Advanced Gateway Interface

Integration Guide v3.13

Updated 8.31.2017

forte

Revision History

Version	Date	Changes
3.03	10/01/2016	Added the following token fields: • `pg_customer_token` (string tokens) • `pg_client_id` (integer tokens) • `pg_payment_token` (string tokens) • `pg_payment_method_id` (integer tokens)
3.04	10/13/2014	Branding Update
3.05	11/19/2014	Removed references to the Windows COM Object integration method
3.06	12/29/2014	updated description of Magtek IPAD device #30050203
3.07	01/13/2015	Added the following to the Forte Verify Response Code table: • `P16` • `P72`
3.08	3/30/2015	Deleted pg_onetime_token from EFT Transaction section of the Message Template.
3.09	5/19/2015	Deleted decommissioned 3DS fields.
3.10	07/02/2015	Added note about `ecom_consumerorderid` field.
3.11	09/21/2015	Deleted DSI integration method and added the Transaction POST SOAP Web Service.
3.12	02/18/2016	Updated the character limits of the merchant-defined fields (i.e., pg_merchant_data_1-10).
3.13	08/31/2017	• Re-added pg_onetime_token parameters to both the Credit Card and EFT Transaction section of the Message Template. • Deleted an inaccurate note from the Message Template regarding the changes from version 3.03.

Table of Contents

Introduction

Overview

The Forte Payment Systems platform
1. Captures purchase information via swipe or key entry
2. Processes credit card, EFT, and recurring transactions
3. Automatically responds to your point-of-sale machine approving or denying the transaction
4. Uploads completed transaction information to Forte's Virtual Terminal application (www.paymentsgateway.net/vt3/login.aspx)

Purpose

This guide provides detailed instructions for integrating your point-of-sale system with the Forte platform.

Audience

This guide is intended for technical team members, such as developers, who have experience with the following:
- Basic programming skills
- Basic integration skills and formats
- Your in-house swipe cards system's formats and protocols
- SSL, SOAP, or RAW HTTP Data methods of data transmission

Assistance

For Help. . .	Call. . .	At. . .
During Integration	Forte Technical Support	888-235-4635 option 3
After Integration	Forte Customer Service	800-337-3060 option 1

Recommendation: Assigning a Project Manager

Forte recommends you assign a project manager to your integration project to manage the necessary setup tasks and connectivity testing. This person should be able to
- create a comprehensive list of tasks to be completed
- create and obtain a list of resources for the dates and durations of the project
- manage team members to ensure tasks are completed on time and on budget
- be available full-time during testing and go-live phases to ensure
    - all testing is complete
    - all staff members are trained on the new system

Integration Overview

Integration Steps

Forte AGI integrations occur in four steps.

Step	Name	Description
1	Define Delivery Method	Decide which delivery method is most compatible with your current setup: • RAW HTTP Post • Transaction SOAP POST
2	Compose the Message	Begin creating messages using message types and associated data fields.
3	Test	To ensure a successful integration, thoroughly test the messages you have created with the options and methods available on the test server.
4	Go-Live	Move all data from the test server to the live server.

Defining a Delivery Method

Available Delivery Methods

Choose one of two methods for your AGI integration.

Method	Description
RAW HTTP POST	A method that uses the HTTP POST protocol to securely deliver messages. This method should only be used if you • cannot do SSL operations • do not run on a Windows™ platform
SOAP Web Service POST	Use this method for creating payment transactions using a Simple Object Access Protocol (SOAP) web service.

RAW HTTP POST Method Description and Sequence

The RAW HTTP POST protocol is intended for

- non-Windows™ merchants
- merchants unable to perform SSL operations
- merchants with access to HTTPS routines

All transactions are routed through the Forte web server in the following sequence of steps:

Step	Description
1	URL encodes the field values (to escape special characters).
2	Concatenate message into an ampersand delimited string.
3	Set the message to be passed as the "content" resource.
4	Perform the POST (URL provided to approved merchants).
5	Forte web server returns newline delimited response message (not HTML).

NOTE: This method sends the POST messages from the merchant's server and not from the customer's browser.

SOAP Web Service POST

The SOAP web service supports two Operations:

- **ExecuteSocketDelimitedQuery**: Accepts "strParameters" and "strDelimiter" parameters
- **ExecuteSocketQuery**: Accepts Name Value Pairs

Other Methods

If you cannot use one of the two delivery methods described above, contact Forte Technical Support to discuss other integration options.

