Secure Web Pay (SWP)

Integration Guide
Version 1.2
Updated June 6, 2009

For the latest update, visit our website:
www.PaymentsGateway.com

500 West Bethany Road Suite 200
Allen, Texas 75013
(800) 337-3060 / Fax (972) 392-5006

Table of Contents

CHAPTER 1 - Introduction

What is Secure Web Pay?

Thank you for selecting the PaymentsGateway Secure Web*Pay* to facilitate your online financial transactions. With Secure Web*Pay*, merchants are able to integrate into PaymentsGateway (PG) and submit electronic payment transactions with relatively minimal ease. Once integrated with the PG, merchants are able to process single or recurring transactions (ACH/eCheck, credit card and debit card) in real time. By using Secure Web*Pay* to send every financial transaction through this single channel, merchants both simplify and enhance their current processing options.

Secure WebPay Advantages

Secure Web*Pay* offers numerous advantages. A key advantage of Secure Web*Pay* is that merchants are not required to have a SSL certificate. Secure Web*Pay* allows merchants to accept payments without hosting the payments information, as the payment or checkout page, may be hosted on the Payments Gateway.

When hosting the payments page on the Payments Gateway, merchants enjoy the ease of a no worry ecommerce experience. Because there is no sensitive data on the merchant side, there is no liability with maintaining credit card information or the possibility of being hacked. Anti-fraud, authorization and verification are all built-in and performed during the checkout process. In addition, routine PCI and NACHA audits on the Payments Gateway side ensure an even more secure and safe transaction.

Secure Web*Pay* also allows for a fully customizable payment page. Merchants are able to customize data elements by selecting those that are required and those that are not. Software companies can take advantage of highly customizable checkout pages by linking directly into the PG system.

Two integration methods are available for integrating Secure Web*Pay* with the Payments Gateway. The HTML method provides a simple, quick setup, while the Transaction Signing method is a little more involved, but provides more advanced features. Both of these methods are explained in more detail in Chapter 2.

What does integrating with Payments*Gateway provide?*

By using Payments*Gateway*, merchants can process sales (debits), refunds (credits), authorizations, funds capturing and verifications. Transactions recurring weekly, bi-weekly, monthly, bi-monthly, quarterly, semi-annually or annually, as well as single transactions, travel efficiently and rapidly through its channels.

In channeling authorizations, settlement and/or funding, Payments*Gateway* helps mitigate risk and reduce fraud. Services such as ATM*Verify*™, NCN*Verify*™ and ID*Verify*™ operate seamlessly with Payments*Gateway's Secure WebPay Platform.*

How to Use This Manual

This manual provides information needed to complete the integration process, including many *Best Practice Tips* learned through many years of helping customers with integration projects. The manual is intended for use by your technical team or a developer who has an understanding of and experience with the following:

- Basic programming skills
- Basic integration skills and formats

Due to many years of helping customers with integration projects, we suggest you determine which delivery method is appropriate for your integration. In the **Secure Web*Pay* Integration** chapter, each integration method is described, along with recommendations for their use and the pros and cons for each.

The **Message Definitions** chapter describes how to complete the second phase of the integration process: composing messages using message types and associated data fields. This chapter is an essential reference tool for any user setting up new messages, but we also recommend that you supplement this chapter by adding your own notes about the messages you create, what they mean within your organization, and what action is to be taken when they are encountered.

The **Testing** chapter describes how to test the messages you've composed, and the options and test methods available on the test server. The importance of complete and thorough testing cannot be overemphasized. It is the key to a positive Go-Live experience for your staff and will keep overall costs of integration low (in terms of time and effort spent on training and troubleshooting problems).

The **Going Live** chapter details the final steps in the integration process. When testing is completed, all future transactions can be directed to the production server. If testing has been thorough, this process will be smooth and problem free.

We offer the **Best Practices** chapter to provide your team with additional best practices information and suggestions about what types of documentation you should create and maintain for your system administrator and users.

The **Appendices** offer additional details and reference information. A **Glossary** is located near the end of the document to provide explanations of unfamiliar terms, and the **Index** is provided to make this guide easier to use.

Chapter 2 - Secure Web*Pay* Integration

The Integration Process

The Secure Web*Pay* integration process is used by:

- New customers who want to set up and integrate Secure Web*Pay* in to the Payments*Gateway*.
- Current users of Payments*Gateway* who wish to make changes to their delivery method or messages.

