That data can be either direct cardholder data or tokenized data.

    A.   Direct cardholder data:
          a.   Trackdata. Can be from a regular or encrypting reader.
          b.   Cardnumber, expmonth and expyear

    B.   Tokenized data:
          a.   Gateid. Note: If expmonth/expyear are sent, they will override what is stored with the gateid
          b.   Processor_token expmonth and expyear
          c.   Transient_token. Note: Any data not supplied when creating a transient token must also be included.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

```
</request>
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[2]	SDK developer key obtained from Slim CD, Inc.[2]
*** Required Fields ***			
transtype	18	Yes	SALE
amount	12	Yes	Total Amount (USD)
*** Conditionally Required Fields *** (see below)			
trackdata	varies	Recommended[1]	Card swipe track data, or encrypted track data (See Appendix F)
cardnumber	20	Sometimes[1]	Card number – used for verification (if swiped, "trackdata is used instead)
expmonth	2	Sometimes[1]	Expiration month (MM) (if swiped, trackdata is used instead)
expyear	4	Sometimes[1]	Exp year (YYYY or YY) (if swiped, trackdata is used instead)
gateid	12	Optional[1]	Gateid returned from previous transaction
processor_token	64	Optional[1]	Returned by processor from previous transaction, if available
transient_token	40	Opional[1]	Retured from call to QueueData (usually from JSON/JavaScript)
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number – Required for ecommerce
email	40	No	Cardholder Email address – Required for ecommerce
CVV2	4	No	CVV2
clientip	20	No	IP of requesting client or person
cardpresent	3	No	Contains the value "yes" or "no"
quasicash	3	No	Contains the value "yes" for quasi-cash
readerpresent	3	No	Contains the value "yes" or "no" for card reader at the location

[1]Slim CD requires cardholder data. That data can be either direct cardholder data or tokenized data.

    A. Direct cardholder data:
        a. Trackdata. Can be from a regular or encrypting reader.
        b. Cardnumber, expmonth and expyear

    B. Tokenized data:
        a. Gateid. Note: If expmonth/expyear are sent, they will override what is stored with the gateid
        b. Processor_token expmonth and expyear
        c. Transient_token. Note: Any data not supplied when creating a transient token must also be included.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
authcode	6	No	Auth code received for Offline Transactions
allow_duplicates	3	No	Contains the value "yes" or "no", default is "no"
duplicates_window	6	No	Specifies the time(in minutes) to look back for a matching (duplicate) transaction. The default is 30, the maximum is 20000. This is only used when allow_duplicates is "no"
corporatecard	3	No	Contains the value "yes" or "no"
recurring	3	No	Contains the value "yes" or "no"
installmentcount	8	No	Number of times to recharge
installmentseqno	8	No	Sequence number
allow_partial	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
billpayment	3	No	Contains the value "yes" or "no". Yes causes a billpayment transaction type to be used for retail, mail/phone and ecommerce merchants.
debtrepayment	3	No	Contains the value "yes" or "no". Yes causes a debt repayment indicator for billpayment transactions (good for check cards only)
quasicash	3	No	Contains the value "yes" or "no". Yes causes a quasi-cash transaction type
deviceid	8	No	Up to 8 byte unique device id field for MasterCard
*** Varied Requirement Fields ***			
ebt	4	Varies	FS, CD, or CASH – Must specify for EBT transactions
retrievalrefno	6	Varies	EBT transactions only
ebt_transid	6	Varies	EBT transactions only
auditno	6	Varies	EBT transactions only
voucherno	6	Varies	EBT transactions only
Restaurant Specific			
gratuity	12	No	amount of the charge that represents tip
Ecommerce Specific			
egoodsindicator	2	No	"P" for physical goods, "D" for digital goods, or "PD" for a combination of both
Direct Marketing Specific			
shipdate	4	No	date in MMDD format
Auto Rental Specific			
extracharge	6	No	0=None, 1=Gasoline, 2=Mileage, 3=Late Return, 4=One way service fee, 5=Driving violation *The data should be left-justified and zero padded to 6 character positions
noshow	1	No	Indicates charges for a reservation that caused a non-cancellation penalty: 0 = None, 1 = No Show
rentaldate	6	No	YYMMDD format, the date the cardholder first drove away with rental
rentaltime	4	No	HHMM format, in 24 hour format
*** Varied Requirement Fields ***			
rentername	20	No	Name of car renter (Mastercard)
returncity	18	No	Name of city which the car is returned (Mastercard)

[1]Slim CD requires cardholder data. That data can be either direct cardholder data or tokenized data.

    A.   Direct cardholder data:
        a.   Trackdata. Can be from a regular or encrypting reader.
        b.   Cardnumber, expmonth and expyear

    B.   Tokenized data:
        a.   Gateid. Note: If expmonth/expyear are sent, they will override what is stored with the gateid
        b.   Processor_token expmonth and expyear
        c.   Transient_token. Note: Any data not supplied when creating a transient token must also be included.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

returnstate	3	No	Name of state/country where rental was returned (Mastercard)
rentalreturndate	6	No	YYMMDD format, the date the cardholder returns rental
rentalreturntime	4	No	HHMM format, in 24 hour format
rentalagrmntnbr	10	No	Rental agreement number
rentalcity	30	No	City where rental originated
rentalstate	3	No	State where rental originated
rentalclassid	8	No	ClassID of rental
returnlocationid	10	No	Location of the return center (Mastercard)
vehicleclassid	4	No	TSYS autorental AMEX **see table below
rentaldistanceunit	1	No	M-miles K-kilometers
rentallocationdba	38	No	Auto rental pickup location, or agency's business name
rentaldistance	5	No	Auto rental distance
*** Varied Requirement Fields ***			
stayduration	2	varies	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number of days of prepayment the charge represents. In the case of a rental, this represents the number of days for the expected rental. This field should be right justified and zero filled to two digits.
Hotel Specific			
chargetype	1	Yes	1=Hotel, 2=Restaurant, 3=Gift Shop
extracharge	6	No	0 = None, 1=RESERVED, 2=Restaurant, 3=Gift Shop, 4=Mini-Bar, 5=Telephone, 6=Other, 7=Laundry *The data should be left justified and zero padded to 6 character positions
noshow	1	No	Indicates charges for a reservation that caused a non-cancellation penalty, or other special charges as follows: 0=None, 1=No Show, 3=Card Deposit, 4=Delayed Charge, 5=Express Service, 6=Assured Reservation
checkindate	6	No	YYMMDD format for the date of check in. In the case of a prepaid reservation, this date represents the date of the reservation. In the case of a noshow charge this would be the date the reservation was made
checkintime	4	No	Check in time as HHMM in 24 hour format (optional for some card processors)
checkoutdate	6	No	YYMMDD format date of the expected departure
checkouttime	4	No	Check out time as HHMM in 24 hour format (optional for some card processors)
roomrate	12	No	Amount of the daily rate for the room
roomamount	12	No	Amount of the room
roomtax	12	No	Tax amount for room
roomratetax	6	No	Tax rate for room (i.e. .05)
nonroomcharge	12	No	Non room charges such as mini bar
customercode	8	No	Internal customer id
roomnumber	6	No	Guest's room number
prestigiousind	1	No	A VISA flag indicating: space=Not participating, D=$500, B=$1000, S=$1500
*** Varied Requirement Fields ***			
stayduration	2	varies	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number days of prepayment the charge represents.

[1]Slim CD requires cardholder data. That data can be either direct cardholder data or tokenized data.

    A.   Direct cardholder data:
         a.   Trackdata. Can be from a regular or encrypting reader.
         b.   Cardnumber, expmonth and expyear

    B.   Tokenized data:
         a.   Gateid. Note: If expmonth/expyear are sent, they will override what is stored with the gateid
         b.   Processor_token expmonth and expyear
         c.   Transient_token. Note: Any data not supplied when creating a transient token must also be included.

[2]Obtain from Slim CD, Inc. at 1-877-475-4623

			In case of check in, this represents the number of days for the expected stay. This field should be right justified and zero filled to two digits.
Debit Specific			
pinblockdata	32 or 36	Yes	Contains the encrypted pin data and ksn in one field. Note that this is processor-specific and should be replaced with ksn and pinblock as separate fields. (we recommend using pinblock and ksn instead of pinblockdata)
pinblock	16		Contains the encrypted pin data, used when sending pin and ksn separately
ksn	16 to 20	Yes	Contains the "Key Serial Number"
cashback	12	No	This contains the amount paid as cash back to the consumer.
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no") Set to "onapproved" to send an e-mail only when the transaction is approved.
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no") Set to "onapproved" to send an e-mail only when the transaction is approved.
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)
send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Encryption Options			
encryption_device	20	No	Default is blank, which causes automatic detection based on input. Valid option list will change, but currently: IDTECH, POSX, MAGTEK. Note that all devices default to KB mode, but HID mode can be used if trackdata is sent as all HEX digits.
encryption_type	20	No	Default is blank, which defaults to TDES. Valid option list will change, but currently: TDES or AES
encryption_key	20	No	Indicator of the key injected in the device. Default is blank, where key is determined from indicators in the trackdata.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Slim CD requires cardholder data. That data can be either direct cardholder data or tokenized data.

   A.   Direct cardholder data:
        a.   Trackdata. Can be from a regular or encrypting reader.
        b.   Cardnumber, expmonth and expyear

   B.   Tokenized data:
        a.   Gateid. Note: If expmonth/expyear are sent, they will override what is stored with the gateid
        b.   Processor_token expmonth and expyear
        c.   Transient_token. Note: Any data not supplied when creating a transient token must also be included.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Note that developers using the COM object can convert a SALE transaction into an OFFLINE transaction simply by providing the AUTH CODE data as part of the transaction.

Also note that **only** the developers using the COM object will have the gratuity added to the total to produce the amount charged to the card. Otherwise, the amount provided is the total amount to charge the card.

Vehicleclassid **for TSYS autorental AMEX**

Code Vehicle class	Code Vehicle class
0001 Mini	0018 24-foot truck
0002 Subcompact	0019 26-foot truck
0003 Economy	0020 Moped
0004 Compact	0021 Stretch
0005 Midsize	0022 Regular
0006 Intermediate	0023 Unique
0007 Standard	0024 Exotic
0008 Full size	0025 Small/medium truck
0009 Luxury	0026 Large truck
0010 Premium	0027 Small SUV
0011 Minivan	0028 Medium SUV
0012 12-Passenger van	0029 Large SUV
0013 Moving van	0030 Exotic SUV
0014 15-Passing van	0031 Four wheel drive
0015 Cargo van	0032 Special
0016 12-foot truck	9999 Miscellaneous
0017 20-foot truck	4spaces Unused

NOTE: Tokenization can be performed by providing the GATEID of an originally approved SALE or LOAD transaction. Cardholder data (name, address, phone, cardnumber) from the original transaction will be used to perform the new transaction. Any supplied fields (except cardnumber) will override data on file.

[1]Slim CD requires cardholder data. That data can be either direct cardholder data or tokenized data.

    A.   Direct cardholder data:
        a.   Trackdata. Can be from a regular or encrypting reader.
        b.   Cardnumber, expmonth and expyear

    B.   Tokenized data:
        a.   Gateid. Note: If expmonth/expyear are sent, they will override what is stored with the gateid
        b.   Processor_token expmonth and expyear
        c.   Transient_token. Note: Any data not supplied when creating a transient token must also be included.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Credit Card Void

The Credit Card Void transaction type is used to void a previous SALE (or FORCE, described later) transaction before the current open BATCH is closed. (This transaction type removes the transaction from the "BATCH" before the BATCH is processed. If the BATCH has already been "Closed", then you must use a "CREDIT CARD CREDIT" instead to refund the purchase.) Note that our implementation for some processors will now perform a FULL REVERSAL to release held funds if the original transaction being VOIDed is an AUTH or SALE.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[2]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
Transtype	18	Yes	VOID
*** Conditionally Required Fields *** (see CREDIT CARD SALE)			
gateid	12	Recommended[1]	Gateid returned from previous transaction
trackdata	varies	Optional[1]	Card swipe track data, or encrypted track data (See Appendix F)
cardnumber	20	Sometimes[1]	Card number – Will be used to search for a previous SALE, FORCE, or OFFLINEFORCE if a gateid is not provided. Will be used to search for a CREDIT if amount is provided and is negative.
expmonth	2	Sometimes[1]	Expiration month (MM) (if swiped, trackdata is used instead)
expyear	4	Sometimes[1]	Exp year (YYYY or YY) (if swiped, trackdata is used instead)
processor_token	64	Optional[1]	Returned from processor on a previous transaction, if available
transient_token	40	Opional[1]	Returned from call to QueueData (usually from JSON/JavaScript)
*** Optional Fields ***			
amount	12	No	(a) Will be used to search for a previous SALE, FORCE, or OFFLINEFORCE if gateid not specified and amount is positive. (b)Will be used to search for a CREDIT if amount is negative. The value must match the amount of the original transaction.
client_transref	20	No	Will be used to search for a previous SALE, FORCE, or OFFLINEFORCE, if gateid is not provided. If provided, it must match original transaction's client_transref to find the original transaction.

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

match_window	6	No	Time to look back, in minutes, for a transaction when gateid is not specified and cardnumber/client_transref are specified.
Username	20	No	Clerk/Salesperson Name/ID
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no")
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no")
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)
send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

The VOID transaction has been enhanced. **It no longer requires a Gateid, although use of a gateid is still STRONGLY recommended as the preferred way to exactly identify the transaction to VOID.** Please see the section on VOID/FORCE – Enhancements for more information.

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Credit Card Credit

The Credit Card Credit (a refund) transaction type credits a credit card account with a specified amount. (This transaction type would be used AFTER the original sale had been processed (closed) in a previous BATCH. To do a credit (refund) while the transaction is still in the current BATCH, use a CREDIT CARD VOID instead.)

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

```
EXAMPLE: ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	CREDIT
amount	12	Yes	Amount to credit (USD)
*** Conditionally Required Fields *** (see CREDIT CARD SALE)			
gateid	12	Recommended[1]	Gateid returned from previous transaction
trackdata	varies	Optional[1]	Card swipe track data, or encrypted track data (See Appendix F)
cardnumber	20	Sometimes[1]	Card number – Will be used to search for a previous SALE/FORCE, if a gateid is not provided
expmonth	2	Sometimes[1]	Expiration month (MM) (if swiped, trackdata is used instead)
expyear	4	Sometimes[1]	Exp year (YYYY or YY) (if swiped, trackdata is used instead)
processor_token	64	Optional[1]	Returned by processor from previous transaction, if available
transient_token	40	Opional[1]	Returned from call to QueueData (usually from JSON/JavaScript)
*** Optional Fields ***			
cardpresent	3	No[2]	Contains the value "yes" or "no" (must be "yes" for PIN DEBIT)
readerpresent	3	No[2]	Contains the value "yes" or "no" for a card reader at the merchant's location (must be "yes" for DEBIT Refund)
blind_credit	3	No	Allow a credit without a previous sale – set this to yes/no
authcode	6	No	Auth code received for Offline Transactions
extra_credit	3	No	Contains the value "yes" or "no", where yes allows more than the original amount to be refunded (useful for tokenization)
username	20	No	Clerk/Salesperson Name/ID

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

CREDIT CARD CREDIT

trackdata	varies	varies	Card swipe track data (Required for DEBIT Refund)
*** Debit Specific ***			
pinblockdata	32	Yes	Contains the KSN and pinblock as one field, and is processor-specific (it us recommended to use pinblcck and ksn separately)
pinblock	16	Yes	Contains the encrypted pin data (use instead of "pinblockdata")
ksn	16 to 20	Yes	Contains the "Key Serial Number"
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no")
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no")
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)
send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Encryption Options			
encryption_device	20	No	Default is blank, which causes automatic detection based on input. Valid option list will change, but currently: IDTECH, POSX, MAGTEK. Note that all devices default to KB mode, but HID mode can be used if trackdata is sent as all HEX digits.
encryption_type	20	No	Default is blank, which defaults to TDES. Valid option list will change, but currently: TDES or AES
encryption_key	20	No	Indicator of the key injected in the device. Default is blank, where key is determined from indicators in the trackdata.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Page **56 of 170**