Composing Messages

Overview

The following sections describe how to format, create, and process messages. Creating acceptable messages requires a correct combination of formatting and content fields. These messages should be tested by you and certified by Forte before they can be moved to a production server.

Recommendation: Documenting Messages

As you create messages, Forte recommends you document the purpose, follow-up procedures, and content approvals associated with each message. This documentation will

- make future maintenance easier
- help in employee training

Formatting

Forte platform messages are comprised of name/value fields pairs in the `name=number` format (e.g., a merchant ID would look like `pg_merchant_id=1000`).

Messaging fields

- are always ASCII text (i.e., no binary data)
- can be placed in any order
- must be separated by an ampersand character when using other delivery methods

Messages should use the appropriate delimiting character after the tag.

Formatting Examle

The following is an example message:

```
pg_merchant_id=1000
pg_password=abc123
pg_transaction_type=20
pg_total_amount=1.00
ecom_billto_postal_name_first=John
ecom_billto_postal_name_last=Smith
ecom_payment_check_account_type=S
ecom_payment_check_account=012345
ecom_payment_check_trn=123456789
pg_merchant_data_1=just a test
endofdata
```

Understanding the Message Template

Codes for Field Types

The following table lists the type codes for the fields described in this chapter (in the Transaction Message Template section and the Credit Card and EFT Transaction sections).

Type	Description	Characters Allowed	Case Sensitive?
M	Money	0-9 (and an optional period)	—
N	Numeric	0-9 (no period)	—
A	Alphanumeric	Any printable ASCII	Yes
L	List-Based Value	Value must be in the specified list	No
D	Date	Format: DD/MM/YYYY	—
T	True/False	"TRUE" or "FALSE" only	No

NOTE: 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.

Codes for Field Requirements

The following table lists the requirement codes for the fields described in this chapter (in the Transaction Message Template section and the Credit Card and EFT Transaction sections).

Code	Requirement	Description
M	Mandatory	Must appear when table's fields are used
O	Optional	May appear when table's fields are used
C	Conditional	See description for exact requirements
R	Response Only	Only appears in response message

Continued

Understanding the Message Template (Cont'd)

Transaction Message Template

The following groups of fields make up the template for the core portion of the transaction message.

Group	Field Name	Description	Type	Req
Header	pg_merchant_id	The merchant's six-digit ID code	N8	M
	pg_password	The merchant's processing password	A20	M
	pg_transaction_type	Indicates the Transaction type	L	M
	pg_merchant_data_(1-10)	Ten fields returned with the response fields	A255	O
Customer/ Order Info	pg_total_amount	Amount to be charged/credited to the customer	M	M
	pg_sales_tax_amount	Sales tax amount.	M	C(PC)
	pg_consumer_id	ID assigned by the merchant and returned with the response	A15	O
	ecom_consumerorderid (Note 1)	ID assigned by the merchant and returned with the response	A36	O
	ecom_walletid	ID assigned by the merchant and returned with the response	A15	O
	pg_customer_token	Unique string ID that references a customer's stored information	A50	O
	pg_client_id	Unique integer ID that references a customer's stored information. Returned with response.	N50	O
	pg_billto_postal_name_company	Company name	A20	O
	ecom_billto_postal_name_first	Customer's first name	A25	M
	ecom_billto_postal_name_last	Customer's last name	A25	M
	ecom_billto_postal_street_line1	Customer's street address	A35	C(AVS)
	ecom_billto_postal_street_line2	Customer's street address	A35	O
	ecom_billto_postal_city	Customer's city	A25	O
	ecom_billto_postal_stateprov	Customer's state (abbreviated)	A10	C(AVS)
	ecom_billto_postal_postalcode	Customer's ZIP code	A10	C(AVS)
	ecom_billto_postal_countrycode	Customer's country	A2	O
	ecom_billto_telecom_phone_number	Customer's phone number	A15	C(AVS)
	ecom_billto_online_email	Customer's email address	A40	C(AVS)
	pg_billto_ssn	Customer's Social Security Number	A11	O

NOTE 1 – Some credit card authorizers/settlement vendors require the ecom_consumerorderid field to be numeric for qualification. For those vendors, Forte normalizes and randomly assigns an integer value to the field.
(PC) – Indicates a field that is required for procurement card transactions and optional otherwise.
(AVS) – Indicates a field that is required for AVS checks specified in the pg_avs_method field and optional otherwise.