The integration process includes the following stages:

1. Define the Integration Method
2. Define Message Composition
3. Testing
4. Going Live

For assistance during integration, please contact please contact technicians at integration@paymentsgateway.com or by calling 888-235-4635, option 3. **After integration is complete**, contact Customer Service at the following number: 888-235-4635, option 2.

Tip: Project Management Best Practices

It has been our experience that the clients who have the easiest, most trouble-free integration projects usually assign an integration project manager. For this reason, we recommend that all clients assign someone to manage their integration project.

In large organizations project managers are normally assigned full-time to integration projects, but in smaller organizations, this is not necessarily so. For a smaller project, the project manager may be the same person who performs the setup work, or perhaps the manager of the technical team. The person assigned to take on the role of project manager should be able to:

- Create a comprehensive list of tasks to be completed
- Create a list of needed resources (either full- or part-time), and work with management to obtain those resources for the dates and durations needed
- Manage team members to ensure they complete all tasks on time
- Be available on a full time basis, if needed, during the testing and go-live phases of the integration project to ensure that:
  - All testing is complete and thorough
  - All staff members are trained on the new system
  - Go-Live is handled efficiently and is successful

Integration Methods

Merchant transactions may be submitted to the Payments Gateway via Secure Web Pay using various methods. The following table shows how Secure Web Pay works with our current integration methods.

	Secure Web Pay (SWP) HTML	Secure Web Pay (SWP) Transaction Signing	Advanced Gateway Interface (AGI)
Checkout Page Hosted on …	Payments Gateway	Payments Gateway	Merchant's Site
Technical Skills Required?	HTML	Secure Web Pay API	Integration Guide API
SSL Certificate Required	No	No	Yes

Table 2.1: Integration Methods

HTML Integration

The HTML Integration method is the easiest way to integrate to the Payments Gateway. Merchants build a standard webpage with a form that submits to the PG. This integration method requires less work to implement and offers **only** sale/debit type transactions which are good for donations, simple low-cost sales, etc. Example HTML code is as follows:

```
<FORM METHOD="post"
ACTION="https://www.Paymentsgateway.net/swp/default.aspx">
 <input name="pg_billto_postal_name_first" type="text" value="Bob"/>
 <input name="pg_billto_postal_name_last" type="text" value="Smith"/>
 <input type="hidden" name="pg_api_login_id" value="APILOGINID"/>
 <INPUT TYPE=SUBMIT>

</FORM>
```

HTML Code Example

Transaction Signing Integration

Transaction Signing Integration requires a server side technology in order to sign the message. If transaction signing is used, eCheck credits and Credit Card credits can be performed.
The transaction needs to be hashed with an HMAC-MD5 hash using the secure transaction key found in the Virtual Terminal.

Hashing Format for Transaction Signing

```
HMAC-MD5("pg_apilogin_id | pg_transaction_type | pg_version_number | pg_total_amount |
pg_utc_time | pg_transaction_order_number", “pg_secure_transaction_key” )
```

Transaction Signing Code Example

```
pg_billto_postal_name_first="Bob"
pg_billto_postal_name_last="Smith"
pg_api_login_id = APILOGINID
pg_transaction_type = 10
pg_version_number = 1.0
pg_total_amount = 5.00
pg_utc_time = 1178571294
pg_transaction_order_number = 100055
pg_secure_transaction_key = SECURETRANSACTIONKEY
```

Items to hash

HMAC-MD5("APILOGINID | 10 | 1.0 | 5.00 | 1178571294 | 100055", SECURETRANSACTIONKEY)

Advanced Gateway Integration (AGI)

Advanced Gateway Integration is the most difficult and does not fall under Secure Web Pay (SWP). Everything is done on the merchant’s side. The payment page is hosted on the merchant site instead of on the Payment Gateway. This integration method requires the use of an SSL certificate.

Chapter 3 - Message Definitions

This chapter describes how to format, create content and process Secure Web*Pay* messages. Generally, there are rules for formatting messages correctly and there are fields which may be used to create content. The correct combination of formatting and fields creates an acceptable message. To ensure your message is processed correctly, you must test them and have them certified before they can be moved to the "live" production server (and therefore made available for use in your system). Later, in the testing section, we'll explain how to access the system and enter the messages you compose here.