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

The Credit Card Auth transaction type authorizes a credit card transaction; however, no actual charge is made to the consumer's credit card. An "Auth" is basically the first part of a "two-part" sale. The "Auth" merely checks with the bank to see if the customer has the capability to charge the probable sale amount. (No money is transferred to the merchant yet. Only the customer's "open to buy" balance (or, available credit) is reduced for a predetermined amount of time.) To place the "AUTH" transaction in the open BATCH, the second part (which is required) is a "FORCE" to place the previous Auth into the current open BATCH to be processed. It then becomes the same as a regular SALE.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

```
EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[2]	SDK developer key obtained from Slim CD, Inc.[2]
*** Required Fields ***			
transtype	18	Yes	AUTH
amount	12	Yes	Amount (USD)
*** Conditionally Required Fields *** (see CREDIT CARD SALE)			
trackdata	varies	Recommended[1]	Card swipe track data, or encrypted track data (See Appendix F)
cardnumber	20	Sometimes [1]	Card number – used for verification (if swiped, "trackdata is used instead)
expmonth	2	Sometimes [1]	Expiration month (MM) (if swiped, trackdata is used instead)
expyear	4	Sometimes [1]	Exp year (YYYY or YY) (if swiped, trackdata is used instead)
gateid	12	Optional[1]	Gateid returned from previous transaction
processor_token	64	Optional[1]	Returned by processor from previous transaction, if available
transient_token	40	Opional[1]	Returned from call to QueueData (usually from JSON/JavaScript)
*** Optional Fields ***			
first_name	20	No[1]	First name of cardholder
last_name	20	No[1]	Last name of cardholder
address	30	No[1]	Cardholder address
city	20	No[1]	City

[1] See the comments for CREDIT/DEBIT CARD SALE (ABOVE)

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

state	20	No[1]	State
zip	10	No[1]	Zip code
country	30	No[1]	Country
phone	20	No[1]	Cardholder phone number – Required for ecommerce
email	40	No[2]	Cardholder Email address – Required for ecommerce
CVV2	4	No	CVV2
clientip	20	No	IP of requesting client or person
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No[3]	Contains the value "yes" or "no"
quasicash	3	No	Contains the value "yes" for quasi-cash
readerpresent	3	No[3]	Contains the value "yes" or "no" for a card reader at the merchant's location
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
authcode	6	No	Auth code received for Offline Transactions
allow_duplicates	3	No	Contains the value "yes" or "no"
corporatecard	3	No	Contains the value "yes" or "no"
Allow_partial	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
billpayment	3	No	Contains the value "yes" or "no". Yes causes a billpayment transaction type to be used for retail, mail/phone and ecommerce merchants.
debtrepayment	3	No	Contains the value "yes" or "no". Yes causes a debt repayment indicator for billpayment transactions (good for check cards only)
quasicash	3	No	Contains the value "yes" or "no". Yes causes a quasi-cash transaction type
*** Varied Requirement Fields ***			
ebt	4	Varies	FS, CD, or CASH – Must specify for EBT transactions
retrievalrefno	6	Varies	EBT transactions only
ebt_transid	6	Varies	EBT transactions only
auditno	6	Varies	EBT transactions only
voucherno	6	Varies	EBT transactions only
Restaurant Specific			
gratuity	12	No	amount of the charge that represents tip
Ecommerce Specific			
egoodsindicator	2	No	"P" for physical goods, "D" for digital goods, or "PD" for a combination of both
Direct Marketing Specific			
shipdate	4	No	date in MMDD format
Auto Rental Specific			
extracharge	6	No	0=None, 1=Gasoline, 2=Mileage, 3=Late Return, 4=One way service fee, 5=Driving violation *The data should be left-justified and zero padded to 6 character positions
noshow	1	No	Indicates charges for a reservation that caused a non-cancellation penalty: 0 = None, 1 = No Show
rentaldate	6	No	YYMMDD format, the date the cardholder first drove away with rental
rentaltime	4	No	HHMM format, in 24 hour format
rentername	20	No	Name of car renter (Mastercard)
returncity	18	No	Name of city which the car is returned (Mastercard)
returnstate	3	No	Name of state/country where rental was returned (Mastercard)
rentalreturndate	6	No	YYMMDD format, the date the cardholder returns rental

[1] See the comments for CREDIT/DEBIT CARD SALE (ABOVE)

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

rentalreturntime	4	No	HHMM format, in 24 hour format
rentalagrmntnbr	10	No	Rental agreement number
rentalcity	30	No	City where rental originated
rentalstate	3	No	State where rental originated
rentalclassid	8	No	ClassID of rental
returnlocationid	10	No	Location of the return center (Mastercard)
*** Varied Requirement Fields ***			
stayduration	2	varies	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number of days of prepayment the charge represents. In the case of a rental, this represents the number of days for the expected rental. This field should be right justified and zero filled to two digits.
Hotel Specific			
chargetype	1	Yes	1=Hotel, 2=Restaurant, 3=Gift Shop
extracharge	6	No	0 = None, 1=RESERVED, 2=Restaurant, 3=Gift Shop, 4=Mini-Bar, 5=Telephone, 6=Other, 7=Laundry *The data should be left justified and zero padded to 6 character positions
noshow	1	No	Indicates charges for a reservation that caused a non-cancellation penalty, or other special charges as follows: 0=None, 1=No Show, 3=Card Deposit, 4=Delayed Charge, 5=Express Service, 6=Assured Reservation
checkindate	6	No	YYMMDD format for the date of check in. In the case of a prepaid reservation, this date represents the date of the reservation. In the case of a noshow charge this would be the date the reservation was made
checkintime	4	No	Check in time as HHMM in 24 hour format (optional for some card processors)
checkoutdate	6	No	YYMMDD format date of the expected departure
checkouttime	4	No	Check out time as HHMM in 24 hour format (optional for some card processors)
roomrate	12	No	Amount of the daily rate for the room
roomamount	12	No	Amount of the room
roomtax	12	No	Tax amount for room
roomratetax	6	No	Tax rate for room (i.e. .05)
nonroomcharge	12	No	Non room charges such as mini bar
customercode	8	No	Internal customer id
roomnumber	6	No	Guest's room number
prestigiousind	1	No	A VISA flag indicating: space=Not participating, D=$500, B=$1000, S=$1500
*** Varied Requirement Fields ***			
stayduration	2	varies	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number days of prepayment the charge represents. In case of check in, this represents the number of days for the expected stay. This field should be right justified and zero filled to two digits.
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no")
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no")
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)
send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Encryption Options			

[1] See the comments for CREDIT/DEBIT CARD SALE (ABOVE)

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

encryption_device	20	No	Default is blank, which causes automatic detection based on input. Valid option list will change, but currently: IDTECH, POSX, MAGTEK. Note that all devices default to KB mode, but HID mode can be used if trackdata is sent as all HEX digits.
encryption_type	20	No	Default is blank, which defaults to TDES. Valid option list will change, but currently: TDES or AES
encryption_key	20	No	Indicator of the key injected in the device. Default is blank, where key is determined from indicators in the trackdata.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Note that developers using the COM object can convert an AUTH transaction into an OFFLINEAUTH transaction simply by providing the AUTH CODE data as part of the transaction.

[1] See the comments for CREDIT/DEBIT CARD SALE (ABOVE)

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

The Credit Card Force transaction type charges a credit card after an authorization has already been obtained by a previous authorization transaction.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
Clientid	10	Yes	Assigned by the Slim CD administrator.
Siteid	10	Yes	Assigned by the Slim CD administrator.
Priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
Ver	16	No	Version ID of product
Product	32	No	Product Name (Application name)
Key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[4]
*** Required Fields ***			
transtype	18	Yes	FORCE
Amount	12	Yes	Final amount (including tip) (USD)
*** Conditionally Required Fields *** (see CREDIT CARD SALE)			
gateid	12	Recommended[1]	Gateid returned from previous transaction
trackdata	varies	Optional[1]	Card swipe track data, or encrypted track data (See Appendix F)
cardnumber	20	Sometimes[1]	Card number – Will be used to search for a previous AUTH, if a gateid is not provided
expmonth	2	Sometimes[1]	Expiration month (MM) (if swiped, trackdata is used instead)
expyear	4	Sometimes[1]	Exp year (YYYY or YY) (if swiped, trackdata is used instead)
processor_token	64	Optional[1]	Returned by processor from previous transaction, if available
transient_token	40	Opional[1]	Returned from call to QueueData (usually from JSON/JavaScript)
*** Optional Fields ***			
allow_duplicates	3	No	Contains the value "yes" or "no"
duplicates_window	6	No	Specifies the time, in minutes, to look back for a duplicate when allow_duplicates is set to "no". The default is 30 minutes.
client_transref	20	No	Will be used to search for a previous AUTH, if gateid is not provided. If provided, it must match original AUTH's client_transref to find the original transaction.
match_window	6	No	Time to look back, in minutes, for an AUTH or OFFLINEAUTH when gateid is not specified and cardnumber/client_transref are specified.
username	20	No	Clerk/Salesperson Name/ID
Restaurant Specific			

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Gratuity	12	No	amount of the charge that represents tip
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no")
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no")
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)
send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Note that developers using the COM object can convert a FORCE transaction into an OFFLINEFORCE transaction simply by providing the AUTH CODE data as part of the transaction.

Also note that **only** the developers using the COM object will have the gratuity added to the total to produce the amount charged to the card. Otherwise, the amount provided is the total amount to charge the card.

The FORCE transaction has been enhanced**. It no longer requires a Gateid, although use of a gateid is still STRONGLY recommended as the preferred way to exactly identify the transaction to FORCE.** Please see the section on VOID/FORCE – Enhancements for more information.

Also note that the FORCE command can now be used on either an AUTH or OFFLINEAUTH, and the system will automatically convert the FORCE to an OFFLINEFORCE if needed.

---

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

The Tip Edit transaction type charges a credit card after a "SALE" or a "FORCE" has already been sent into the BATCH, but the BATCH is still open. (This is typically used to change a gratuity amount after the "SALE" or "FORCE", but before the transaction is processed by closing the BATCH..)

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

```
EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[2]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	TIPEDIT
amount	12	Yes	Final amount (USD)
gratuity	12	Yes	Portion of the amount that represents the tip
*** Conditionally Required Fields *** (see CREDIT CARD SALE)			
gateid	12	Recommended[1]	Gateid returned from previous transaction
trackdata	varies	Optional[1]	Card swipe track data, or encrypted track data (See Appendix F)
cardnumber	20	Sometimes[1]	Card number – Will be used to search for a previous AUTH/SALE, if a gateid is not provided
expmonth	2	Sometimes[1]	Expiration month (MM) (if swiped, trackdata is used instead)
expyear	4	Sometimes[1]	Exp year (YYYY or YY) (if swiped, trackdata is used instead)
processor_token	64	Optional[1]	Returned by processor from previous transaction, if available
transient_token	40	Opional[1]	Returned from call to QueueData (usually from JSON/JavaScript)
*** Optional Fields ***			
username	20	No	Clerk/Salesperson Name/ID
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no")
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no")
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

			the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)
send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Note that **only** the developers using the COM object will have the gratuity added to the total to produce the amount charged to the card. Otherwise, the amount provided is the total amount to charge the card.

Also note that TIPEDIT can be used to adjust the final amount of any transaction, if supported by the processor. Set the gratuity to 0.00 and the amount to the new final amount to achieve this result.

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

The Credit Card Offline transaction type is the same as a SALE transaction except that a voice authorization is given and the authorization code is passed to the Gateway. (This is typically used when the normal processing is not available and authorizations must be acquired by telephone. After normal on-line processing is again available, this transaction is how you enter it into the system.)

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	OFFLINE
cardnumber	20	Yes	Card number – used for verification
expmonth	2	Yes	Expiration month (MM)
expyear	4	Yes	Exp year (YYYY or YY)
amount	12	Yes	Original amount (USD)
authcode	6	Yes	Auth code received for Offline Transactions
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number – Required for ecommerce
email	40	No	Cardholder Email address – Required for ecommerce
CVV2	4	No	CVV2
clientip	20	No	IP of requesting client or person
trackdata	varies	No	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

quasicash	3	No	Contains the value "yes" for quasi-cash
readerpresent	3	No	Contains the value "yes" or "no" for a card reader at the merchant's location
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
allow_duplicates	3	No	Contains the value "yes" or "no"
corporatecard	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
*** Varied Requirement Fields ***			
ebt	4	Varies	FS, CD, or CASH – Must specify for EBT transactions
retrievalrefno	6	Varies	EBT transactions only
ebt_transid	6	Varies	EBT transactions only
auditno	6	Varies	EBT transactions only
voucherno	6	Varies	EBT transactions only
Restaurant Specific			
gratuity	12	No	amount of the charge that represents tip
Ecommerce Specific			
egoodsindicator	2	No	"P" for physical goods, "D" for digital goods, or "PD" for a combination of both
Direct Marketing Specific			
shipdate	4	No	date in MMDD format
Auto Rental Specific			
extracharge	6	No	0=None, 1=Gasoline, 2=Mileage, 3=Late Return, 4=One way service fee, 5=Driving violation *The data should be left-justified and zero padded to 6 character positions
noshow	1	No	Indicates charges for a reservation that caused a non-cancellation penalty: 0 = None, 1 = No Show
rentaldate	6	No	YYMMDD format, the date the cardholder first drove away with rental
rentaltime	4	No	HHMM format, in 24 hour format
rentername	20	No	Name of car renter (Mastercard)
returncity	18	No	Name of city which the car is returned (Mastercard)
returnstate	3	No	Name of state/country where rental was returned (Mastercard)
rentalreturndate	6	No	YYMMDD format, the date the cardholder returns rental
rentalreturntime	4	No	HHMM format, in 24 hour format
rentalagrmntnbr	10	No	Rental agreement number
rentalcity	30	No	City where rental originated
rentalstate	3	No	State where rental originated
rentalclassid	8	No	ClassID of rental
returnlocationid	10	No	Location of the return center (Mastercard)
*** Varied Requirement Fields ***			
stayduration	2	varies	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number of days of prepayment the charge represents. In the case of a rental, this represents the number of days for the expected rental. This field should be right justified and zero filled to two digits.
Hotel Specific			
*** Required Fields ***			
chargetype	1	Yes	1=Hotel, 2=Restaurant, 3=Gift Shop
*** Optional Fields ***			
extracharge	6	No	0 = None, 1=RESERVED, 2=Restaurant, 3=Gift Shop, 4=Mini-Bar, 5=Telephone, 6=Other, 7=Laundry *The data should be left justified and zero padded to 6 character positions

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

noshow	1	No	Indicates charges for a reservation that caused a non-cancellation penalty, or other special charges as follows: 0=None, 1=No Show, 3=Card Deposit, 4=Delayed Charge, 5=Express Service, 6=Assured Reservation
checkindate	6	No	YYMMDD format for the date of check in. In the case of a prepaid reservation, this date represents the date of the reservation. In the case of a noshow charge this would be the date the reservation was made
checkintime	4	No	Check in time as HHMM in 24 hour format (optional for some card processors)
checkoutdate	6	No	YYMMDD format date of the expected departure
checkouttime	4	No	Check out time as HHMM in 24 hour format (optional for some card processors)
roomrate	12	No	Amount of the daily rate for the room
roomamount	12	No	Amount of the room
roomtax	12	No	Tax amount for room
roomratetax	6	No	Tax rate for room (i.e. .05)
nonroomcharge	12	No	Non room charges such as mini bar
customercode	8	No	Internal customer id
roomnumber	6	No	Guest's room number
prestigiousind	1	No	A VISA flag indicating: space=Not participating, D=$500, B=$1000, S=$1500
*** Varied Requirement Fields ***			
stayduration	2	varies	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number days of prepayment the charge represents. In case of check in, this represents the number of days for the expected stay. This field should be right justified and zero filled to two digits.
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no")
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no")
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)
send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

Credit Card Offline Auth

The Credit Card Offline Auth transaction type is the same as an AUTH transaction except that a voice authorization is given and the authorization code is passed to the Gateway. (This is typically used when the normal processing is not available and authorizations must be acquired by telephone. After normal on-line processing is again available, this transaction is how you enter an "Auth" into the system.)