Continued

Understanding the Message Template (Cont'd)

Transaction Message Template

Group	Field Name	Description	Type	Req
Customer/ Order Info	pg_billto_dl_number	Customer's Driver's License number	A20	O
	pg_billto_dl_state	Customer's Driver's License state	A2	O
	pg_billto_date_of_birth	Customer's date of birth in MM/DD/YYYY format	D	O
	pg_entered_by	Name or ID of the person entering the data; appears in the Virtual Terminal transaction display window	A20	O
Recurring	pg_schedule_quantity	Specifies the number of recurring transactions	N9	C(R)
	pg_schedule_frequency	Specifies the frequency of the recurring transactions. Use the following values: 10 = Weekly (Every 7 Days) 15 = Bi-Weekly (Every 14 Days) 20 = Monthly (Same day every month) 25 = Bi-Monthly (Every 2 months) 30 = Quarterly (Every 3 months) 35 = Semi-Annually (Twice a year) 40 = Yearly (Once a year)	L	C(R)
	pg_schedule_recurring_amount	Specifies the amount of the recurring transaction if different from the initial transaction.	M	C(R)
	pg_schedule_start_date	Specifies the start date of the next recurring transaction in the MM/DD/YYYY format and may only be used with the pg_schedule_recurring_amount field	D	C(R)
Misc.	pg_customer_ip_address	Customer's originating IP address (used for fraud prevention)	A80	O
	pg_merchant_recurring	When used in conjunction with CC transactions, a recurring indicator will be included with the authorization message to the issuer, which may affect qualification	T	O
	pg_software_name	Name of the software application used to create the transaction	A20	O
	pg_software_version	Version of the software application used to create the transaction	A20	O
	pg_avs_method	Specifies which AVS checks to perform on the transaction (if any); makes some optional fields required.	N5	O
Credit Card Transactions	pg_payment_token	Unique string ID that references a customer's stored payment information. Returned with response.	A50	O
	pg_payment_method_id	Unique integer ID that references a customer's stored payment information. Returned with response.	N50	O

(R) – Indicates a field that is required for recurring transactions and optional otherwise.

Continued

Understanding the Message Template (Cont'd)

Transaction Message Template

Group	Field Name	Description	Type	Req
Credit Card Transactions	pg_onetime_token	A one-time-use token that references stored payment information.	A50	O
	ecom_payment_card_type	The credit card issuer. Use one of the following values: Type / Issuer VISA / VISA MAST / MasterCard AMER / American Express DISC / Discover Card DINE / Diner's Club JCB / JCB	L	M
	ecom_payment_card_name	Cardholder name as it appears on the card	A50	M
	ecom_payment_card_number	Card account number	N16	M
	ecom_payment_card_expdate_month	Numeric month of expiration (e.g., January = 1)	N2	M
	ecom_payment_card_expdate_year	Four-digit year of expiration	N4	M
	ecom_payment_card_verification	CVV2/verification number	N5	O
	pg_procurement_card	Indicates procurement card transaction, requires pg_sales_tax and pg_customer_acct_code fields	T	O
	pg_customer_acct_code	Accounting information for procurement card transactions	A17	C(pc)
	pg_cc_swipe_data	Magstripe data from track one or two	A80	O
	pg_cc_enc_swipe_data	Full set of swipe data received from the encrypting device	A1500	O
	pg_cc_enc_decryptor	The eight-digit device part number in parenthesis below specifying which swipe device was used. Currently only the following models and part numbers are supported when capturing encrypted card data: • IPAD (30050203) • Dynamag (21073062) • iDynamo - used for iPhone mobile apps (21073084) • uDynamo - used for Android mobile apps (21073092)	L20	O

(PC) – Indicates a field that is required for procurement card transactions and optional otherwise

Continued

Understanding the Message Template (Cont'd)