Type Indicators

In the Transaction Message Template table, shown on page 6, you will notice that each field contains a *Type* indicator. The entry in the Type column indicates the expected format of the field. For example, an "N" indicates a Numeric field. To gain an understanding of the data shown in the Transaction Message Template table, review the full list of value data types below.

TYPE	DESCRIPTION	CHARACTERS ALLOWED	CASE SENSITIVE
M	Money	0-9 (and an optional period)	N/A
N	Numeric	0-9 (no period)	N/A
A	Alphanumeric	Any printable ASCII	Yes
L	List-based value*	Value must be in the specified list	No
D	Date	DD/MM/YYYY	N/A
T	True/False*	"True" or "False" only	No

- ***List-based*** values refer to an additional table that lists acceptable values. The value used in the message must be one included in the value list.
- ***True/False*** fields are considered false if there is no indicator present in the *Type* field of the message.

Field Requirements

The Transaction Message Template also includes a *Required* column. Entries in this column indicate the "When" or "In What Circumstances" the fields may be used. For example, if there is an "M" in the Type column, the field's use is mandatory. Some fields may have the notation "C" for conditional. When this notation occurs, use of the field is explained in the description section that follows the table.

A list of value data types appears in the following table and may be used to interpret the tables in this chapter.

CODE	REQUIREMENT	DESCRIPTION
M	Mandatory	Must appear when the field is used.
O	Optional	May appear when the field is used.
C	Conditional	See description for exact requirements.
R	Response Only	Only appears in response messages.

Transaction Message Template

The following table contains the data fields that make up the transaction messages that are sent to the Payments*Gateway*. The fields are grouped in the table as follows: Header, Customer/Order Information, Recurring and Transaction Signing.

The table contains the following information for each data field:

- *Field Name* – Name of the value being submitted to the PG
- *Type* – Expected format of the field
- *Required* – Indicates if the field is required on the transaction
- *Description* – Provides additional details on how the field is used

TRANSACTION MESSAGE TEMPLATE				
FIELD GROUP	**FIELD NAME**	**TYPE**	**REQ'D**	**DESCRIPTION**
Header	pg_api_login_id	A10	M	API Login ID found in the Virtual Terminal
	pg_return_url	A100		URL of the page customer will be returned to
	pg_transaction_type	L	O	10= CC Sale 13 = CC Credit 20=eCheck Sale 23=eCheck Credit 11=CC Auth 21=eCheck Auth
	pg_continue_url	A100	O	URL of the page customer will be returned to if *Continue* is pressed
	pg_version_number	A3	M	Current Version is "1.0".
	pg_total_amount	M	M	Total amount of transaction to be charged/credited to customer
	pg_save_client	N	O	Save client info
Customer/ Order Information	pg_sales_tax_amount	M	O	*Sales Tax amount
	pg_consumer_id	A15	O	Consumer ID
	pg_client_id (return)	N	O	Client ID
	pg_payment_method_id (return)	N	O	Payment Method ID
	pg_consumerorderid (invoice_number)	A15	O	Invoice Number
	pg_walletid (description)	A15	O	Description
	pg_merchant_data_1	A40	O	
	pg_merchant_data_2	A40	O	
	pg_merchant_data_3	A40	O	
	pg_merchant_data_4	A40	O	
	pg_billto_postal_name_company	A20	O	Company Name
	pg_billto_postal_name_first	A25	M	First Name
	pg_billto_postal_name_last	A25	M	Last Name
	pg_billto_postal_street_line1	A35	O	Address1
	pg_billto_postal_street_line2	A35	O	Address2
	pg_billto_postal_city	A25	O	City
	pg_billto_postal_stateprov	A10	O	State
	pg_billto_postal_postalcode	A10	O	Zip Code