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

```
EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	OFFLINEAUTH
cardnumber	20	Yes	Card number – used for verification
expmonth	2	Yes	Expiration month (MM)
expyear	4	Yes	Exp year (YYYY or YY)
amount	12	Yes	Original amount (USD)
authcode	6	Yes	Auth code received for Offline Transactions
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
CVV2	4	No	CVV2
clientip	20	No	IP of requesting client or person
trackdata	varies	No	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

quasicash	3	No	Contains the value "yes" for quasi-cash
readerpresent	3	No	Contains the value "yes" or "no" for a card reader at the merchant's location
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
allow_duplicates	3	No	Contains the value "yes" or "no"
corporatecard	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
*** Varied Requirement Fields ***			
ebt	4	Varies	FS, CD, or CASH – Must specify for EBT transactions
retrievalrefno	6	Varies	EBT transactions only
ebt_transid	6	Varies	EBT transactions only
auditno	6	Varies	EBT transactions only
voucherno	6	Varies	EBT transactions only
Restaurant Specific			
gratuity	12	No	amount of the charge that represents tip
Ecommerce Specific			
egoodsindicator	2	No	"P" for physical goods, "D" for digital goods, or "PD" for a combination of both
Direct Marketing Specific			
shipdate	4	No	date in MMDD format
Auto Rental Specific			
extracharge	6	No	0=None, 1=Gasoline, 2=Mileage, 3=Late Return, 4=One way service fee, 5=Driving violation *The data should be left-justified and zero padded to 6 character positions
noshow	1	No	Indicates charges for a reservation that caused a non-cancellation penalty: 0 = None, 1 = No Show
rentaldate	6	No	YYMMDD format, the date the cardholder first drove away with rental
rentaltime	4	No	HHMM format, in 24 hour format
rentername	20	No	Name of car renter (Mastercard)
returncity	18	No	Name of city which the car is returned (Mastercard)
returnstate	3	No	Name of state/country where rental was returned (Mastercard)
rentalreturndate	6	No	YYMMDD format, the date the cardholder returns rental
rentalreturntime	4	No	HHMM format, in 24 hour format
rentalagrmntnbr	10	No	Rental agreement number
rentalcity	30	No	City where rental originated
rentalstate	3	No	State where rental originated
rentalclassid	8	No	ClassID of rental
returnlocationid	10	No	Location of the return center (Mastercard)
*** Varied Requirement Fields ***			
stayduration	2	varies	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number of days of prepayment the charge represents. In the case of a rental, this represents the number of days for the expected rental. This field should be right justified and zero filled to two digits.
Hotel Specific			
*** Required Fields ***			
chargetype	1	Yes	1=Hotel, 2=Restaurant, 3=Gift Shop
*** Optional Fields ***			
extracharge	6	No	0 = None, 1=RESERVED, 2=Restaurant, 3=Gift Shop, 4=Mini-Bar,

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

			5=Telephone, 6=Other, 7=Laundry *The data should be left justified and zero padded to 6 character positions
noshow	1	No	Indicates charges for a reservation that caused a non-cancellation penalty, or other special charges as follows: 0=None, 1=No Show, 3=Card Deposit, 4=Delayed Charge, 5=Express Service, 6=Assured Reservation
check indate	6	No	YYMMDD format for the date of check in. In the case of a prepaid reservation, this date represents the date of the reservation. In the case of a noshow charge this would be the date the reservation was made
checkintime	4	No	Check in time as HHMM in 24 hour format (optional for some card processors)
checkoutdate	6	No	YYMMDD format date of the expected departure
checkouttime	4	No	Check out time as HHMM in 24 hour format (optional for some card processors)
roomrate	12	No	Amount of the daily rate for the room
roomamount	12	No	Amount of the room
roomtax	12	No	Tax amount for room
roomratetax	6	No	Tax rate for room (i.e. .05)
nonroomcharge	12	No	Non room charges such as mini bar
customercode	8	No	Internal customer id
roomnumber	6	No	Guest's room number
prestigiousind	1	No	A VISA flag indicating: space=Not participating, D=$500, B=$1000, S=$1500
*** Varied Requirement Fields ***			
stayduration	2	varies	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number days of prepayment the charge represents. In case of check in, this represents the number of days for the expected stay. This field should be right justified and zero filled to two digits.
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no")
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no")
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)
send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

The Credit Card Offline Force transaction type charges a credit card after a voice authorization has already been obtained by a previous OFFLINE AUTH transaction. (This is typically used when the normal processing is not available and authorizations must be acquired by telephone. After normal on-line processing is again available, this transaction is how you enter a "FORCE" into the system.)

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	OFFLINEFORCE
gateid	12	Yes	GateID from a previous AUTH transaction.
amount	12	Yes	Final amount (USD)
*** Optional Fields ***			
duplicates_window	6	No	Specifies the time, in minutes, to look back for a duplicate when allow_duplicates is set to "no". The default is 30 minutes.
cardnumber	19	No	Will be used to search for a previous AUTH, if a gateid is not provided
client_transref	20	No	Will be used to search for a previous AUTH, if gateid is not provided. Must match exactly to find the old transaction.
match_window	6	No	Time to look back, in minutes, for an AUTH or OFFLINEAUTH when gateid is not specified and cardnumber/client_transref are specified.
username	20	No	Clerk/Salesperson Name/ID
Email-SMS Options			
send_email	3	No	Set to "yes" to automatically send a receipt to the cardholder via email, if their email address was provided (default is "no")
send_cc	3	No	Set to "yes" to automatically send a receipt to the merchant via email (default is "no")
cc_email	40	No	Provides the email address to be used in the "from" field when sending an email and as the "to" field when send_cc=yes. This overrides the default from the merchant profile (client id email on file)

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

send_sms	3	No	Set to "yes" to send an SMS-Text of the receipt to the phone number provide in the phone field.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Note that **only** the developers using the COM object will have the gratuity added to the total to produce the amount charged to the card. Otherwise, the amount provided is the total amount to charge the card.

The FORCE transaction has been enhanced. **It no longer requires a Gateid, although use of a gateid is still STRONGLY recommended as the preferred way to exactly identify the transaction to FORCE.** Please see the section on VOID/FORCE – Enhancements for more information.

Also note that the FORCE command can now be used on either an AUTH or OFFLINEAUTH, and the system will automatically convert the FORCE to an OFFLINEFORCE if needed.

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Credit Card Update

The Credit Card Update transaction type updates a previous transaction with a corporate card with tax and purchase order information.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

```
EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[2]	SDK developer key obtained from Slim CD, Inc.[2]
*** Required Fields ***			
transtype	18	Yes	UPDATE
po	20	Yes	Purchase Order number
salestax	12	Yes	Amount of sales tax (USD)
salestaxtype	1	Yes	0 – N/A, 1 – Tax Amount, 2 – Exempt Amount
*** Conditionally Required Fields *** (see CREDIT CARD SALE)			
gateid	12	Recommended[1]	Gateid returned from previous transaction
trackdata	varies	Optional[1]	Card swipe track data, or encrypted track data (See Appendix F)
cardnumber	20	Sometimes[1]	Card number – used for verification (if swiped, "trackdata is used instead)
expmonth	2	Sometimes[1]	Expiration month (MM) (if swiped, trackdata is used instead)
expyear	4	Sometimes[1]	Exp year (YYYY or YY) (if swiped, trackdata is used instead)
processor_token	64	Optional[1]	Returned by processor from previous transaction, if available
transient_token	40	Opional[1]	Returned from call to QueueData (usually from JSON/JavaScript)
*** Optional Fields ***			
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

The Credit Card Incremental Auth transaction type increases the authorization amount on a previous AUTH request.

IMPORTANT

"Credit Card Incremental Auth" For some processors, *ONLY* works for Hotels and Direct Marketing/Mail Order (MOTO) businesses! Attempts to use this transtype for other business types will fail.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** **Login Fields** ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.
*** **Required Fields** ***			
transtype	18	Yes	INCREMENTAL
gateid	12	Yes	GateID from a previous transaction.
amount	12	Yes	Amount to be added to auth (USD)
*** **Optional Fields** ***			
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the

---

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

			password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Credit Card Reversal

The Credit Card Reversal transaction type reverses part or all of a previous AUTH transaction.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

```
EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	REVERSAL
gateid	12	Yes	GateID from a previous transaction.
amount	12	Yes	Amount to remove from original auth (USD)
*** Optional Fields ***			
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Credit Card LOAD

The Credit Card LOAD transaction type loads data into the SLIM CD system and returns a SLIM CD Gateid that can be used for tokenization.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

```
EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	LOAD
amount	12	Yes	Please use 0.00
*** Conditionally Required Fields *** (see CREDIT CARD SALE)			
cardnumber	20	Recommended[1]	Card number – Will be validated with standard LUHN MOD10 check
expmonth	2	Recommended[1]	Expiration month (MM) (if swiped, trackdata is used instead)
expyear	4	Recommended[1]	Exp year (YYYY or YY) (if swiped, trackdata is used instead)
processor_token	64	Optional[1]	Returned by processor from previous transaction, if available
trackdata	varies	Optional[1]	Card swipe track data, or encrypted track data (See Appendix F)
transient_token	40	Opional[1]	Returned from call to QueueData (usually from JSON/JavaScript)
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
username	20	No	Clerk/Salesperson Name/ID

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Encryption Options			
encryption_device	20	No	Default is blank, which causes automatic detection based on input. Valid option list will change, but currently: IDTECH, POSX, MAGTEK. Note that all devices default to KB mode, but HID mode can be used if trackdata is sent as all HEX digits.
encryption_type	20	No	Default is blank, which defaults to TDES. Valid option list will change, but currently: TDES or AES
encryption_key	20	No	Indicator of the key injected in the device. Default is blank, where key is determined from indicators in the trackdata.
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] See the comments for CREDIT CARD SALE (ABOVE). Note that providing GATEID is much faster and more accurate than providing CARDNUMBER.

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Check Processing

The Check Processing transaction type provides a way to verify/charge a check payment.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields for FM1, FM2 and ACH ***			
transtype	18	Yes	See Appendix A for list of supported CHECK transactions types
checks	3	Yes	Must contain the value 'yes'
amount	12	Yes	Amount to remove from original auth (USD)
*** Conditionally Required Fields for FM1, FM2 and ACH ***			
checktype	1	Usually	corporate , personal, or savings (specified as C,P, or S respectively)
micr	40	Usually	Contains raw data read from bottom of check
routeno	9	Usually	Bank routing number on check
accountno	20	Usually	Bank account number on check
checkno	9	Usually	Check number on the check
*** Required Fields for FM2 ***			
statecode	2	No	2 letter state code of driver's license (NOTE: This field should contain "SW" if driver's license is swiped or "01" if driver's license barcode is scanned.)
driverlic	20	No	Drivers License number of check writer. This field will contain the driver's license number if hand-entered, or the raw magnetic stripe data if swiped (including sentinels). If the driver's license barcode is optically scanned, this field will contain a "Base64" string of the binary data scanned from the barcode.
*** Required Fields for ACH ***			
seccode	3	Usually	NACHA processing code (WEB, TEL, POP or ACH).
statecode	2	Sometimes	2 letter state code of driver's license
driverlic	20	Sometimes	Drivers License number of check writer
*** Optional Fields ***			
cashback	12	No	This contains the amount paid as cash back to the consumer.
gateid	12	No	GateID from a previous transaction. Used for tokenization.
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
username	20	No	Clerk/Salesperson Name/ID
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
birthdate	6	No	YYMMDD format for the date of birth
accountno	20	No	Account number on check
micrreader	3	No	Indicates presence of MICR reader – "yes" or "no"
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

All check transactions must have the field "checks" specified as "yes".

ACH processing supports Point of Purchase (POP) check conversion. POP can be a SALE transaction, where the check is converted into an electronic draft and the paper is handed back to the customer. POP transactions performed as a SALE should have the checking information read by a hardware check MICR reader.

POP transactions can also be performed as an AUTH transaction, where the check is simply validated and retained so that it can be deposited to your bank as part of your normal nightly banking deposits.

ACH transactions to draft funds from a checking account can use the "seccode" value of WEB, TEL or ACH. WEB and TEL indicate specific methods of acceptance and have their own rules. The "ACH" seccode in the SLIM CD system will actually cause the transaction to be translated into either a PPD or CCD transaction in the NACHA system. If the checktype is set to "C", the transaction will be a CCD (for business checks). Otherwise, the PPD type will be used.

If a Gateid is provided on an ACH transaction, the MICR, ACCOUNT

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

Gift Card

Electronic gift/stored value cards are convenient (carried in wallet with other plastics), secure (cannot be photocopied) and as a branded, "billboard in a wallet", they create a shopping destination for the cardholder. Gift Cards, Loyalty Cards, Prepaid Cards, and Promotional cards all create a physical connection with customers that insulates them from competing offers and results in increased purchase quantity and frequency. The electronic gift card concept is quickly becoming the medium of choice for retailers across the country. Plastic has several advantages such as: No Cash Back (value remaining after partial redemption stays on the card, ensuring additional opportunities to solidify the customer relationship or retain funds) and real time Data Capture (virtually eliminates both the losses associated with paper gift certificates and costly tracking and accounting).

A customer enters the merchant's business and asks to purchase a gift / stored value card. The merchant takes a card from inventory and activates it for the dollar amount desired by swiping the card through a POS terminal. The card is now ready to be used at any participating store location.

Note that some gift card providers support a CVV2 value, a Security Code (seccode) value, and/or both. SLIM CD does not require these fields, but the host processor may. Please note that the security code is often found on a scratch-off area on the back of the card.

Gift card transactions may also return a remaining balance value. This value can be found in the <reply> XML's <datablock> section, as the value of the <bal> tag. They can also return a partially approved amount in the <approvedgcamt> field. Please check that approved amounts match the expected amount. If they do not match, you should either accept an alternate form of payment for the balance, or VOID the transaction.

Example of a BALANCE transaction:

```xml
<?xml version="1.0" encoding="utf-8"?>
<reply>
  <response>Success</response>
  <responsecode>0</responsecode>
  <description>OK</description>

<responseurl>https://trans.slimcd.com/wswebservices/transact.asmx/PostXML</responseurl>
  <datablock>
    <authcode>039334</authcode>
    <approved>Y</approved>
    <gateid>307094</gateid>
    <invoiceno>
    </invoiceno>
    <firstname>
    </firstname>
    <lastname>
```

```
      </lastname>
      <last4>9305</last4>
      <expmonth>12</expmonth>
      <expyear>2049</expyear>
      <approvedamt>0.00</approvedamt>
      <approvedgcamt>0.00</approvedgcamt>
      <bal>6.00</bal>
   </datablock>
</reply>
```

**Gift Card
Activate**

The Gift Card Activate transaction type activates a new gift card and sets the initial card balance.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	ACTIVATE
cardnumber	20	Yes	Card number
amount	12	Yes	Amount to set on card (USD)
cardtype	2	Yes	GC for gift card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is

			associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Gift Card Add Value

The Gift Card Add Value transaction type adds the specified amount to the existing card balance.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

```
EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	ADD
cardnumber	20	Yes	Card number
amount	12	Yes	Amount to add to card balance (USD)
cardtype	2	Yes	GC for gift card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized

			"clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

The Gift Card Auth transaction type holds funds for a gift card transaction.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	AUTH
cardnumber	20	Yes	Card number
cardtype	2	Yes	GC for gift card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is

			associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Gift Card Void

The Gift Card Void transaction type voids a previous gift card transaction.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	VOID
cardnumber	20	Yes	Card number
cardtype	2	Yes	GC for gift card
gateid	12	Yes	GateID of previous transaction to be voided
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the

			ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

The Gift Card Balance transaction type returns the current balance on the gift card.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	BALANCE
cardnumber	20	Yes	Card number
cardtype	2	Yes	GC for gift card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized

			"clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Gift Card Replace

The Gift Card Replace transaction type replaces an old gift card with a new gift card.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	REPLACE
cardnumber	20	Yes	Card number of the new card
cardtype	2	Yes	GC for gift card or LC for loyalty card
oldcardpan	19	Yes	Card number of card being replaced
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the

			ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Gift Card Deactivate w/Refund

The Gift Card Deactivate with Refund transaction type deactivates a gift card and refunds the balance.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	DEACTIVATE
cardnumber	20	Yes	Card number
cardtype	2	Yes	GC for gift card
withrefund	3	Yes	Must contain the value "yes"
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code

Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

The Gift Card Deactivate without refund transaction type deactivates a gift card without refunding the balance.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	DEACTIVATE
cardnumber	20	Yes	Card number
cardtype	2	Yes	GC for gift card
cashback	3	Yes	Must contain the value "no"
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount to set on card (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID

cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

<div align="right">

Gift Card Sale

</div>

The Gift Card Sale - Retail transaction type charges the amount specified from the existing card balance. If the merchant has the Split Tender flag set and there is insufficient funds on the card to cover the transaction, the card balance will be brought to zero and the balance due will be returned.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
cardnumber	20	Yes	Card number
amount	12	Yes	Amount to remove from card balance (USD)
cardtype	2	Yes	GC for gift card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
gratuity	12	No	Amount of gratuity if restraurant

cvv2	4	No	CVV2 Data from card
seccode	12	No	Security Code
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

Loyalty Card

Points-based loyalty card programs are very similar to gift card programs with the exception that they are used to reward frequent customers. Under this program, the system automatically adds a specified Reward Value to your customer's card when that customer has accumulated a designated number of points.

.... For example: when 50 points are accumulated, $10 of spendable value is loaded on the card and the 50 points are removed. This cycle simply repeats, continually motivating the customer to return to earn more points and more reward dollars.

Loyalty Card Activation

The Loyalty Card Activation transaction type activates a new loyalty card and sets the initial card balance.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
cardnumber	20	Yes	Card number
amount	12	Yes	Amount to set on card (USD)
cardtype	2	Yes	LC for loyalty card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

			POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

The Loyalty Card Add Points transaction type adds points to the card based upon the amount.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	ADD
cardnumber	20	Yes	Card number
amount	12	Yes	Amount to add to card balance (USD)
cardtype	2	Yes	LC for loyalty card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

			POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

The Loyalty Card Void transaction type voids a previous loyalty card transaction.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	VOID
cardnumber	20	Yes	Card number
cardtype	2	Yes	LC for loyalty card
gateid	12	Yes	Gateid IDof previous transaction to be voided
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

			support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

Loyalty Card Balance

The Loyalty Card Balance transaction type returns the current point balance and loyalty level if greater than zero on the loyalty card.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	BALANCE
cardnumber	20	Yes	Card number
cardtype	2	Yes	LC for loyalty card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

			support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

Loyalty Card Replace

The Loyalty Card Replace transaction type replaces an old loyalty card with a new loyalty card.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	REPLACE
cardnumber	20	Yes	Card number of the new card
cardtype	2	Yes	LC for loyalty card
oldcardpan	19	Yes	Card number of card being replaced
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

			support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

The Loyalty Card Deactivate transaction type deactivates a loyalty card.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	DEACTIVATE
cardnumber	20	Yes	Card number
cardtype	2	Yes	LC for loyalty card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
amount	12	No	Amount (USD)
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by

---

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

			POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

Loyalty Card Redemption

The Loyalty Card Redemption transaction type removes points from a consumer's loyalty card.

The first 6 fields (clientid, siteid, password, ver, product and key) are sent as the first 6 arguments in the call to PostXML. All fields after "key" are sent as XML "nodes" in the field "XMLData".

EXAMPLE:

```
ReturnXMLNode = PostXML(ClientID, SiteID, PriceID, Pwd, Ver, Product, Key, XMLData)
```

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[1]	SDK developer key obtained from Slim CD, Inc.[1]
*** Required Fields ***			
transtype	18	Yes	REDEM
cardnumber	20	Yes	Card number
points	12	Yes	Number of points to redem
cardtype	2	Yes	LC for loyalty card
*** Optional Fields ***			
first_name	20	No	First name of cardholder
last_name	20	No	Last name of cardholder
address	30	No	Cardholder address
city	20	No	City
state	20	No	State
zip	10	No	Zip code
country	30	No	Country
phone	20	No	Cardholder phone number
email	40	No	Cardholder Email address
clientip	20	No	IP of requesting client or person
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
trackdata	varies	No[3]	Card swipe track data (See Appendix F)
cardpresent	3	No	Contains the value "yes" or "no"
username	20	No	Clerk/Salesperson Name/ID
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

			POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] Obtain from Slim CD, Inc. at 1-877-475-4623

EBT stands for Electronic Benefits Transfer

State and Federal governments, to provide benefits to US citizens, use the EBT system. The ETB system is used for a variety of purposes, including Food Stamps, WIC (a nutritional program for Women, Infants, and Children), and Social Security. EBT transactions can be performed using a card swipe device and a pin pad. Some EBT transactions can also be keyed. The Gateway system supports three types of EBT transactions: Food Stamps, CD Cash Withdrawal, and Cash Benefits. Keyed transactions can be performed, but PIN entry is still required. Paper EBT/Food Stamps can also be processed as an Electronic Voucher. Electronic Vouchers do not require a PIN, but do require other data. EBT transactions must set the "ebt" field to one of three values, "FS", "CD", or "CB". These represent Food Stamps, CD Cash Withdrawal, and Cash Benefits, respectively.

The valid transaction types (transtype) are: SALE, CREDIT, and OFFLINE. The OFFLINE transaction is used to load data into the terminal when:

ebt=FS, CD, or CB

The fields below are used when keying in, or re-entering, a transaction if the terminal is accidentally cleared. Note this does not occur when using the SLIM CD, but is supported anyway as an "OFFLINE" transaction:

```
Retrievalrefno
ebt_transid
auditno
```

The final EBT field is used for Food Stamp Electronic Vouchers.

```
voucherno
```

Note that the CD and CB EBT types allow the "cashback" field to be used to give cash back to the cardholder, in a fashion similar to a DEBIT transaction.

The fields for EBT are:

EBT/FS:

For a "swiped" SALE

```
ebt=FS
transtype=SALE
```

[1] If "trackdata" contains "swiped" card track data then the fields "cardpresent" and "readerpresent" should be present and contain "yes"

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

```
trackdata
pinblock
amount
```

For keyed SALE:

```
ebt=FS
transtype=SALE
cardnumber
expiration date
pinblock
amount
```

For swiped REFUND

```
ebt=FS
transtype=CREDIT
trackdata
pinblock
amount
```

For keyed REFUND:

```
ebt=FS
transtype=CREDIT
cardnumber
expiration date
pinblock
amount
```

EBT/FS - RE-ENTER transactions

It is possible to create a "re-enter" transaction, which places the information into the Gateway for settlement. This is done using the "OFFLINE"

```
ebt=FS
transtype=OFFLINE
cardnumber
expiration date
amount
retrievalrefno
ebt_transid
auditno
```

EBT/FS - VOUCHER transactions

[1] If "trackdata" contains "swiped" card track data then the fields "cardpresent" and "readerpresent" should be present and contain "yes"

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

For keyed SALE:

```
ebt=FS
transtype=SALE
cardnumber
expiration date
voucherno
amount
```

EBT/ CB (Cash Benefits)

Same as EBT, except the "cashback" flag can be set and VOUCHER is not supported

Supports: SALE, OFFLINE, CREDIT

EBT/CD

Same as EBT/ CASH, except the "cashback" flag is not needed and CREDIT is not allowed.

Supports: SALE, OFFLINE

Field Name	Max Length	Required	Description
*** Login Fields ***			
clientid	10	Yes	Assigned by the Slim CD administrator.
siteid	10	Yes	Assigned by the Slim CD administrator.
priceid	10	Yes	Assigned by the Slim CD administrator.
password	20	Yes	Plaintext password for the client account
ver	16	No	Version ID of product
product	32	No	Product Name (Application name)
key	20	Yes[2]	SDK developer key obtained from Slim CD, Inc.[2]
*** Required Fields ***			
transtype	18	Yes	SALE, OFFLINE, or CREDIT
cardnumber	20	Yes	Card number – used for verification
expmonth	2	Yes	Expiration month (MM)
expyear	4	Yes	Exp year (YYYY or YY)
amount	12	Yes	Original amount (USD)
retrievalrefno	6	Yes	EBT transactions only
pinblockdata	16	Yes	Contains the encrypted pin data
ksn	16	Yes	Contains the "Key Set Identifier"
*** Optional Fields ***			

[1] If "trackdata" contains "swiped" card track data then the fields "cardpresent" and "readerpresent" should be present and contain "yes"

[2] Obtain from Slim CD, Inc. at 1-877-475-4623

gateid	12	No	Gateid will be generated for future references
clientip	20	No	IP of requesting client or person
trackdata	varies	No[1]	Card swipe track data (See Appendix F)
cardpresent	3	No[1]	Contains the value "yes" or "no"
readerpresent	3	No[1]	Contains the value "yes" or "no" for card reader at the location
client_transref	20	No	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
authcode	6	No	Auth code received for Offline Transactions
allow_duplicates	3	No	Contains the value "yes" or "no"
cashback	12	No	This contains the amount paid as cash back to the consumer.
*** Varied Requirement Fields ***			
auditno	6	Varies	EBT transactions only
ebt	4	Varies	FS, CD, or CASH – Must specify for EBT transactions
ebt_transid	15	Varies	EBT transactions only
retrievalrefno	12	Varies	EBT transactions only
voucherno	6	Varies	EBT transactions only
Authentication Options			
loginname	20	No	This field can be sent so that the "password" supplied is associated with this "web clerk" user instead of with the ClientID. Specifying a "loginname" allows developers to support SLIM CD WEB CLERKS, or to have a specialized "clerk" created on the SLIM CD website that is used by POS/WEB integration. Note that if a web clerk is created for use by POS/WEB integration. it will not be affected if the password for the CLIENTID is changed. Also note that if "loginname" is provided, and "username" is not provided, the "loginname" will be recorded as the clerk's username for reports.

[1] If "trackdata" contains "swiped" card track data then the fields "cardpresent" and "readerpresent" should be present and contain "yes"

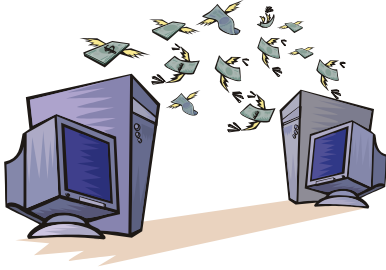[2] Obtain from Slim CD, Inc. at 1-877-475-4623

Closing Batches

Since the BATCH file holds all of the day's transactions, this gives you a way to send the transactions held within the Batch to the credit card processor. This is called "CLOSING" the Batch and is accomplished with a "CLOSEBATCH" function.

All of the transactions to be processed are stored in a Batch on our server system. Many merchants will be set up to "auto-close" the Batch at a specific time. Merchants may also elect to close the BATCH manually using the merchant interface. In addition, the programmer's API can close the Batch when desired.

To close the currently open Batch, call the WEB service function "CloseBatch" as follows:

```
ReturnXMLNode = CloseBatch(UserName, Password, BatchNumber)

WHERE:
```

> UserName = ClientID
> Password = the client's password
> BatchNumber = either an empty string ("") to close the current Batch file or a number to close a "specific" open BATCH file.

Reports

How to Get Reports

Reports are available in two ways: "Preformatted Gateway reports" and "Data only" reports.

Using the Gateways' Existing Reports

Pre-formatted, ready-to-read reports are available with any browser by using the URL:

https://stats.slimcd.com/soft/Interface/default.asp

Your users may simply log in using their own browser, or you can programmatically launch a browser window for them. For example if you are using Microsoft's VB.NET, the following line of code will launch the user's default browser and allow them to login to the reports:

```
System.Diagnostics.Process.Start("cmd /c ""start https://stats.slimcd.com /soft/Interface/default.asp""")
```

Generating Your Own Reports

If you would rather "roll-your-own" reports, the data is available through the Gateway, although it's only available using the SOAP WEB service interface (the data is not available using "name-pairs" interfaces.). The report data is divided into two specific types, Batch-oriented report data and Transaction-oriented report data. Batch reports provide information on Batches that represent deposits into the merchant's bank account. In addition, individual transactions from Batches can be listed. Transaction reports provide a way to access transaction information within date ranges regardless of in which Batch the transaction occurred.

Gateway-oriented report data:

GetBatchHistory

Get the batch totals for a particular site/date range.

Calling method:

```
ReturnXMLNode = GetBatchHistory(UserName, Password, SiteID, StartDate, EndDate)
```

WHERE:

    UserName = the ClientID
    Password = the Client password
    SiteID = the SiteID to get the Gateway history for
    StartDate = get batch history records starting on this date
    EndDate = get batch history records ending on this date

GetBatchHistory2

Get the batch totals for a particular site/date range.

Calling method:

```
ReturnXMLNode = GetBatchHistory2(UserName, Password, SiteID, StartDate, EndDate)
```

WHERE:

    UserName = the ClientID
    Password = the Client password
    SiteID = the SiteID to get the Gateway history for
    StartDate = get batch history records starting on this date
    EndDate = get batch history records ending on this date

GetBatchSummary

Get the approved batch totals for a particular site/date range.

Calling method:

```
ReturnXMLNode = GetBatchSummary(UserName, Password, SiteID, StartDate, EndDate)
```

WHERE:

    UserName = the ClientID
    Password = the Client password
    SiteID = the SiteID to get the batch summary for
    StartDate = get batch summary starting on this date
    EndDate = get batch summary ending on this date

GetClosedBatchTransactions

Get the transactions in a specific Batch.

Calling method:

```
ReturnXMLNode = GetClosedBatchTransactions(UserName, Password, SiteID, BatchID)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> SiteID = the SiteID to get the Gateway history for
> BatchID = the BATCH from which to get the transactions

GetOpenBatch

Get all transactions in the current unsettled Batch.

Calling method:

```
ReturnXMLNode = GetOpenBatch(UserName, Password, SiteID)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> SiteID = the SiteID to get the Gateway history for

GetDailySummary

Get the daily summary for a particular date range.

Calling method:

```
ReturnXMLNode = GetDailySummary(UserName, Password, MetaBankID, BankID, ClientID, SiteID,
StartDate, EndDate)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> MetaBankID = the MetaBankID to get the summary for
> BankID = the BankID to get the summary for
> ClientID = the ClientID to get the summary for
> SiteID = the SiteID to get the summary for
> StartDate = get summary starting on this date
> EndDate = get summary ending on this date

User-oriented reports:

GetUserSettings

Get settings for a specific user.

Calling method:

```
ReturnXMLNode = GetUserSettings(UserName, Password)
```

WHERE:

UserName = the UserName
Password = the Password

GetUserClients

Get a list of clients for a specific user.

Calling method:

```
ReturnXMLNode = GetUserClients(UserName, Password)
```

WHERE:

UserName = the UserName
Password = the Client Password

GetUserClientSites

Get a list of sites for a specific user/clientid.

Calling method:

```
ReturnXMLNode = GetUserClientSites(UserName, Password, ClientID, SiteID)
```

WHERE:

UserName = the UserName
Password = the Password
ClientID = the ClientID
SiteID = the SiteID

Transaction-oriented reports:

GetOpenAuths

Get the unforced "authonly" transactions for a particular site/date range.

Calling method:

```
ReturnXMLNode = GetOpenAuths(UserName, Password, SiteID, StartDate, EndDate)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> SiteID = the SiteID to get the unforced authonly records for
> StartDate = get unforced authonly records starting on this date (if this field is omitted, it
>   defaults to 60 days prior to today)
> EndDate = get unforced authonly records ending on this date

GetTransactionDetails

Get the details of a specific transaction.

Calling method:

```
ReturnXMLNode = GetTransactionDetails(UserName, Password, GatewayID)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> GatewayID = the GatewayID to get the transaction details for

SearchTransactions

Get transactions based on field values.

SearchTransactions searches through ALL records (including the currently open BATCH!) for transactions matching your supplied search criteria.

Calling method:

```
ReturnXMLNode = SearchTransactions(UserName, Password, RecordCountOnly, MaxRecords, StartDate,
EndDate, SearchXML)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> RecordCountOnly = a "BOOLEAN", that, when "TRUE", only returns the number of records this
>   search would return, when "FALSE", return the actual record(s)
> MaxRecords = Maximum number of records to return
> StartDate = Return records entered since this date (formatted MM/DD/YYYY)
> EndDate = Return records entered before this date
> SearchXML = the search field(s) and values to search for, formatted into XML, as in:
>> <search>
>>> <firstname>L</firstname>
>>> <lastname>Lohan</lastname>
>> </search>

The following are valid fields for the SearchXML (all fields are optional, and may occur in any order):

```
cardnumber          -text
amount              -decimal
cardid              -integer
server              -text
clienttransref      -text
firstname           -text
lastname            -text
city                -text
state               -text
zip                 -text
email               -text
cardpart            -text
clientid            -text
siteid              -text
reverseorder        -integer (0 or 1)
```

A minimum SearchXML field required would be:

```
<search></search>
```

A valid SearchXML field, which would find all transactions where the firstname started with 'Lo', would be:

```
<search>
        <firstname>Lo</firstname>
</search>
```

A valid SearchXML which would find all transactions where the firstname started with 'L' and lastname started with 'Lohan' would be:

```
<search>
        <firstname>L</firstname>
        <lastname>Lohan</lastname>
</search>
```

SAMPLE REPORT OUTPUT:

```xml
<?xml version="1.0" encoding="utf-8" ?>
- <reply>
    <response>Success</response>
    <responsecode>0</responsecode>
    <description>OK</description>

    <responseurl>http://stats.slimcd.com/wswebservices/reports.
    asmx/GetBatchHistory</responseurl>
  - <datablock>
    - <Batches>
      - <BatchRecord>
```

```
        <ID>1</ID>
        <BatchDate>6/15/2005 8:57:00 AM</BatchDate>
        <MerchantID>999999999999</MerchantID>
        <BatchNumber>001</BatchNumber>
        <RecordCount>4</RecordCount>
        <ResponseCode>GB</ResponseCode>
        <ResponseText>_ACCEPTED</ResponseText>
        <Amount>8.0000</Amount>
        <ErrorCodes />
        <ErrorData />
    </BatchRecord>
-   <BatchRecord>
        <ID>2</ID>
        <BatchDate>6/15/2005 9:08:00 AM</BatchDate>
        <MerchantID>999999999999</MerchantID>
        <BatchNumber>002</BatchNumber>
        <RecordCount>2</RecordCount>
        <ResponseCode>GB</ResponseCode>
        <ResponseText>_ACCEPTED</ResponseText>
        <Amount>8.0000</Amount>
        <ErrorCodes />
        <ErrorData />
    </BatchRecord>
-   <BatchRecord>
        <ID>3</ID>
        <BatchDate>6/15/2005 2:23:00 PM</BatchDate>
        <MerchantID>999999999999</MerchantID>
        <BatchNumber>003</BatchNumber>
        <RecordCount>11</RecordCount>
        <ResponseCode>GB</ResponseCode>
        <ResponseText>_ACCEPTED</ResponseText>
        <Amount>48.6400</Amount>
        <ErrorCodes />
        <ErrorData />
    </BatchRecord>
-   <BatchRecord>
        <ID>4</ID>
        <BatchDate>6/20/2005 12:48:00 PM</BatchDate>
        <MerchantID>999999999999</MerchantID>
        <BatchNumber>004</BatchNumber>
        <RecordCount>6</RecordCount>
        <ResponseCode>GB</ResponseCode>
        <ResponseText>_ACCEPTED</ResponseText>
        <Amount>140.6400</Amount>
        <ErrorCodes />
        <ErrorData />
    </BatchRecord>
-   <BatchRecord>
        <ID>5</ID>
```

```
          <BatchDate>7/26/2005 9:56:00 AM</BatchDate>
          <MerchantID>999999999999</MerchantID>
          <BatchNumber>003</BatchNumber>
          <RecordCount>-3</RecordCount>
          <ResponseCode>RB</ResponseCode>
          <ResponseText />
          <Amount>46.0600</Amount>
          <ErrorCodes>E/H</ErrorCodes>
          <ErrorData>001</ErrorData>
      </BatchRecord>
    </Batches>
  </datablock>
</reply>
```

Images, Signatures and Receipts

How to Handle Images and Signatures

Check images and signature images are handled through the Gateway WEB service named "Images". The following operations are supported. For a formal definition, please review the Service Description.

- UploadCheck
  Upload a check image to the database.

- UploadSignature
  Upload a signature image to the database.

- UploadReceipt
  Upload a receipt image to the database.

- DownloadCheck
  Download a check image from the database.

- DownloadSignature
  Download a signature image from the database.

- DownloadReceipt
  Download a receipt image from the database.

- ConvertImage
  Converts a stored image from one format to another

UploadCheck

Uploads a check image.

UploadCheck uploads a check image to the Gateway.

Calling method:

```
UploadImage(UserName, Password, SiteID, GateID, FileType, ImageData)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> SiteID = the Client's SiteID
> GateID = Gateway ID of the transaction associated with the image to be uploaded
> FileType = one of the following:
> > "T" = TIFF
> > "J" = JPEG
> > "B" = BMP
> ImageData = the image data in ASCII Base64 format

UploadSignature

Uploads a signature image.