Transaction Message Template

Group	Field Name	Description	Type	Req
Credit Card Transactions	pg_partial_auth_allowed_flag	For merchants approved to process partial authorizations, set this field to override default merchant settings. Merchant accounts are generally provisioned with partial authorizations defaulted to off but can be defaulted to on by contacting Forte's Customer Service Department. Supported authorizing vendors: GlobalPayments and FirstData	T	O
	pg_mail_or_phone_order	Indicates mail order or phone order transaction (as opposed to an online transaction)	T	O
EFT Transactions	pg_payment_token	Unique string ID that references a customer's stored payment information	A50	O
	pg_payment_method_id	Unique integer ID that references a customer's stored payment information	N50	O
	pg_onetime_token	A one-time-use token that references stored payment information.	A50	O
	ecom_payment_check_trn	Transit routing number (ABA) for customer's account	N9	M
	ecom_payment_check_account	Customer's account number	N17	M
	ecom_payment_check_account_type	Type of bank account. Use one of the following values: • s = Savings • c = Checking	L	M
	ecom_payment_check_checkno	Check number for point-of-sale transactions	N10	O

Continued

Understanding the Message Template (Cont'd)

Transaction Message Template

Group	Field Name	Description	Type	Req
EFT Transactions	pg_entry_class_code	The standard entry class code. Use one of the following values: • ARC • CCD • CIE • CTX • POP • POS • PPD • RCK • TEL • WEB **NOTE:** If the entry class code is not specified, the pg_entry_class_code will default to PPD or, if established, the override entry class code value within the Merchant Setup. Specify the proper entry class code for each transaction. Improper entry class code usage can result in fines for NACHA violations or hurt the merchant's ability to prevent items from being returned (charged back) in customer dispute situations.	A3	O
Admin Message	pg_merchant_id	The merchant's six-digit ID code	N8	M
	pg_password	The merchant's processing password	A20	M
	pg_transaction_type	Indicates the Transaction type	L	M
	pg_merchant_data_[1-10]	Ten fields returned with the response fields	A255	O
	pg_original_trace_number	The trace number returned by the original transaction to be affected	A36	M
	pg_original_authorization_code	The authorization code returned with the above trace number (voids and captures only)	A80	C(AC)

(AC) – The pg_original_authorization_code field is only required for credit card and EFT capture and void transactions.

Working with Recurring Transactions

Overview

The fields listed in the Recurring group of the Transaction Message Template establish recurring transactions by specifying the frequency and quantity of a recurring transaction. These transactions occur until

- the specified quantity is reached (if it is non-zero)
- the transaction is suspended or deleted by the merchant

NOTE: Voided and declined transactions do not count toward the specified quantity.

Use Cases

These fields provide the flexibility to accommodate a number of use cases. For example, if the customer wanted to specify a set number of recurring transactions after making an initial transaction (such as a down payment), the merchant would specify the `pg_total_amount` field for the initial transaction and use the `pg_scheduled_recurring_amount` field to define the amount for the subsequent recurring transactions (the initial transaction does not count toward the specified quantity).

To define when the first recurring transaction should occur, use the `pg_schedule_start_date` field with the `pg_schedule_recurring_amount` field. If the start date is on or before the day the initial transaction is processed, the next start date will be the following day.

Requirments for Recurring Transactions

Recurring transactions submitted like the use case described above depend upon an approved initial transaction. Forte will not schedule a recurring transaction if the original transaction is declined. Additionally, a merchant cannot use this method to create recurring transactions that do not begin at the time of submission. Setting the initial amount to less than $1 (including $0) results in a decline from most credit card processors/banks.

Example 1: Simple Monthly Payments

If the transaction is approved, the code displayed below will result in eleven more $10 transaction being processed on the same day of the month that the initial transaction was approved.

```
pg_total_amount=10.00
pg_schedule_quantity=12
pg_schedule_frequency=20
```

Example 2: Different Recurring Amount and Deferred Recurring Start Date

The following code displays a recurring transaction with a different initial transaction amount and a specified start date. If the initial $100 transaction is approved, eight more $25 transactions will be processed monthly beginning on 6/1/2005.

```
pg_total_amount=100.00
pg_schedule_quantity=8
pg_schedule_frequency=20
pg_schedule_recurring_amount=25.00
pg_schedule_start_date=06/01/2005
```

Understanding the Response Message Template

Response Message Template

The following table lists the fields that may appear in the response message. Some fields are always present, some will be present if they were in the original message, and others will be present based on other criteria including the original message transaction type. The Comments column indicates in what circumstances the fields appear in the message.