	pg_billto_telecom_phone_number	A15	O	Telephone Number
	pg_billto_online_email	A40	O	E-mail address
	pg_shipto_postal_name	A35	O	Ship to Name
	pg_shipto_postal_street_line1	A35	O	Ship to Address1
Customer/ Order Information	pg_shipto_postal_street_line2	A35	O	Ship to Address2
	pg_shipto_postal_city	A25	O	Ship to City
	pg_shipto_postal_stateprov	A10	O	Ship to State
	pg_shipto_postal_postalcode	A10	O	Ship to Zip Code
	pg_customer_ip_address	A80	O	
	pg_scheduled_transaction	T	O	0 = Not a schedule transaction 1 = scheduled transaction
Recurring	pg_schedule_quantity	N9	C (R)	
	pg_schedule_frequency	L	C (R)	10 = Weekly 15 = Bi-weekly 20 = Monthly 25 = Bi- monthly 30 = Quarterly 35 = Semi-annually 40 = Annually
	pg_schedule_start_date	D	C (R)	
	pg_secure_transaction_key		M	Secure Transaction Key found in the Virtual Terminal. Required for Transaction Signing.
Transaction Signing	pg_api_login_id		M	API Login ID in the Virtual Terminal
	pg_transaction_type	L	M	10= CC Sale 13 = CC Credit 20=eCheck Sale 23=eCheck Credit 11=CC Auth 21=eCheck Auth
	pg_version_number	A3	M	Current Version 1.0
	pg_total_amount		M	Total amount of transaction to be charged/credited to customer
	pg_utc_time		M	UTC time in ticks
	pg_transaction_order_number		M	Random number identifying the transaction.
	pg_ts_hash		M	Hashed field

Field Name Notes:

pg_api_login_id: The API login found in the Virtual Terminal

pg_return_url: This is the page the customer will be returned to after a transaction has been completed. This page should contain server side script to parse the data being posted to it, however, it can be a static HTML page as well. If this field is not set or invalid, the entire transaction is done on the Payments Gateway. In order to be a valid return URL the value sent must match at least one of the values in the Virtual Terminal.

Note: Secure Web Pay will post all but the following values to the return URL page.

- API Login ID
- Secure Transaction Key
- Credit Card Number
- Card Holder Name
- Card Exp Date
- Procurement Acct Code
- CCV
- eCheck Account Number
- eCheck TRN

pg_transaction_type: This field is optional for HTML integration. If it is not sent the customer will have both sale options available depending on the permissions in Virtual Terminal. This field is required when using the Transaction Signing Method.

- 10 = Credit Card Sale – Customer's card is charged and will be automatically settled at the end of the day.
- 11 = Credit Card Authorization – Customer's card is charged but will not be settled until a Capture message is completed.
- 13 = CC Credit* - Customer's credit card is credited and will be automatically settled at the end of the business day.
- 20 = eCheck Sale – Transaction is completed and the funds will be captured at the end of the day.
- 21 = eCheck Authorization – Transaction is authorized, but the funds are not captured until a Capture message is completed.
- 23 = eCheck Credit* - Transaction is completed and the funds will be transferred at the end of the day.

**Transaction must be signed*

pg_continue_url: After any payment has been completed *Pass* or *Fail*, this will be the URL directed to when the *Continue* button is pressed. If a valid return URL has been specified this field is ignored.

pg_version_number: Current version is "1.0".

pg_save_client: Determines if a client or payment method should be created. If not passed nothing is created for this transaction. If created the pg_payment_method_id and/or pg_client_id are returned. These values can be used in the transaction or CMI web services.

- 1 = Saves payment method
- 2 = Saves client and payment method.

pg_sales_tax_amount: Sales tax amount of the transaction. This is an API only field, meaning it is a read only field on the payment page.

pg_consumer_id: This field identifies the customer and can be searched on in the Virtual Terminal.

pg_client_id: If a client is created this ID will be posted back.

pg_payment_method_id: If a payment method is created this ID will be posted back.

pg_consumerorderid (Invoice_number): Invoice number of the transaction. Shows up in the Virtual Terminal as Consumer Order ID or your labeled field.

pg_walletid: Description of the transaction. Shows up in the Virtual Terminal as wallet id or as your labeled field.

pg_secure_transaction_key: The secure transaction key found in the Virtual Terminal. Please treat this as you would a password and change often. Do not store in an insecure area.

pg_utc_time: UTC time in ticks.

pg_transaction_order_number: Any random number identifying the transaction.

pg_ts_hash: The hash generated by the transaction signing fields.

Recurring Transaction Templates

Recurring fields are used to establish a recurring transaction. Transactions will be created and processed at the stated frequency (as long as the recurring transaction is in an 'active' state). The transactions will be created and processed until the specified quantity is reached (if it is non-zero) or until the transaction is suspended or deleted by the merchant. Voided and declined transactions do not count towards the specified quantity.