UploadSignature uploads a signature image to the Gateway.

Calling method:

```
UploadSignature(UserName, Password, SiteID, GateID, FileType, ImageData)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> SiteID = the Client's SiteID
> GateID = Gateway ID of the transaction associated with the image to be uploaded
> FileType = one of the following:
> > "1" = SLIMCD
> > "2" = VERIFONE
> > "3" = HYPERCOM
>
> ImageData = the image data in ASCII Base64 format

UploadReceipt

Uploads a receipt image.

UploadReceipt uploads a receipt image to the Gateway.

Calling method:

```
UploadReceipt(UserName, Password, SiteID, GateID, FileType, ImageData)
```

WHERE:

UserName = the ClientID
Password = the Client password
SiteID = the Client's SiteID
GateID = Gateway ID of the transaction associated with the image to be uploaded
FileType = one of the following:
    H = HTML receipt
    T = TIFF Image of receipt
    X = ASCII Text
    J = Jpeg
    P = Binary Print Stream
    B = Windows Bitmap
    A = Adobe PDF
    G = GIF

ImageData = the image data in ASCII Base64 format

DownloadCheck

Downloads a check image.

DownloadCheck downloads a previously uploaded check image from the Gateway.

Calling method:

```
Image = DownloadCheck(UserName, Password, GetwayID)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> GateID = Gateway ID for the image desired to download

```
The returned result of DownloadCheck will be the binary image data encoded as a Base64 string
returned in an XMLNode element named "imagedata", as:
```

```xml
<?xml version="1.0" encoding="utf-8" ?>
<reply>
      <response>Success</response>
      <responsecode>0</responsecode>
      <description>Image download succeeded for 45155</description>
      <responseurl>http://.../ images.asmx/DownloadCheck</responseurl>
      <datablock>
              <filetype>c</filetype>
              <fileformat>image/bmp</fileformat>
              <imagedata>"abcdefgh....=="</imagedata>
      </datablock>
</reply>
```

DownloadSignature

Downloads a signature image.

DownloadSignature downloads a previously uploaded signature image from the Gateway.

Calling method:

```
Image = DownloadSignature(UserName, Password, GetwayID)
```

WHERE:

UserName = the ClientID
Password = the Client password
GateID = Gateway ID for the image desired to download

```
The returned result of DownloadCheck will be the binary image data encoded as a Base64 string
returned in an XMLNode element named "imagedata", as:
```

```xml
<?xml version="1.0" encoding="utf-8" ?>
<reply>
    <response>Success</response>
    <responsecode>0</responsecode>
    <description>Image download succeeded for 45155</description>
    <responseurl>http://.../ images.asmx/DownloadSignature</responseurl>
    <datablock>
        <filetype>c</filetype>
        <fileformat>image/bmp</fileformat>
        <imagedata>"abcdefgh....=="</imagedata>
    </datablock>
</reply>
```

DownloadReceipt

Downloads a receipt image.

DownloadReceipt downloads a previously uploaded receipt image from the Gateway.

Calling method:

```
Image = DownloadReceipt(UserName, Password, GetwayID)
```

WHERE:

UserName = the ClientID
Password = the Client password
GateID = Gateway ID for the image desired to download

```
The returned result of DownloadReceipt will be the binary image data encoded as a Base64
string returned in an XMLNode element named "imagedata", as:
```

```xml
  <?xml version="1.0" encoding="utf-8" ?>
- <reply>
       <response>Success</response>
       <responsecode>0</responsecode>
       <description>Image download succeeded for 45155</description>
       <responseurl>http://.../ images.asmx/DownloadReceipt</responseurl>
-      <datablock>
               <filetype>c</filetype>
               <fileformat>image/bmp</fileformat>
               <imagedata>"abcdefgh....=="</imagedata>
       </datablock>
  </reply>
```

ConvertImage (FUTURE ENHANCEMENT)

Converts a stored image from one format to another.

ConvertImage can convert stored signatures, check images and receipt images from their stored format to another, such as from a TIFF, JPEG or GIF to a BMP format.

Supported conversions:

Signatures checks and receipt images stored as the following:

- Vectors (Slim CD, Verifone or Hypercom)
- TIFF
- JPEG
- BMP
- GIF

May be converted to:

- JPEG
- TIFF
- BMP
- GIF
- WMF
- EMF
- EXIF
- PNG

Calling method:

```
Image = ConvertImage(UserName, Password, ImageType, GateID, DesiredFiletype)
```

WHERE:

> UserName = the ClientID
> Password = the Client password
> ImageType = "S" for Signature, "C" for Check, or "R" for Receipt
> GateID = Gateway ID for the image desired to download
> DesiredFiletype = "jpg", "jpeg", "tiff", "tif", "bmp", "gif", "wmf", "emf","exif" or "png"

The returned result of ConvertImage will be the binary image data encoded as a Base64 string returned in an XMLNode element named "imagedata", as:

```
<?xml version="1.0" encoding="utf-8" ?>
- <reply>
        <response>Success</response>
        <responsecode>0</responsecode>
        <description>Image successfully converted.</description>
        <responseurl>http://.../ images.asmx/ConvertImage </responseurl>
-       <datablock>
                <filetype>s</filetype>
                <fileformat>image/bmp</fileformat>
                <imagedata>"abcdefgh....=="</imagedata>
        </datablock>
  </reply>
```

Appendix A – Transtype Codes

The following chart lists all the currently supported transaction type codes that may be used in requests to the Slim CD web services.

TRANSACTION TYPES	
TYPE	CODE
CREDIT CARD SPECIFIC	
Credit Card Credit	CREDIT
Credit Card Void	VOID
Credit Card Sale	SALE
Credit Card Auth	AUTH
Credit Card Force	FORCE
Credit Card Tip Edit	TIPEDIT
Credit Card Offline	OFFLINE
Credit Card Offline Auth	OFFLINEAUTH
Credit Card Offline Force	OFFLINEFORCE
Credit Card Update	UPDATE
Credit Card Incremental Auth	INCREMENTAL
Credit Card Reversal	REVERSAL
CHECK SPECIFIC	
Verify check info (Certegy FM1, FM2, etc)	CHECKVER
Verify check info (Certegy ECC or GETI)	AUTH
Process Check (CrossCheck, ECC, GETI)	SALE
Cancel Process Check (Crosscheck, GETI)	VOID
Credit funds back in acct (CrossCheck, GETI)	CREDIT
GIFT CARD/LOYALTY CARD SPECIFIC	
Gift Card Activate	ACTIVATE
Gift Card Add Value	ADD
Gift Card Void	VOID
Gift Card Balance	BALANCE
Gift Card Replace	REPLACE
Gift Card Deactivate with Refund	DEACTIVATE
Gift Card Deactivate w/o Refund	DEACTIVATE
Gift Card Sale – Retail	SALE
Gift Card Sale – Restaurant	SALE
Loyalty Card Activation	ACTIVATE
Loyalty Card Add Points	ADD
Loyalty Card Void	VOID
Loyalty Card Balance	BALANCE
Loyalty Card Replace	REPLACE
Loyalty Card Deactivate	DEACTIVATE
Loyalty Card Redemption	REDEM

Appendix B – All Fields

The following chart lists all the currently supported transaction fields that may be used in requests/responses to and from the Slim CD web services

Field Name	Max. Width`	Description
accountno	20	Account number on check
address	30	Cardholder address
allow_duplicates	3	Contains the value "yes" or "no"
Allow_partial	3	Contains the value "yes" or "no". The default is "no". When set to "yes", an approval can occur where the approved amount is LESS than the requested amount. It is the developer's responsibility to check the "approvedamt" field to determine
Amount	12	Amount to credit (USD)
AmtClinic4W	12	Amount of Clinic/other qualified medical
AmtDental4X	12	Amount of Dental
AmtHealth4S	12	Total Amount of Healthcare (if zero, IIAS processing will not be performed)
AmtPrescr4U	12	Amount of Prescription/Rx
AmtVision4V	12	Amount of Vision/Optical
auditno	6	EBT trans only
authcode	6	Auth code received for Offline Transactions
birthdate	6	YYMMDD format for the date of birth
blind_credit	3	Allow a credit with no previous sale - set this to yes/no
cardnumber	20	Card number – used for verification
cardpresent	3	Contains the value "yes" or "no"
cardtype	2	GC for gift card or LC for loyalty card
cashback	12	This contains the amount paid as cash back to the consumer.
chargetype	1	1=Hotel, 2=Restaurant, 3=Gift Shop
check indate	6	YYMMDD format for the date of check in. In the case of a prepaid reservation, this date represents the date of the reservation. In the case of a no-show charge this would be the date the reservation was made
checkintime	4	Check in time as HHMM in 24 hour format (optional for some processors)
checkno	9	Check number on the check
checkoutdate	6	YYMMDD format date of the expected departure
checkouttime	4	Check out time as HHMM in 24 hour format (optional for some processors)
checks	3	Must contain the value 'yes'
checktype	12	Business, personal, or savings
city	20	City
client_transref	20	Client-supplied transaction identifier, (i.e. receipt number) which will be passed to the credit card network for specific industry types.
clientid	10	Assigned by the Slim CD administrator.
clientip	20	IP of requesting client or person
corporatecard	3	Contains the value "yes" or "no"
country	30	Country

customercode	8	Internal customer id
cvv2	4	CVV2
duplicates_window	6	Amount of time, in minutes, to look back for a duplicate. Default is 30 minutes, maximum is 20,000 minutes. This value is only used if "allow_duplicates" is set to "no"
driverlic	2096	Drivers License, SSN, Military ID or Passport ID number, or Base64 Driver's License bar code binary data
ebt	4	FS, CD, or CASH – Must specify for EBT trans
ebt_transid	15	EBT trans only
egoodsindicator	2	"P" for physical goods, "D" for digital goods, or "PD" for a combination of both
email	40	Cardholder Email address
expmonth	2	Expiration month (MM)
expyear	4	Exp year (YYYY or YY)
extracharge	6	0=None, 1=Gasoline, 2=Mileage, 3=Late Return, 4=One way service fee, 5=Driving violation *The data should be left-justified and zero padded to 6 character positions
extra_credit	3	Allow a credit to exceed the amount of the original sale when the gateid of an original sale/force is provided - set this to yes/no (default is no)
first_name	20	First name of cardholder
gateid	12	Gateway ID for previous transaction.
gratuity	12	Amount of the charge that represents tip
IIASindicator	3	Contains the value "yes" or "no". When Yes, IIAS processing is attempted.
IISAVerified	3	Contains the value "yes" or "no". Indicates if POS system verified IIAS data.
installmentcount	9	Rebilling value - number of payments in this installment cycle
installmentseqno	9	Rebilling value - number of the installment being sent.
ksn	16 or 20	The Key Serial Number from the pinpad, to be used with "pinblock" as a replacement for the larger "pinblockdata" that contains both fields in one.
last_name	20	Last name of cardholder
match_window	6	Specifies how far back (in minutes) to look for a match when processing a VOID, FORCE, or OFFLINEFORCE without a gateid. Default is 2880 (2 days) for a VOID, and 21600 (15 days) for a FORCE or OFFLINEFORCE.
micr	40	Contains raw data read from bottom of check
micrreader	3	Indicates presence of MICR reader – "yes" or "no"
nonroomcharge	12	Non room charges such as mini bar, laundry, telephone, other
noshow	1	Indicates charges for a reservation that caused a non-cancellation penalty: 0 = None, 1 = No Show
oldcardpan	19	Card number of card being replaced
password	20	Plaintext password for the client account
phone	20	Cardholder phone number
pinblock	16	Contains the encrypted pinblock to use with the ksn value.
pinblockdata	32	Contains the encrypted pin code and decryption key from the pinpad. This variable is processor-specific, and should be replaced by using the "pinblock" and "ksn" values as separate fields.
po	9	purchase order number - used in purchase card data
points	?	valuetech value - see lou
prestigiousind	1	A VISA flag indicating: space=Not participating, D=$500, B=$1000, S=$1500

priceid	10	Assigned by the Slim CD administrator.
quasicash	3	Contains the value "yes" for quasi-cash
readerpresent	3	Contains the value "yes" or "no" for a card reader at the merchant's location
recurring	2	whether this is a recurring transaction - set to yes/no
RentalAgrmntNbr	10	Rental agreement number
rentalcity	30	City where rental originated
RentalClassID	8	ClassID of rental
rentaldate	6	YYMMDD format, the date the cardholder first drove away with rental
rentaltime	4	Rental time as HHMM in 24 hour format
rentalreturndate	6	YYMMDD format, the date the cardholder returns rental
rentalreturntime	4	Rental return time as HHMM in 24 hour format
rentalstate	3	State where rental originated
rentername	20	Name of car renter (Mastercard)
retrievalrefno	12	EBT trans only
returncity	18	Name of city which the car is returned (Mastercard)
returnlocationid	10	Location of the return center (Mastercard)
returnstate	3	Name of state/country where rental was returned (Mastercard)
roomamount	12	Amount of the room
roomnumber	6	Guest's room number
roomrate	12	Amount of the daily rate for the room
roomratetax	6	Tax rate for room (i.e. .05)
roomtax	12	Tax amount for room
routno	9	Bank routing number on check
salestax	9	Decimal / purchase card data
salestaxtype	9	integer / purchase card data
seccode	3	NACHA processing types for ACH checks (WEB, TEL, ACH, or POP)
serialno	9	check value
shipdate	4	date in MMDD format
sigdata	5000	String representing signature data
siteid	10	Assigned by the Slim CD administrator.
state	20	State
statecode	2	2 letter state code
stayduration	2	This field is required when making a reservation or checking in. When booking a reservation, this should contain the number of days of prepayment the charge represents. In the case of a rental, this represents the number of days for the expected rental. This field should be right justified and zero filled to two digits.
trackdata	varies	Card swipe track data (See Appendix F)
transtype	18	(See Transaction Type Table for Legal Values)
username	20	Clerk/Salesperson Name/ID
voucherno	6	EBT trans only

withrefund	3	Contains the value "yes" or "no" for Credit Card Deactivate
zip	10	Zip code

REPLY FIELDS

The typical reply from the gateway will appear as follows:

XML:

```
<reply>
    <responsecode>0</responsecode>
    <response>Success</response>
    <description>OK</description>
    <responseurl>https://trans.slimcd.com/wswebservices/transact.asmx/PostXML</responseurl>
    <datablock>
            <authcode>TAS018</authcode>
            <approved>Y</approved>
            <cvv2reply>
            </cvv2reply>
            <avsreply>
            </avsreply>
            <gateid>166798</gateid>
            <bankid>41631</bankid>
            <corporatecardindicator>
            </corporatecardindicator>
            <invoiceno>0</invoiceno>
            <firstname>test</firstname>
            <lastname>person</lastname>
            <cardtype>Visa</cardtype>
            <last4>9999</last4>
            <expmonth>11</expmonth>
            <expyear>10</expyear>
            <processor_token>
            </processor_token>
            <approvedhsaamt>2</approvedhsaamt>
            <approvedamt>2</approvedamt>
            <bal>0.0</bal>
    </datablock>
</reply>
```

Field Name	Description
responsecode	Contains zero (0) if successful, non-zero if unsuccessful
response	Contains the word "Success", "Failed" or "Error"
description	Contains an English description of the result (may be used as feedback to the user)
responseurl	Contains the full URL of the service that generated this response
datablock	Contains the actual result, which varies depending on the requested transaction. It typically includes: ▪ authcode – as received from the processor ▪ "Y" = Approved ▪ "B" = Approved, but identifies a Corporate Card that may need more information for the best rates (see UPDATE Transaction) ▪ "N" = Declined

	▪ "D" = Disallowed due to fraud scrubbing
	▪ "E" = Error (See Error codes in Appendix)
	▪ gateid – unique identifying number assigned to this transaction by the Gateway (used to reference this transaction in further actions
	▪ bal – if not present or empty, the balance information is not available. If zero or other amount, this balance was returned by the processor and should be printed on the receipt.
	▪ Approvedamt – if present, this indicates the amount for which the authorization request was actually approved. Please note that if FSA/HSA cards are processed OR if the Allow_partial flag was set for credit/debit cards, this amount must be checked to see if a full approval was performed or just a partial approval the POS device must collect the remaining funds from some other form of payment.
	▪ other information, as needed…. (see below)

Or as a URL encoded post return string (individual fields instead of a datablock):

response=Success&responsecode=0&description=OK&responseurl=OCC_DEMO/wswebservices/transact.a
smx&authcode=VITAL9&approved=Y&cvv2reply=&avsreply=N&gateid=44173

IMPORTANT INFORMATION:

1. Most credit/debit card transactions will either be approved or declined. An error condition can arise if there are processing problems. However, the flag "Allow_partial" will create a situation where a card can be approved for less than requested. It is up to the POS developer to send the "allow_parital" flag if they wish to support this case. If sent, it is the developer's responsibility to identify when a partial approval has occurred by checking the value of the requested amount against the value of the "approvedamt". If the approvedamt is less than the requested amount, a partial approval has occurred. In this event, the POS developer must either request another form of payment, or issue a VOID to release the approvedamt funds back to the cardholder.

2. New rules from VISA/MC now indicate that if balance information is returned, it is up to the developer to display that information on the consumer's receipt. SLIM CD is facilitating this requirement by supplying additional information in the "bal" field. If it exists and is not empty, the value (even if zero) must be displayed on the receipt for the cardholder.

3. SLIM CD will now return the first_name, last_name, last4 (last 4 digits of the card), expmonth, expyear, cardtype, and processor_token. These fields can be used to print a receipt when sending encrypted track data, and can be used when performing tokenization. Note that the processor_token field will be filled in as SLIM CD completes integration with processors that supply their own tokens.

4. On certain errors and declines, we recommend checking the DESCRIPTION field for information. In the event that field is empty, SLIM CD recommends checking the AUTHCODE field for information from the processor.

5. The GATEID value can be used as a SLIM CD token to perform new SALE transactions. The Processor_Token value can be used instead of a cardnumber when the merchant processor is configured to return those values. This gives parallel tokens within responses from SLIM CD.

Appendix C – Test Accounts

When using the Slim CD accounts on the production servers, here are the test accounts:

TSYS test accounts:

Business Type	Client ID	Site ID	Price ID	Key	Password
Retail	1032	228226448	74	SVD-072-5QQ6-5K58	289075
Restaurant	1033	910590174	75	SVD-430-VFM9-V8TV	239807
Mail Order / Direct Marketing	1034	930098895	76	SVD-483-TLT5-RTFS	234980
e-Commerce	1035	655877458	77	SVD-491-QMFQ-WU89	234980e
Hotel	1036	500664807	78	SVD-469-SQS8-SADC	234980
Auto Rental	1162	997359810	237	SVD-424-KR5Y-Y6H8	234980t

Please note that you can use these test accounts to verify credit, debit, check and gift cards. Different accounts are associated with different processors.

Accounts 1032-1036 and 1162 above are on TSYS's test host.

> CARDNUMBERS: For accounts 1032-1036 and 1162, you may use any cardnumber that passes the LUHN MOD-10 check.
>
> APPROVALS: Amounts from $0.01 - $1.99 may produce different DECLINE messages. To obtain approvals, use amounts from $2.00 up, but less than $1000.
>
> CVV2: For any VISA (card beginning with a 4), use 999.
> For MasterCard (cards beginning with a 5), use 998
> For AMEX cards, use 9997
> For Discover cards, use 996

VANTIV/53 test account:

Business Type	Client ID	Site ID	Price ID	Key	Password
Retail	6373	559818480	8900	SCD-395-4J7G-42XG	test123

> Vantiv test host – Credit Card Numbers:
>
> Visa SWIPE: 4445222299990007
> Visa MANUAL: 4417122900000002
>
> MasterCard SWIPE: 5444009999222205
> MasterCard MANUAL: 5410000000000003
>
> AMEX SWIPE: 341111597242000
> AMEX MANUAL: 340123456789014

Discover SWIPE: 6011000990911111
Discover MANUAL: 6011000990911111

Full set of Discover Card BINS:
        3059990000000022
        3566007770019474
        36259600000004
        3899990000000026
        6221260004598744
        6250940045822542
        6288002323842381
        6499999900055789

Vantiv test host – Gift Card Number: 5858836401000004

Appendix D – Error Codes

The following table provides a list of error messages that can be produced by the Slim CD gateway software.

Code: One-letter identifier, where "E" is an input error and "N" is a decline.
Message: The message returned
Explanation: A simple explanation of the error.

Error Codes from the Slim CD Gateway:

Code	Message	Explanation
E	*This application is currently unavailable. Please contact the systems administrator if you have any questions.	The Slim CD system is down.
E	*clientid-priceid-siteid-missing	You must supply a clientid, siteid, and priceid for each transaction
E	*INVALID_ACCOUNT_DATA_SENT	The supplied clientid or password information is not accurate.
E	*INVALID_ACCOUNT_DATA_SENT	The clientid, siteid and priceid combination is not accurate or acceptable
E	*INVALID_SITEID_OR_PRICEID	The siteid or priceid is not assigned to this clientid.
E	*BS_ERROR	The string "visa" or "Mastercard" appears as the first or last name
E	*INVALID_GATEID_SENT	The gatewayid is not valid
E	*--INVALID_GATEID	The gatewayid is not valid (no such record in our database)
E	*--GATEID_NEEDED	Gateway id needed to perform a FORCE, CREDIT, VOID, FORCE, OFFLINEFORCE, INCREMENTAL, or REVERSAL
E	--NO_MATCH_FOUND	No gateid was specified for a VOID, FORCE, or OFFLINEFORCE, and the cardnumber/client_transref value (and amount for VOID) could not be matched against an approved transaction that occurred within the "match_window" timeframe
E	ERR-Insufficient_data	The signature data does not have a corresponding clientid, siteid, priceid, gateway id, or the signature data is empty.
E	ERR-cant_insert_signature	Error inserting the signature into the database
E	ERR-bad_permissions_or_no_gateid	Could not find the gateid transaction for this signature.
E	*--ONLY_FORCE_AN_APPROVED_ TRANSACTION.	The gateid does not represent an approved transaction.

E	*--ONLY_VOID_AN_APPROVED_ TRANSACTION.	The gateid does not represent an approved transaction.
E	*--ONLY_CREDIT_AN_APPROVED_ TRANSACTION.	The gateid does not represent an approved transaction.
E	*ONLY_OFFLINEFORCE_AN_APPROVED_TRANSACTION.	The gateid does not represent an approved transaction.
E	*ONLY_INCREMENTAL_AN_APPROVED_TRANSACTION.	The gateid does not represent an approved transaction.
E	*ONLY_REVERSAL_AN_APPROVED_ TRANSACTION.	The gateid does not represent an approved transaction.
E	*AMOUNT_MISMATCH.	Can only void a transaction for the exact amount of the original transaction -OR- Can only FORCE a transaction for the exact amount of the original transaction, UNLESS the business type is Restaurat, Hotel, or AutoRental
E	*REFUND_AMOUNT_CANNOT_EXCEED_PURCHASE_AMO UNT.	The amount of the refund must not exceed the amount of the original purchase.
E	*ONLY_FORCE_AN_AUTH.	The gateid provided does not identify an "AUTH" transaction.
E	*ONLY_OFFLINEFORCE_AN_ OFFLINEAUTH.	The gateid provided does not identify an "OFFLINEAUTH" transaction.
E	*CANNOT_VOID_AN_AUTH	The gateid identifies an AUTH transaction. This transaction type cannot be VOIDed.
E	*CANNOT_VOID_AN_OFFLINEAUTH	The gateid identifies an OFFLINEAUTH transaction This transaction type cannot be VOIDed.
E	*CANNOT_VOID_AN_INCREMENTAL	The gateid identifies an INCREMENTAL transaction This transaction type cannot be VOIDed.
E	*CANNOT_VOID_AN_REVERSAL	The gateid identifies a REVERSAL transaction This transaction type cannot be VOIDed.
E	*CANNOT_VOID_AN_VOID	The gateid identifies a VOID transaction This transaction type cannot be VOIDed.
E	*CANNOT_VOID_AN_AUTH	The gateid identifies an AUTH transaction This transaction type cannot be VOIDed.
E	*ONLY_CREDIT_A_SALE_OR_FORCE_OR_OFFLINE_OR_OF FLINEFORCE.	The gateid does not identify a SALE, FORCE, OFFLINE, or OFFLINEFORCE transaction
E	*MOD10_FAILED_OR_CARDEXPIRED	The card number was entered incorrectly, or the card's expiration date has been exceeded.
E	*Need the name of the state, or at least 2 characters for the state abbreviation.	The value of "state" is unacceptable
E	*Unknown state:	The value of "state" is unacceptable

E	*Need the name of the state, or at least 2 characters for the state abbreviation.	The state name or abbreviation was not recognized.
E	*Invalid phone number beginning with +. Unwanted Country code.	The phone number is unacceptable.
E	*Invalid characters in phone number.	The phone number is unacceptable.
E	*Invalid phone number beginning with 0. Unwanted Country code or 011.	The phone number is unacceptable.
E	(1)Database Error - SQL Execution Error	Database connection issue. Retry in a day.
E	(6)Communication Error - Unexpected Out-of-Sequenc	Database connection issue. Retry in a day.
E	(8)Communication Error - I/O Error Connection rese	VITAL ISSUE. Retry later.
E	(8)Communication Error - I/O Error null	VITAL ISSUE. Retry later.
E	BAD MICR SWIPE. RE-ENTER	Bad MICR data sent to processor. Examine your MICR swipe.
E	Invalid State Code	Check your statecode.
E	No database connection available. Connection Error	Network issue. Retry later.
E	R*(3)Communication Error - I/O Error :Connection r	Network issue. Retry later.
E	RESEND WITH DLNO / DOB	The system wants more information. Supply an ID# and date of birth.
E	timeout / cannot connect to servlet	Network issue. Retry later.
E	(3)Communication Error - I/O Error :Connection tim	Network issue. Retry later.
E	(3)Communication Error - I/O Error :java.security.	Network issue. Retry later.
E	No database connection available. Connection Error	Network issue. Retry later.
E	Need Driver License Number	Resend with driverlicense
E	Need MICR Data	Resend with valid MICR data
E	Need State Code	Resend with valid crosscheck statecode
E	timeout / cannot connect to servlet	Network issue. Retry later
E	COMMUNICATION ERROR: System.Net.WebException:	Network issue. Retry later
E	GENERAL TRANSACTION ERROR	Do Not Retry
E	Host connection closed prematurely before receivi	Network issue. Retry later
E	INVALID REFERENCE NUMBER	check your data
E	NO STATION RECORD EXISTS	bad merchant profile
E	REFERENCE NUMBER/AMOUNT MISMATCH	
E	*	decline
E	**8*;salestax:0.00;salestaxtype:1;encryptedpin:;or	web server not responding
E	**Card must be presen;gatetransid:2008363;orderno	card must be present
E	**con_To_InterLive_Er;gatetransid:1774394;orderno:	web server not responding
E	**Merchant_0653386_no;gatetransid:1614812;orderno:	merchant not borded properly
E	**SUMID_QUERY_3Time;gatetransid:1614735;orderno:1	web server not responding
E	*__;gatetransid:2127441;orderno:EUK0900;installmen	web server not responding
E	*200509061538	web server not responding
E	*3Invalid_frm_InterLive	web server not responding
E	*Already_Forced	do not attempt any further action
E	*Already_Voided	do not attempt any further action
E	*Card must be present for retail	need to have card present
E	*Card must present	need to have card present
E	*con_To_InterLive_Err= 10061	Invalid Data send from FDM
E	*FDMS_GateInsert.Open() fail	Invalid Data send from FDM
E	*Invalid authcode: 8464711	Preauthorization code entered wrong

E	*Invalid_amt:0	check the amount being charged
E	*Invalid_zip	change the zip code entered
E	*InvalidRespDataLen	Invalid Data send from FDM
E	*Merchant_0162362_not_found	Try again, renter the transaction
E	*Not_Settled	The type of transaction submitted must have prior settlement
E	*NoOpenClientToFDMS	Try again, renter the transaction
E	*NotMOTECO428805	the type of transaction submitted must be MOTO/E-COM
E	*recv_FRM_IntLive_Err= 10054	reset connection
E	*recv_FRM_IntLive_Err= 10060	timeout error, reset connection
E	*RetryToFDMS	retry transaction
E	*Settled_CREDIT_it	use credit instead of void
E	*SUMID_QUERY_3Times	need to obtain batch # prior to transaction
E	*WriteCommClient_fail	Try again, renter the transaction
E	FDSTDLL	web server not responding
E	FileErR	file error
E	FileErU	file error
E	InvAm	invalid amount
E	InvMIDIM	invalid merchant ID
E	InvMTIR	invalid merchant ID
E	InvMTIRX	invalid merchant ID
E	invTIDR	Invalid terminal ID
E	invTIDU	Invalid terminal ID
E	SysErrRX	web server not responding
E	TinactRX	web server not responding
E	TinactU	web server not responding
E	(4)Communication Error : I/O Error Connection refu	Network issue. Retry later
E	(3)Communication Error - I/O Error :Read timed out	Network issue. Retry later
E	Invalid Approval Code	check authorization code
E	No database connection available. Connection Error	Network issue. Retry later
E	Original Transaction is NOT a SALE. Check VitalGat	check your original ticket #
E	Transaction has been FORCED	do not attempt any further action
E	Transaction has been settled. Use CREDIT	use credit instead of void
E	(1)Database Error - SQL Execution Error String or	Network issue. Retry later
E	(3)Communication Error - I/O Error :java.security	Network issue. Retry later
E	(8)Communication Error - I/O Error null	Network issue. Retry later
E	Auth Transaction has been forced. Use Force Transa	Use Force
E	Can NOT VOID this Transaction. Check Gateway Recor	transaction has been settled
E	No database connection available. Connection Error	Network issue. Retry later
E	Tip Edit only allowed on FORCE or SALE. Check HLGa	check your data
E	Transaction has been settled. Use CREDIT	use credit instead of void
E	Error:-2147012744:The server returned an invalid o	Network issue. Retry later
E	Error:-2147012894:The operation timed out	Network issue. Retry later
E	IO Exception in Open Socket-Connection refused: co	Network issue. Retry later
E	No database connection available. Connection Error	Network issue. Retry later
E	BAD-MID-TID-USERNAME	Fix your merchant profile and try again
E	Cant connect to PAYMENTECH - retry in 15 seconds	Network issue. Retry later
E	CONN_ERR_TO_PMT	Network issue. Retry later