Recurring Transaction Fields

The following fields are used for processing recurring transactions:

- **pg_scheduled_transaction:** 0 = Not a schedule
  1 = Scheduled transaction
  **Note:* Default = 0

- **pg_schedule_quantity:** specifies the number recurring transactions

- **pg_schedule_frequency:** specifies the frequency of the recurring transaction. Please note **Table 3.4** below.

- **pg_schedule_start_date:** Specifies start date of the next recurring transaction (MM/DD/YYYY).

Note:
(R) – recurring transactions must have both pg_schedule_quantity and pg_schedule_frequency, but pg_schedule_recurring amount and pg_schedule_start_date are optional.

VALUE	FREQUENCY	PERIOD
10	weekly	every seven days
15	biweekly	every fourteen days
20	monthly	same day every month
25	bi-monthly	every two months
30	quarterly	every 3 months
35	semiannually	twice a year
40	yearly	once year

Table 3.4 Frequency Values for Recurring Transactions

Chapter 4 - Testing

Previous chapters detailed the message fields, their use in various message types and the message delivery method. This chapter provides the final pieces of information you need to actually setup, test, certify and bring your system up live with your messages.

When you perform testing on the PG system:

- Our servers are available 24x7.

- You can build and test at your own pace.

- Example codes are provided for your use.

- There are “canned” or preset responses for some messages so you’ll know what response indicates a successful transaction when conducting testing (listed in Appendices).

When testing is complete, messages are ready to be placed in a live production environment and the system is ready for operation.

How to Prepare for Testing

In previous chapters, there was no discussion of "hands on" work, or instructions for how to sign on to the PG system, because the task of composing messages should be worked out before entering them into the system.

When you have composed messages you wish to enter into the PG system, you will need a system sign on, which includes a Merchant ID and password.

URL

The URL for the Secure Web*Pay method* is listed below.

	https://swp.paymentsgateway.net/default.aspx

Figure 4.1-Parameters for Test Transactions

Differences Between Test and Live Servers

The test and live servers are virtually the same. The major differences are listed below.

- Test CC transactions are run through the authorizing vendors test system
- Test CC transactions are never settled
- Test EFT transactions are never settled
- Test recurring transactions are never processed

Chapter 5 - Going Live

"Going live" or "Go-live" means that your system is ready to work in a production environment. From a technical standpoint, this step involves a minor change to the delivery method and a call to us to have your live account set up.

Of course, from a logistical standpoint and the point of view of end users, go-live can be a stressful experience if testing has not been conducted thoroughly and adequate training has not been provided. We recommend that you involve the integration project manager and any education/training personnel involved with the project to coordinate schedules and efforts for go-live.

Are You Ready?

If you have not yet tested all messages you have created, we encourage you to STOP and go back to the testing stage. Time spent on testing will be repaid in terms of time not spent trying to explain what went wrong to every user on the system during go-live.

If training has not been conducted related to messages that users are unfamiliar with, we recommend you STOP and ensure that all users receive instructions on the meaning of messages

that have been created and any new procedures for dealing with those messages.

Chapter 6 - Best Practices

This chapter summarizes best practices for integrating and maintaining the PG system.

Tools Available to Help You

Integration Support	Support for customers currently undergoing integration, or needing assistance with integration or testing issues integration@paymentsgateway.com.
Software Downloads	To make updating your software easier than ever http://www.achdirect.com/eval/sec/login.aspx.
Merchant Training	To ensure a successful implementation project http://www.achdirect.com/eval/sec/login.aspx.

Obtaining Help from Customer Service

We often receive calls from a member of a client's staff who needs help with a transaction. While we are always happy to help our customers, there are specific pieces of information we must have to provide assistance. Please train your employees to have the following information on hand when they contact Customer Service for help:

- Merchant ID
- Date of transaction in question
- Amount of transaction
- Name of purchaser

Reconciliation is Critical for EFTs

The reconciliation process is your responsibility and it is an often-overlooked process which can be costly if not done properly.

With credit card transactions, you know immediately whether those transactions will be paid. However, with ACH and EFT transactions, settlement will not occur for a minimum of 3-4 days and chargebacks can occur for up to 90 days.

For this reason, it is very important that you reconcile settlement information with your authorizations on a regular basis. You may obtain your settlement information from PaymentsGateway and compare it to your authorizations. If you are pulling down EFT settlement info, a match against transaction results is a good way to ensure accuracy.