E	CONN_ERROR	Network issue. Retry later
E	Error communicating to the host	Retry later
E	initfail-Cannot access merchant index: 10 ()	Servlet is exclusive locked for that merchant. Retry later
E	initfail-Cannot access merchant index: 10 (Batchin	Merchant is batching. Retry later
E	initfail-Cannot perform a reversal authorization o	Do Not Retry
E	initfail-Cannot VOID a settled transaction	use credit instead of void
E	initfail-Cannot void this transaction type	use credit instead of void
E	initfail-Only approved transactions can be updated	do not attempt to modify a declined transaction
E	initfail-This transaction has already been voided	Do Not Retry
E	initfail-Transaction has been voided	Do Not Retry
E	NETCONNECT-ERR	General error. Retry later
E	The request contained an invalid password: [028100	CHECK your merchant profile setup
E	The request contained an invalid password: [P:0281	CHECK your merchant profile setup
E	ValdateMID: MID In Mime [02810004258] Doesn't Mat	merchant profile error
E	ValdateMID: MID In Mime [43122022188] Doesn't Mat	CHECK your merchant profile setup
E	Auth Transaction has been forced. Use Force Transa	operate on the force, not the auth
E	Error:-2147012744:The server returned an invalid o	Network issue. Retry later
E	Tip Edit only supported for Restaurant: 2	only use tipedit for restaurant
E	Transaction can NOT be FORCED. Check RBSGate Recor	transaction has already been forced or is not forceable
E	Transaction has been FORCED	do not attempt any further action
E	(10)Database Error - SQL Execution Error Transacti	contact systems administrator
E	(3)Communication Error - I/O Error :Connection ref	Network issue. Retry later
E	(3)Communication Error - I/O Error :Connection res	Network issue. Retry later
E	(3)Communication Error - I/O Error :Connection tim	Network issue. Retry later
E	(3)Communication Error - I/O Error :java.security	Network issue. Retry later
E	(3)Communication Error - I/O Error :No buffer spac	Network issue. Retry later
E	(3)Communication Error - I/O Error :Remote host cl	Network issue. Retry later
E	(5.1)Communication Error - No Transmission from Vi	Network issue. Retry later
E	(6)Communication Error - Unexpected Out-of-Sequenc	Network issue. Retry later
E	(8)Communication Error - I/O Error Connection rese	Network issue. Retry later.
E	(8)Communication Error - I/O Error null	Network issue. Retry later
E	(8)Communication Error - I/O Error Read timed out	Network issue. Retry later
E	Bad Stay Duration Value	make sure stay duration is numeric
E	Can NOT VOID this Transaction. Check Gateway Recor	transaction has already been settled
E	Cannot_create_xmlhttp:13:Type mismatch	Retry later
E	Error:-2147012744:The server returned an invalid o	Retry later
E	Error:-2147012867:A connection with the server cou	Retry later
E	Error:-2147012894:The operation timed out	Retry later
E	Inconsistent Busisness Type in client data	Check merchant profile on SLIMCD, make sure you have the right businesstype
E	Increment/Reversal is only supported by VISA	only use incremental reversal with a visa
E	Incremental is only supported for Hotel	only use incremental with a hotel merchant account
E	Installment Sequence Number can NOT be greater tha	check your specification
E	Invalid Auth Transaction Code	make sure auth transaction code is not all zeros

E	Invalid CVV2 Data:11o2/o. Only alpha-numeric allow	make sure there are no invalid characters in cvv2 data field
E	Invalid PIN data length	contact slimCD hardware support
E	Invalid Retrieval Reference No	contact slimCD hardware support
E	Missing PIN data	contact slimCD hardware support
E	Need Local Transaction Date	contact developer support
E	No database connection available. Connection Error	Network issue. Retry later
E	No offline CREDIT/FORCE/VOID allowed for Original	contact developer support
E	Transaction already SETTLED/FORCED. Please use SAL	please use sale
E	Transaction can NOT be FORCED. Check VitalGate Rec	transaction has already been forced or is not forceable
E	Transaction has been FORCED	Do Not Retry
E	Transaction has been settled. Use CREDIT	use credit instead of void
E	Transaction is not SALE/OFFLINE. Can not be CREDIT	you cannot credit this transaction
E	Unmatched Merchant ID:0	contact Slim CD support
E	Unrecognized Response from Processor	contact Slim CD support

Certain data input errors will be presented as a SINGLE MESSAGE containing multiple errors. Those messages can include the following sub-strings:

Code	Message portion	Explanation
E	--ERROR READING CARD - RESCAN.	The track data from scanning the card contains errors. Please re-scan the card.
E	--NEED_FIRSTNAME.	The first name was not provided.
E	--INVALID_FIRSTNAME.	The first name contains unacceptable text.
E	--NEED_LASTNAME	The last name was not provided.
E	--INVALID_LASTNAME	The last name contains unacceptable text.
E	--FIRSTNAME_MATCHES_LASTNAME	The first name and last name cannot be the same.
E	--NEED_STREET	The street address was not provided
E	--INVALID_STREET.	The street is invalid (does not contain reasonable street information)
E	--NEED_CITY	The city was not provided
E	--NEED_STATE	The state was not provided
E	--NEED_ZIP	The zip code / postal code was not provided
E	--NEED_COUNTRY	The country was not provided.
E	--UNKNOWN_NOT_ACCEPTABLE	The country cannot be set to "Unknown".
E	--UNKNOWN_NOT_ACCEPTABLE	The country cannot be set to "Select".
E	--NEED_PHONE	The telephone number was not provided
E	--MISSING_CARD	The cardnumber was not provided.
E	--NEED_EXPMONTH	The expiration month for the card was not provided.
E	--NEED_EXPYEAR	The expiration year for the card was not provided.
E	--AMT_MISSING	There was no amount specified
E	--CLIENTID_MISSING	The clientid value was not provided. (internal error)
E	--TRANSTYPE_MISSING	There was no transaction type specified.
E	--AUTHCODE_NEEDED	OFFLINE or OFFLINEAUTH transactions require an AUTHCODE
E	--NEED_EMAIL	Email address is required, but not supplied
E	--INVALID_EMAIL	Inaccurate email address information.
E	--INVALID_EMAIL:No @ sign detected. An @ sign is part of every e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:An @ sign cannot be the last character of the e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:An @ sign cannot be the first character of the e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:A valid e-mail address contains only one @ sign.	Inaccurate email address information.
E	--INVALID_EMAIL:No period detected. An e-mail address contains at least one period.	Inaccurate email address information.

E	--INVALID_EMAIL:THE LAST CHARACTER OF THE E-MAIL ADDRESS CANNOT BE A PERIOD.	Inaccurate email address information.
E	--INVALID_EMAIL:The first character of the e-mail address cannot be a period.	Inaccurate email address information.
E	--INVALID_EMAIL:A valid e-mail address cannot contain a comma. If you have a Compuserve account, substitute a period for the comma in your Compuserve ID, like so: 12345.6789@compuserve.com.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have a space as part of a single e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have an asterisk in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have a close parenthesis sign in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have an open parenthes= sign in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have a greater than sign in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have a less than sign in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have a colon in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have a semicolon in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have a double quote in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:You cannot have two consecutive periods in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:An @ sign cannot be directly followed by a period in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:An @ sign cannot be directly preceded by a period in an e-mail address.	Inaccurate email address information.
E	--INVALID_EMAIL:The extension '.edy' is not a valid extension in an e-mail address. You probably mean '.edu'	Inaccurate email address information.
E	--INVALID_EMAIL:The extension '.con' is not a valid extension in an e-mail address. You probably mean '.com'	Inaccurate email address information.
E	--MISSING_SSL	The transaction must be performed using SSL (Secure Sockets Layer)
E	--IP_NOT_ALLOWED	The IP address for this transaction is not allowed.
E	--BAD_PASSWD	The password supplied is inaccurate.
E	--AUTH_FORCE_DISALLOWED	This account cannot use the AUTH/FORCE or OFFLINEAUTH/OFFLINEFORCE combinations

CERTEGY ECC Responses from the Gateway:

CODE	MESSAGE
Y	CHECK IS ACCEPTED FOR CONVERSION
Y	CONVERSION OF CHECK REVERSED
N	CHECK HAS ALREADY BEEN VOIDED
N	COMM ERROR BBAL
N	GENERAL TRANSACTION ERROR
N	CHECK IS NOT ACCEPTED FOR CONVERSION
N	INVALID MERCHANT
N	SUCCESSFULL BANK BALANCE REPORT
N	NO STATION RECORD EXISTS
N	DUPLICATE EFT
N	INVALID REFERENCE NUMBER
N	TOO LATE TO VOID CHECK
N	REFERENCE NUMBER/AMOUNT MISMATCH
N	CALL AUTHORIZATION CENTER
N	NO CONNECTION BS
E	Improper Merchant ID! Value:
E	Unknown Service Type:
E	Lane Number not between 0 and 99:
E	Department Number not between 0 and 99999999:
E	Receipt Header too long! (Max = 256 chars)
E	Receipt Footer too long! (Max = 256 chars)
E	Boolean value expected! (T, t, Y, Yes, y, yes, 1 or F, f, N, No, n, no, 0)
E	Invalid Service Type! (Should be FM1 or FM2!)
E	Improper Merchant ID! Value:& Value)
E	Merchant with that ID does not exist!
E	Improper MICR Line! (Should be 15-36 chars!)
E	Check amount invalid! (must be 0 to $99,999.99)
E	Third Party Reference Number incorrect format (should be 4-9 Alpha chars)!
E	ID Date of Birth incorrect format (should be 4-8 chars)!
E	Invalid phone number! (Should be 10 digits no hyphens!)
E	Invalid State Code:
E	Invalid driver's license state code!
E	Invalid ID number! (Should be 5-2048 characters.)
E	Invalid driver's magnetic track 1 data!
E	Invalid driver's magnetic track 2 data!
E	Invalid driver's magnetic track 3 data!
E	Invalid SSN! (Should be 9 chars no hyphens!)
E	Invalid Passport number (4-24 chars)!
E	Invalid Military ID (4-24 chars)!
E	Barcoded ID incorrectly formed Base64 format!
E	Lane Number Max = 99!
E	Clerk ID Max = 9999!
E	Unknown Product Type:& Value)
E	Unknown Transaction Type:& Value)
E	Incorrect Report Date Format:& Value)
E	Incorrect Terminal_Type Format:& Value)

E	Unknown XML name:+ tmpStr)
E	Incorrectly formed XML Transaction! (No child nodes)
E	Missing ID! (Must have DL# SSN Passport Military or barcoded ID!)
E	Unknown Service Type! (Must be FM1 or FM2!)
E	Incorrect Merchant ID length!
E	Missing Terminal Type!
E	AUTH not allowed! (Merchant setting allows "Conversion Only)
E	Check Amount not set!
E	Manually entered check number missing!
E	MICR Source not set or incorrect!
E	MICR Line missing for "SALE" or "AUTH"!
E	Reference number must be set for "VOID"!
E	Return Amount must be set for "VOID"!
E	Date for Bank Balance Report not set!
E	Transaction Type not set or incorrect!
E	Server URL missing!
E	Incorrect TCP/IP Port number!
E	Incorrect host TimeOut setting!
E	No response received from processor!
E	Received NAK response to request!
E	Unrecognized response!
E	No response received!
E	Bad request format:
E	MerchantID missing or incorrect for ACK Message!
E	Reference no. missing or incorrect for ACK Message!
E	Could not connect to the website!
E	~The connection to the website failed!
E	~Unable to send the transaction request!
E	~Failed to receive ENQ!
E	~Connection shut down while waiting on ENQ!
E	~Unable to receive the ENQ!
E	~Unable to send the transaction request!
E	Unable to receive the reply!
E	~Unable to send the ACK!
E	~Host connection closed prematurely before receiving response
E	Unable to receive the reply!
E	~Unable to send the ACK message!
E	~Unable to setup to receive the EOT reply!
E	~Unable to receive the EOT reply!

Decline Codes from the Gateway:

Code	EMessage portion	Explanation
N	*APPROVELIMIT_REACHED_FOR_THIS_CARD	Internal fraud screening due to frequent use of card.
N	-AVS_CODE_BANNED	The Address Verification Results (AVS) are not acceptable to the merchant's bank. Use the billing address of the cardholder. -OR- A merchant's customer used a non-us card and the merchant's bank will only allow the acceptance of US-issued cards.
N	INVALID_GIDENTIFIER-xxx	xxx will represent the invalid processor identifer. This is a configuration issue, and must be fixed by the Slim CD support staff.
N	DECLINE_51	Declined due to internal fraud screening
N	DECLINE_56	Declined due to internal fraud screening
N	DECLINE_57	Declined due to internal fraud screening
N	DECLINE_58	Internal fraud screening due to frequent use of card.
N	*ERROR:Country/state do not match area-code/phone number:1	Fraud screening
N	*ERROR:Zipcode does not match country/state verification failed:2	Fraud screening
N	*ERROR:Zipcode does not match country/state:3	Fraud screening
N	*This phone number cannot be accepted	Fraud screening – a mobile phone or pager has been identified in a transaction.
N	53_Do_not_honor	Card is not acceptable.
N	CARD_BANNED	Card is not acceptable.
N	IP_BANNED	IP is not acceptable.
N	CCODE_BANNED	Country is not acceptable.
N	BAD_CCODE	County code not accurate
N	BAD_IPCC	IP not allowed
N	BAD_BINCC	Card Not allowed
N	NEGATIVEDB_BLOCK	Transaction declined
N	NEGATIVEIP_BLOCK	Transaction declined
N	CCODE_BLOCK	Transaction declined
N	IPCARDLIMIT	Transaction declined
N	DECLINELIMIT	Transaction declined
N	APPROVELIMIT	Transaction declined

Decline codes from the processor will be in the following format:

DECLINE XX YZ

Where the XX is a processor-specific numeric decline code. The Y represents an AVS character, and the Z represents the CVV2 reply.

AVS /CVV2 Codes from the Gateway:

The following table presents AVS character replies and their meanings:

AVS Letter	Meaning
A	Address matches - Zip Code does not
B	Street address match, Postal code in wrong format. (international issuer)
C	Street address and postal code in wrong formats
D	Street address and postal code match (international issuer)
E	AVS Error
G	Service not supported by non-US issuer
I	Address information not verified by international issuer.
M	Street Address and Postal code match (international issuer)
N	No match on address or Zip Code
O	No Response sent
P	Postal codes match, Street address not verified due to incompatible formats.
R	Retry - system is unavailable or timed out
S	Service not supported by issuer
U	Address information is unavailable
W	9-digit Zip Code matches - address does not
X	Exact match to address and 9-digit Zip Code
Y	Address and 5-digit Zip Code match
Z	5-digit zip matches - address does not
Space	No response sent from issuer or processor

The following table presents CVV2 replies and their meanings:

CVV2 Letter	Meaning
M	CVV Matched.
N	CVV No Match.
P	Not Processed.
S	CVV is on the card, but the shopper has indicated that CVV is not present.
U	Issuer is not VISA certified for CVV and has not provided Visa encryption keys or both.

WARNING: PLEASE NOTE THAT A TRANSACTION CAN BE APPROVED EVEN IF THE AVS OR CVV2 REPLY DO NOT MATCH! IT IS UP TO THE MERCHANT TO VOID TRANSACTIONS THAT RESULT IN AVS OR CVV2 REPLIES THEY DO NOT WISH TO ACCEPT THE TRANSACTION.

Appendix E – URL Encoding

HTML form data is usually URL-encoded to package it in a GET or POST submission. In a nutshell, here's how you URL-encode the name-value pairs of the form data:

NOTE

Only alphanumerics [0-9 a-z A-Z] and reserved characters **when used for their reserved purposes** (such as /:=&+@, etc.) are called "SAFE" and may be used unencoded within a URL or response. (If the "reserved characters" appear as DATA, they also must be encoded!)

---

1. Convert all "unsafe" characters in the names and values to "%xx", where "xx" is the ascii value of the character, in hex. "Unsafe" characters include =, &, %, +, non-printable characters, and any others you want to encode--there's no danger in encoding too many characters. For simplicity, you might encode all non-alphanumeric characters.

2. **Change all spaces to plusses!**

3. String the names and values together with = and &, like name1=value1&name2=value2&name3=value3

4. This string is your message body for POST submissions, or the query string for GET submissions.

For example, if a form has a field called "name" that's set to "Lucy", and a field called "neighbors" that's set to "Fred & Ethel", the URL-encoded form data would be
> name=Lucy&neighbors=Fred+%26+Ethel
> with a length of 34.

Technically, the term "URL-encoding" refers only to step 1 above; it's defined in RFC 2396, section 2.4 (previously in RFC 1738, section 2.2). Commonly, the term refers to this entire process of packaging form data into one long string.

PLEASE NOTE that when using the POSTHTTP method, you must URL Encode the values for each name=value pair while building one large string. Once that string is built, you must URL Encode this entire string when placing it into the "postdata=" field.

<u>VB 6 code to URLEncode data:</u>

```
Function URLEncode(ByVal Text As String) As String
    Dim i As Integer
    Dim acode As Integer
    Dim char As String

    URLEncode = Text

    For i = Len(URLEncode) To 1 Step -1
        acode = Asc(Mid$(URLEncode, i, 1))
        Select Case acode
            Case 48 To 57, 65 To 90, 97 To 122
                ' don't touch alphanumeric chars
            Case 32
                ' replace space with "+"
                Mid$(URLEncode, i, 1) = "+"
            Case Else
                ' replace punctuation chars with "%hex"
                URLEncode = Left$(URLEncode, i - 1) & "%" & Hex$(acode) & Mid$ _
                    (URLEncode, i + 1)
        End Select
    Next

End Function
```

The tables on the following pages show proper URL encoding for the entire ASCII set

---

URL-encoding from %00 to %ff

ASCII Value	URL-encode	ASCII Value	URL-encode	ASCII Value	URL-encode
Æ	%00	0	%30	`	%60
	%01	1	%31	a	%61
	%02	2	%32	b	%62
	%03	3	%33	c	%63
	%04	4	%34	d	%64
	%05	5	%35	e	%65
	%06	6	%36	f	%66
	%07	7	%37	g	%67
backspace	%08	8	%38	h	%68
tab	%09	9	%39	i	%69
linefeed	%0a	:	%3a	j	%6a
	%0b	;	%3b	k	%6b
	%0c	<	%3c	l	%6c
c return	%0d	=	%3d	m	%6d
	%0e	>	%3e	n	%6e
	%0f	?	%3f	o	%6f
	%10	@	%40	p	%70
	%11	A	%41	q	%71
	%12	B	%42	r	%72
	%13	C	%43	s	%73
	%14	D	%44	t	%74
	%15	E	%45	u	%75
	%16	F	%46	v	%76
	%17	G	%47	w	%77
	%18	H	%48	x	%78
	%19	I	%49	y	%79
	%1a	J	%4a	z	%7a
	%1b	K	%4b	{	%7b
	%1c	L	%4c			%7c
	%1d	M	%4d	}	%7d
	%1e	N	%4e	~	%7e
	%1f	O	%4f		%7f
space	%20	P	%50	€	%80
!	%21	Q	%51		%81
"	%22	R	%52	‚	%82
#	%23	S	%53	ƒ	%83
$	%24	T	%54	„	%84
%	%25	U	%55	…	%85
&	%26	V	%56	†	%86
'	%27	W	%57	‡	%87
(	%28	X	%58	ˆ	%88
)	%29	Y	%59	‰	%89
*	%2a	Z	%5a	Š	%8a
+	%2b	[	%5b	‹	%8b
,	%2c	\	%5c	Œ	%8c
-	%2d	]	%5d		%8d
.	%2e	^	%5e	Ž	%8e
/	%2f	_	%5f		%8f
	%90	À	%c0	ð	%f0
'	%91	Á	%c1	ñ	%f1
'	%92	Â	%c2	ò	%f2
"	%93	Ã	%c3	ó	%f3

"	%94	Ä	%c4	ô	%f4
•	%95	Å	%c5	õ	%f5
–	%96	Æ	%c6	ö	%f6
—	%97	Ç	%c7	÷	%f7
~	%98	È	%c8	ø	%f8
™	%99	É	%c9	ù	%f9
š	%9a	Ê	%ca	ú	%fa
›	%9b	Ë	%cb	û	%fb
œ	%9c	Ì	%cc	ü	%fc
	%9d	Í	%cd	ý	%fd
ž	%9e	Î	%ce	þ	%fe
Ÿ	%9f	Ï	%cf	ÿ	%ff
	%a0	Ð	%d0		
¡	%a1	Ñ	%d1		
¢	%a2	Ò	%d2		
£	%a3	Ó	%d3		
	%a4	Ô	%d4		
¥	%a5	Õ	%d5		
		%a6	Ö	%d6		
§	%a7		%d7		
¨	%a8	Ø	%d8		
©	%a9	Ù	%d9		
ª	%aa	Ú	%da		
«	%ab	Û	%db		
¬	%ac	Ü	%dc		
	%ad	Ý	%dd		
®	%ae	Þ	%de		
¯	%af	ß	%df		
°	%b0	à	%e0		
±	%b1	á	%e1		
²	%b2	â	%e2		
³	%b3	ã	%e3		
´	%b4	ä	%e4		
µ	%b5	å	%e5		
¶	%b6	æ	%e6		
·	%b7	ç	%e7		
¸	%b8	è	%e8		
¹	%b9	é	%e9		
º	%ba	ê	%ea		
»	%bb	ë	%eb		
¼	%bc	ì	%ec		
½	%bd	í	%ed		
¾	%be	î	%ee		
¿	%bf	ï	%ef		

Appendix F – Magnetic Tracks and Readers

MAGNETIC CARD READERS

Magnetic card readers come in all kinds, shapes and sizes. They also vary in what data they send to the computer. Some send data exactly as desired and described below. Some others strip off the sentinels (described below) and only send the track data (in that instance, you'll have to add the start and end sentinels back onto the track data strings yourself before sending the field "trackdata" to the gateway!) Some readers send different characters for start and end sentinels (in which case you'll have to replace the reader's sentinels with the ones described for each track, above.)

Readers also vary in what they send if an error occurs. Many send the "E" in the data as described below under "Errors". Others may not send anything at all.

MAGNETIC TRACKS

All credit, debit and similar electronic banking cards in use today conform to the international standards ISO 7813 (tracks 1 and 2) and ISO 4909 (track 3). Although these standards describe up to three tracks, electronic banking cards only use tracks 1 and 2 (and many cards only use track 2!). For proper processing, the gateway software expects magnetic track data for both tracks to be sent in the field named "trackdata". (Note: This is only for cards that are "swiped" through a magnetic card reader! Data for cards that are keyed in manually are sent separately in the fields "cardnumber", "expmonth" and "expyear".)

To distinguish each track's data, certain "start" and "stop" characters are defined for each track.

TRACK 2

As mentioned above, many cards only use track two. Track two data includes the primary account number, a country code and the expiration date among other information. When sending track two data in the "trackdata" field, it must be preceded with a start character (called a start sentinel) which is the ASCII semi-colon (;), and must end with a "stop character" (called an end sentinel) which is an ASCII question mark (?). Therefore a properly formatted track two data field would look similar to the following:

;... track two data… …?

Therefore, sending the field "trackdata" in XML format would appear like this:

<trackdata>;... track two data… …?</trackdata>

TRACK 1

Some cards will also contain data on track one. (Track one will never be found without a track two also being present!) If there is data on track one, it should also be sent in the field named "trackdata".

Track one data must be preceded with a start character (start sentinel) which is the ASCII per-cent sign (%), and must end with a stop character (called an end sentinel) which is an ASCII question mark (?).

When sending track one along with track two, the two tracks are concatenated into one string in the "trackdata" field, with track one preceding track two) similar to the following when sent in XML format:

<trackdata>%... track one data… …?;... track two data… …?</trackdata>

ERRORS

Occasionally, while swiping a card through a magnetic reader, errors will occur and the track(s) can't be read. If this occurs, the track data is simply replaced by the ASCII character capital "E". Therefore, an unsuccessful read of track 2 on a card without any data on track one would look like:

<trackdata>;E?trackdata>

Likewise if an error occurred on track one, but track two was successfully read, the field would look like:

<trackdata>%E?;…...track two data… …?</trackdata>

In a similar vein, if an error occurred on track two, but track one was successfully read, the field would look like this:

<trackdata>%... track one data… …?;E?</trackdata>

(If your reader is the "keyboard" or "wedge" variety that sends data to the computer just like you had typed the data on the keyboard, you may test your reader's data by opening a text editor program and swiping a card. The data sent by the reader will appear in the text editor, as if you had typed it. You may also "TEST" your reader's track error reporting by partially swiping a card and then prematurely pulling the card straight up out of the reader before the whole track is read. The reader may respond with an "E" in the data string as described.)

ENCRYPTED READERS

Encrypted readers can produce different formats for Keyboard (KB) or HID mode. SLIM CD's decryption software will attempt to automatically detect the format. The encrypted data can be parsed and will often contain clear-text fields that can be used by the developer to extract the cardholder name and other data. SLIM CD will decrypt this data and return it as part of the transaction to assist those developers that do not wish to attempt parsing the track data strings themselves.

SLIM CD's decryption services attempt to automatically detect the message format used and the encryption key required. However, the CREDIT CARD SALE transaction describes different field values that can be provided to assist the SLIM CD gateway's auto-detection features.

Appendix G – IIAS Processing for Flex Spending Cards (FSA/HSA/HRA cards)

The SLIM CD Gateway can now accept certain purchase details about the Flexible Spending Account purchases. FSA/HSA/HRA cards are Visa/MasterCard/Amex cards that have been attached to employee benefits or some other type of health (or other) benefits.

These Flexible Spending Account cards can also have different "purses" used for different healthcare options (Dental, Vision, etc). When a normal service provider for Dental swipes the card, their business type (identified by the business SIC/MCC code in the processor's system) can allow these merchants to accept FSA cards properly.

Pharmacies are the primary business that needs to send the IIAS indicators, as they may be required to scan the products and verify the UPC codes match those allowed by the Internal Revenue Service (IRS) of the US Government. However, not all pharmacies must check individual products. The current rules (as of this writing) state that IF 90% of the items sold by the pharmacy qualify for IIAS, the individual items need not be scanned.

The SLIM CD implementation is as follows:

DATA TO SEND:
To invoke the IIAS-style processing:

1. Set the "IIASindicator" to "yes".
2. Set the "IISAVerified" flag to "yes" if the individual items were verified, "no" if they were not.

To specify the total amounts set in the individual fields,

1. Set the "Amount" field to the total amount of the sale
2. Set the "AmtHealth4S" field to indicate the amount to be charged as Health Benefits

Once the total amount is specified, then the individual amounts can be specified so that the funds come from the correct purse.

1. Set the "AmtPrescr4U" field to the total amount of the RX or Prescription products.
2. Set the "AmtVision4V" field to the amount of the Vision/Optical products/services
3. Set the "AmtDental4X" field to the amount of the Dental products/services
4. Set the "AmtClinic4W" field to the amount of the Clinic or Other qualified medical products/services

RESPONSE DATA:

The system will return the amount of the approved transaction in the following field:

"approvedamt"

Note that the following logic should be deployed:

if approved = Y or approved = B then

   approvedamt is the amount actually approved ,

  if   amount-approvedamt > 0 then

    Collect any remaining balance

  end if

end if

Please see the below example of sending and receiving IIAS/HSA information:

Sending HSA data:

```
clientid=1111&siteid=249949344&priceid=149&password=visa123&key=:75400000&product=SLIMCD&ver=5.95L&XMLData=
<request>
        <last_name/>
        <first_name/>
        <address/>
        <city/>
        <state>FL</state>
        <zip/>
        <country>US</country>
        <phone/>
        <email/>
        <cardnumber>4012000033330026</cardnumber>
        <expyear>2008</expyear>
        <expmonth>11</expmonth>
        <amount>11.10</amount>
        <cvv2/>
        <clientip>192.168.0.101</clientip>
        <IIASindicator>yeS</IIASindicator>
        <IISAVerified>yEs</IISAVerified>
        <AmtHealth4S>5.55</AmtHealth4S>
        <AmtPrescr4U>0.00</AmtPrescr4U>
        <AmtVision4V>0.00</AmtVision4V>
        <AmtClinic4W>0.00</AmtClinic4W>
        <AmtDental4X>0.00</AmtDental4X>
        <TrackData/>
        <cardpresent>yes</cardpresent>
        <readerpresent>yes</readerpresent>
        <allow_duplicates>yes</allow_duplicates>
        <pinblockdata/>
        <username>ADMIN</username>
        <transtype>SALE</transtype>
        <occgatewayid/>
        <gateid/>
        <gratuity>0.00</gratuity>
        <authcode/>
        <client_transref/>

</request>
```

Response with PARTIAL APPROVAL (shown in APROVEDAMT less than AMOUNT):

```xml
<?xml version="1.0" encoding="utf-8"?>
<reply>
 <response>Success</response>
 <responsecode>0</responsecode>
 <description>OK</description>
 <responseurl>http://dev.slimcd.com/wswebservices/transact.asmx/PostXML</responseurl>
 <datablock>
  <authcode>TAS962</authcode>
  <approved>Y</approved>
  <cvv2reply>
  </cvv2reply>
  <avsreply>N</avsreply>
  <gateid>93575</gateid>
  <bankid>37218</bankid>
  <corporatecardindicator>
  </corporatecardindicator>
  <invoiceno>0</invoiceno>
  <approvedamt>5.55</approveaamt>
 </datablock>
</reply>
```

Note that if the consumer declines to pay the balance or the transaction must be reversed for any reason, a REVERSAL transaction must be performed to release the funds back to the cardholder. Then, a VOID can be performed to avoid settling the transaction in the batch.

Please note that on some processors, AUTH-REVERSAL transactions are only possible if the original transaction was an AUTH. If you need to support AUTH-REVERSALS, you will have to submit AUTH transactions for the initial authorization and FORCE transactions at the conclusion of the sale.

As an update to the above information, VISA/MC/Discover have now implemented the ability to perform FULL REVERSALS for all card types and SLIM CD will be implementing this on all VOID transactions performed on AUTHs or SALEs.

Appendix H – Encryption and Tokenization

SLIM CD Tokenization

The SLIM CD Gateway has always provided tokenization services. The SLIM CD "gateid" value, as returned from each transaction, represents a unique index to all of the information for that transaction. New transactions can be produced by simply supplying the gateid of an approved SALE or AUTH, with a new amount and transaction type. Any data except for the cardnumber can be overridden automatically by supplying new values in the new transaction.

Using this functionality, developers can store SLIM CD gateids as tokens. New transactions can be created without storing cardholder data by sending the gateid value instead of the cardnumber. Updated billing addresses or expiration dates can be supplied with the gateid and the newer information is merged with any information on file with the gateid when processing the new transaction. This allows developers to store the single gateid value for repeat customers, refunds, auth/force processing, and more. While not required, SLIM CD recommends using the gateid of the most recent approved transaction as the token since it will represent the latest collection of information.

Loading initial data

Many merchants have databases of cardnumbers they wish to move to an outside storage facility. SLIM CD has created two new features, LOAD and Zero-dollar AUTH just for this purpose. These transaction types can be used to load data into the SLIM CD system, each with its own advantages. The LOAD function simply stores the data into the SLIM CD servers and returns a gateid as a token. The AUTH transaction can be used with a zero-dollar amount. It will validate either address information or CVV2 information (you must supply one or the other). The zero-dollar AUTH is a great way to validate cardnumber entry and obtain tokens for transactions that will be posted later.

Retrieving data

The SLIM CD gateway now returns information that can assist the developer when offloading data storage to SLIM CD or when using encryption. The gateway will now return the following:

- first name
- last name,
- type of card (Visa/MasterCard/Amex/Discover/etc)
- last 4 digits of the card
- expiration month & year
- a processor-specific token (optional, and may require special setup to enable)

This data can be used to print receipts or identify when cardnumber data on file will expire.

Processing Refunds

Void, Force, and Credit have always worked best and fastest when supplying the "gateid" of the original transaction. This prevents a database search by cardnumber and is the most efficient way to operate. If a gateid is supplied when performing a CREDIT, do not supply the "blind_credit" flag. That flag is only

used when sending cardnumber or processor_token to SLIM CD without a gateid. In the event the developer wishes to use a single "gateid" value for the life of a customer, that original gateid can be used for refunds. However, the amount of the refund may soon exceed the amount of the original transaction. To avoid this, set the value of "extra_credit" to "yes". This will allow sending refunds for more than the amount of the transaction identified by the gateid.

Hardware Encryption

SLIM CD has added support for hardware devices that perform encryption. We are starting with devices from POSX and IDTECH. These devices will encrypt swiped data in hardware. Simply pass the encrypted data to SLIM CD's gateway in the "trackdata" field. Our systems will decrypt the track data, keeping your application out of scope for touching this cardholder data. The returned data will provide enough information to print receipts. SLIM CD stores the actual cardnumber and returns a gateid token that can be used to process future transactions. SLIM CD's implementation allows hardware encryption to be utilized across all SLIM CD processors, as the processor will receive normal cardnumber data from SLIM CD. This solution provides developers and their merchants with true freedom from the locked-in approach of processor-specific encryption techniques.

Processor Tokens

Many processors offer the option to use tokenized values instead of cardnumbers. If enabled, these values will be returned in the "processor_token" field from SLIM CD. Developers can store these values with SLIM CD's gateid token as a way of providing redundancy and alternate access to cardholder data. Note that storing these tokens does not alter your PCI scope, so that if you are already out of scope (store no cardnumbers), you can store these values without changing that status.

SLIM CD can accept the processor_token value as input as well. Note that on some processors, SLIM CD will have had to return the token from a previous transaction for us to successfully accept it in lieu of a cardnumber. For other processors, SLIM CD can accept tokens produced outside of SLIM CD. Please contact SLIM CD to verify if we can accept processor_tokens for your particular situation.