Settlement files are available for download from the PaymentsGateway web site. Please see the File Specification Document available for download from the developer's repository at https://www.paymentsgateway.net/development/login.asp or in the Virtual Terminal Knowledge Center section.

Documentation is the Key to Easier Maintenance

Even if you have the perfect, trouble free integration and go-live, it is critical that you document what you did carefully. Why? So that those having to maintain the system after you will understand what you did and why you did it. You may plan to do the maintenance yourself, but if you are unavailable when a key change must be made immediately, good documentation will allow the needed changes to be made correctly.

Generally, you should document the following:

Delivery Method: Document why the delivery method is selected, the thought process that leads you to select that method, who approves it and the date of the approval.

Messages: Document the business purpose of each message, any alternate drafts considered, who approves the message and the date approved.

Testing: Document the test methods you use (including any test scripts), who participates in the tests, who approves the test results, and dates.

Certification: If any changes are made to messages as a result of certification testing, be sure to adjust the documentation. If staff training is delivered during this phase, archive copies of the training materials. Also, you may wish to document who is trained and on what dates.

Go-live: Document all problems encountered during go-live (if any). Document individuals having problems with particular parts of the system (even if they were trained on how to use it, because there are occasional misunderstandings during training classes).

What is the reason for all of this documentation? In a few months when a problem occurs, you'll know what was encountered when the system was being integrated, tested or during go-live and whether users were ever trained on that topic. You will then know how to go about researching the problem and contacting the users who need to understand about the correction, or who may need additional information. Documentation seems to some an unnecessary chore, but the resulting tools can make finding problems in the future faster and easier (and therefore much cheaper).

Ask Questions

During the integration process or after you are running on a "live" system, be sure to ask questions when you don't understand something or need a clarification. Please contact us; don't guess or assume. We're here to help you!

Appendix A: Response Codes

Updated lists of codes, sample files and other information located in these appendices may be found on the Developer web site:
https://www.paymentsgateway.net/development/login.asp

Please consult the web for the most updated version of this guide and supplemental materials. If you do not yet have a developer's login, contact your Customer Support Representative for assistance.

The ***pg_response_code*** values returned in the response message are listed in the following three tables.

Approved and Declined Responses

These responses are returned for all processed transactions. The A01 response is the only code ever returned for approved transactions. The "U" codes are for declined transactions. In some cases the *pg_response_description* field value will differ from that in the "description" column.

Code	Description	Comments
A01	APPROVED	Transaction approved/completed
U01	MERCH AUTH REVOKED	Merchant not allowed to access customer account (EFT only)
U02	ACCOUNT NOT APPROVED	Customer account is in the "known bad" account list (EFT only)
U03	DAILY TRANS LIMIT	Merchant daily limit exceeded (EFT only)
U04	MONTHLY TRANS LIMIT	Merchant monthly limit exceeded (EFT only)
U05	AVS FAILURE ZIPCODE	AVS state/zipcode check failed
U06	AVS FAILURE AREACODE	AVS state/area code check failed
U07	AVS FAILURE EMAIL	AVS anonymous email check failed

Code	Description	Comments
U10	DUPLICATE TRANSACTION	Transaction has the same attributes as another transaction within the time set by the merchant
U11	RECUR TRANS NOT FOUND	Transaction types 40-42 only
U12	UPDATE NOT ALLOWED	Original transaction not voidable or capture-able
U13	ORIG TRANS NOT FOUND	Transaction to be voided or captured was not found
U14	BAD TYPE FOR ORIG TRANS	Void/capture and original transaction types do not agree (CC/EFT)
U15	ALREADY VOIDED ALREADY CAPTURED	Transaction was previously voided or captured
U18	UPDATE FAILED	Void or Capture failed
U19	INVALID TRN	Account ABA number if invalid
U20	INVALID CREDIT CARD NUMBER	Credit card number is invalid
U21	BAD START DATE	Date is malformed
U22	SWIPE DATA FAILURE	Swipe data is malformed
U23	INVALID EXPIRATION DATE	Malformed expiration date
U25	INVALID AMOUNT	Negative amount
U26	INVALID DATA**	Invalid data present in transaction
U51	MERCHANT STATUS	Merchant is not “live”
U52	TYPE NOT ALLOWED	Merchant not approved for transaction type (CC or EFT)
U53	PER TRANS LIMIT	Transaction amount exceeds merchant’s per transaction limit (EFTs only)
U54	INVALID MERCHANT CONFIG	Merchant’s configuration requires updating – call customer support
U80	PREAUTH DECLINE	Transaction was declined due to preauthorization (ATM Verify) result
U81	PREAUTH TIMEOUT	Preauthorizer not responding (Verify Only transactions only)
U82	PREAUTH ERROR	Preauthorizer error (Verify Only transactions only)
U83	AUTH DECLINE*	Transaction was declined due to authorizer declination
U84	AUTH TIMEOUT	Authorizer not responding

Code	Description	Comments
U85	AUTH ERROR	Authorizer error
U86	AVS FAILURE AUTH	Authorizer AVS check failed
U87	AUTH BUSY	Authorizing Vendor busy, may be resubmitted (CC only)
U88	PREAUTH BUSY	Verification vendor busy, may be resubmitted (type 26 only)
U89	AUTH UNAVAIL	Vendor service unavailable (CC only)
U90	PREAUTH UNAVAIL	Verification service unavailable (type 26 only)

Table A.1 - Approved and Declined Responses

**pg_response_description* will contain the text of the vendor's response

***pg_response_description* will contain a more specific message

Formatting Error Responses

These are the codes returned when formatting errors are found. The response description field will actually list all offending fields in the message (to the 80 character limit). The description field will be formatted as:

<code>:<fieldname>[,<code>:<fieldname> …]

The *pg_response_code* will contain the first error type encountered. All formatting errors begin with "F".

Code	Description	Comments
F01	MANDITORY FIELD MISSING	Required field is missing
F03	INVALID FIELD NAME	Name is not recognized
F04	INVALID FIELD VALUE	Value is not allowed
F05	DUPLICATE FIELD	Field is repeated in message
F07	CONFLICTING FIELD	Fields cannot both be present

Table A.2- Formatting Error Codes

Fatal Exceptions Responses

These exceptions will stop the processing of a well-formed message due to security or other considerations. All fatal exceptions begin with an "E."

Code	Description	Comments
E10	INVALID MERCH OR PASSWD	Merchant ID or password in incorrect
E20	MERCHANT TIMEOUT	Transaction message not received (I/O flush required?)
E90	BAD MERCH IP ADDR	Originating IP not on merchant's approved IP list
E99	INTERNAL ERROR	An unspecified error has occurred

Table A.3 – Fatal Exceptions

Glossary

ACH

Automated Clearing House is a national network for batch-oriented electronic funds transfer. ACH transactions are governed by NACHA operating rules, and provide a method for transferring funds between banks using the Federal Reserve System. Most (but not all) financial institutions use the ACH network.

Types of ACH payments include:

- Direct deposits of all types including tax refunds, payroll and government benefits (like Social Security)
- Direct payment of bills such as utilities, mortgages, loans and insurance policies
- Federal, state and local tax payments
- Business-to-business payments
- E-checks
- E-commerce payments.

API Login

API Login ID is the assigned ID for accessing the API. This ID is needed for using Secure Web*Pay*. It can be found in the Virtual Terminal.

Approval

An approval is any transaction approved by the credit provider or the check writer's bank. Approvals are granted after an authorization has been requested by a merchant.

ATM Verify™

A risk management tool that fights against bad check writers by verifying in real time if a checking account currently has sufficient funds to cover the amount of a presented check and if the account is, as of opening of that business day, an open, valid account with positive funds or if the account is currently NSF, closed or cannot be located.

Authorization

Only used for credit card transactions, an authorization is a request from a merchant to charge a cardholder. If approved, the authorization will decrease the customer's available credit, but will not actually capture any funds. An authorization is the first step in the *delayed settlement process*, where the merchant may obtain an approval, but it is not settled within a specific period of time, the authorization will expire. The credit provider determines the delay period.

Authorization Code

Numeric or alphanumeric code issued by the credit provider and used to reference the authorization.

Auth Only

In this type of authorization, the merchant does not intent to capture funds until a later date. Often, funds are not captured on these authorizations.

Capture

Refers to the "capture" of funds at the end of a transaction. This typically follows "settlement" of the transaction, where the amount is actually debited to the customer's account.

COM

Acronym for Component Object Model, COM is a software architecture created by Microsoft and used as a basis for their interprocess communication. COM is language- independent, works within object-oriented programs/designs and is extremely versatile.

In the PG system, COM is an accepted method for data posts, though not the preferred method. DSI is our preferred method (see below).

Decline

A transaction which is not approved by the credit provider/issuer. No authorization is issued.

EFT

Electronic Funds Transfer (EFT) provides for electronic payments and collections. EFT is safe, secure, efficient, and less expensive than paper check payments and collections. EFT is the preferred method of payment for the government. As stated by the Treasury web site "it costs the U.S. government $.83 to issue each check payment, it costs only $.08 to issue an EFT payment."

Full Gateway Integration

One method that may be used to integrate into the Payments Gateway. It is the most difficult method, does not fall under Secure Web*Pay* and requires the use of an SSL certificate.

HTML Integration

The easiest way to integrate to the Payments Gateway where merchants build a standard webpage with a form that submits to the PG. This method offers only sale/debit type transactions.

IDVerify™

Assists businesses in determining, in real time, the likelihood that the person being spoken to is who they claim to be by generating a question and the corresponding answer. The correct person answers quickly and easily whereas the identity thief . Questions are generated from an aggregate of data sources including the Social Security Administration, regional Bell operating companies, credit header data, government watch lists and many other proprietary sources.

Merchant ID

This is the identification number for your organization, used by Payments*Gateway* to identify you in all communications. It is critical that anyone contacting Customer Service or Technical Support for assistance know this ID number.

NACHA

National Automated Clearing House Association. A body which develops and maintains the

NACHA Operating Rules and oversees all ACH activities and procedures. NACHA is also responsible for the sale and distribution of payment-related publications and providing national education.

NCNVerify™

A negative check database services that can determine whether or not a specific checking account has:

- dishonored items currently pending with a participating merchant
- a "positive" check-writing history with merchants supplying data to the National Check Network
- not been seen by National Check Network before, alerting you that the account in question may be new or simply not exist.

Payments*Gateway*

A high-capacity modular payment processing platform designed for maximum flexibility and availability.

PCI

Payment Card Industry. A measure that mandates merchants and service providers to meet certain minimum standards of security when they store, process and transmit cardholder data. PCI audits are performed on the Payments Gateway side to ensure a secure and safe transaction.

Pre Auth

See Auth Only – same meaning.

Pre Notification

Prior to sending the first ACH transaction to an ACH receiver or the ACH receiver's account, the ACH originator may (optionally) send a pre-notification to be processed to the customer's account. This provides notice of the intent to send additional items and the date on which they will be drafted from the customer's account.

Procurement Card

Similar to credit cards and gift cards, procurement cards are typically issued by organizations to enable employees to purchase supplies or items for company use.

RAW

In computer terminology, this refers to unprocessed data. This term came originally from the UNIX platform and generally refers to data that is passed along without being interpreted or processed in any way.

The PG system will accept RAW HTTP data posts, but this is not a preferred method. We strongly recommend users adopt the DSI or Windows methods for posting data.

Recurring Transaction

A transaction that occurs more than once. Messages are created to determine how many recurring transactions occur, the frequency and the start date.

Reversal

If a transaction has already settled and should have been voided, it can be reversed by issuing a credit to correct the error.

Secure Web*Pay*

A method of submitting electronic transactions to the Payments Gateway with relatively minimal ease and security.

Secure Transaction Key

Found in the Virtual Terminal, the secure transaction key is used as an additional layer of authentication when submitting transaction requests from your Web site. The Transaction key should be treated as a password, i.e. securely stored and changed often. It is only needed for transaction signing.

Settlement

In this process, authorized transactions are sent to the processor for payment to the merchant. This process finalizes the transaction and allows funds due the merchant to be "captured" and routed to the merchant's bank for deposit. (In other words, the merchant cannot be paid until the transaction is settled.) It can take several days for funds to reach settlement. Credit card settlement may be within one day, while settlement for checks may take up to 90 days.

SSL

SSL is an acronym for Secure Sockets Layer, a communications protocol used to transmit private documents or information via the Internet. SSL encrypts data using a private key that is transferred over the SSL connection. Web sites that require an SSL connection have an address that begins with *https://* rather than *http://*.

Void

To void a transaction is to cancel one that has been authorized, but not yet settled. Settled transactions may not be voided, they must be reversed.