Field Name	Type	Description	Comments
pg_merchant_id	N8	The merchant's six-digit ID code	Always present
pg_transaction_type	L	Indicates the Transaction type	Always present
pg_merchant_data_[1-10]	A255	Up to ten merchant-defined fields	Echoed if specified
pg_total_amount	M	Amount to be charged/credited to the customer	Echoed if specified
pg_sales_tax_amount	M	Sales tax amount.	Echoed if specified
pg_customer_token	A50	Unique string ID that references a customer's stored information	Echoed if specified
pg_client_id	N50	Unique integer ID that references a customer's stored information. Returned with response.	Echoed if specified
pg_consumer_id	A15	ID assigned by the merchant and returned with the response	Echoed if specified
ecom_consumerorderid	A36	ID assigned by the merchant and returned with the response	Echoed if specified
pg_payment_token	A50	Unique string ID that references a customer's stored payment information. Returned with response.	Echoed if specified
pg_payment_method_id	N50	Unique integer ID that references a customer's stored payment information. Returned with response.	Echoed if specified
ecom_walletid	A15	ID assigned by the merchant and returned with the response	Echoed if specified
ecom_billto_postal_name_first	A25	Customer's first name	Echoed if specified
ecom_billto_postal_name_last	A25	Customer's last name	Echoed if specified
pg_billto_postal_name_company	A20	Company name	Echoed if specified
ecom_billto_online_email	A40	Customer's email address	Echoed if specified
pg_response_type	L	Single letter response that indicates the success or failure of a transaction. Use the following values: • A = Approved • D = Declined • E = Error	Always present
pg_response_code	A3	Three-character code representing the transaction result (see the tables in Appendix A)	Always present

Continued

Understanding the Response Message Template (Cont'd)

Response Message Template

Field Name	Type	Description	Comments
pg_response_description	A80	Text description of the transaction results	Always present
pg_avs_result	N5	Five digits representing the outcome of the requested AVS checks (see Appendix C for detailed information)	Present if AVS method specified
pg_avs_codes	L	Single letter response that indicates the outcome of the requested AVS checks (see Appendix C for detailed information). Use the following values: • X = Match: Street Address and 9-digit Zip Code both match • Y = Match: Street Address and 5-digit Zip Code both match • A = Partial Match: Street Address matches, but both 5-digit and 9-digit Zip Code do not match • W = Partial Match: Street Address does not match, but 9-digit Zip Code matches • Z = Partial Match: Street Address does not match, but 5-digit Zip Code matches • N = No Match: Street Address, 5-digit Zip Code, and 9-digit Zip Code all do not match • U = System Unavailable: Address information unavailable. Forte returns this response if the Street Address is a non-US address, if the AVS service is unavailable, or if the AVS service for a particular US bank is not properly functioning. • R = System Unavailable: Forte will retry the AVS check because the issuer's system is unavailable or the request times out. • E = Invalid: AVS data is invalid • S = Not Supported: The US issuing bank does not support AVS checks.	Always present
pg_trace_number	A36	36-character code uniquely identifying the transaction	Always present
pg_authorization_code	A8	8-character approval code from the vendor providing credit card or EFT authorization	Present if authorization performed

Continued

Understanding the Response Message Template (Cont'd)

Response Message Template

Field Name	Type	Description	Comments
pg_preauth_result	L	Pre-authorization check result (i.e., Forte Verify) using the following values: • POS = Positive • NEG = Negative • UNK = No information available	Present if pre-authorization performed
pg_preauth_description	A80	Text description of the pre-authorization result	Present if pre-authorization performed
pg_preauth_neg_report	A80	Negative database response information (unformatted)	Generally, provides details on the negative pre-authorization decline and contact information for consumer inquiries when it is available
pr_requested_amount	M	The originally requested amount for partially authorized transactions.	The originally requested amount for partially authorized transactions
pg_available_card_balance	M	Field present if partial authorization or balance inquiry was performed and balance was returned by the authorizing vendor.	Present if partial authorization or balance inquiry was performed and balance was returned by the authorizing vendor
pg_cvv2_result	A1	This field is present in Global and Vital responses for credit card transaction with CVV2 information. Single character result code (e.g., M for match or N for no match). Other responses are possible but may be ignored. **NOTE**: Transactions declining or being approved on the CVV2 code is at the sole discretion of the card issuer.	Present for credit card transactions with CVV2 information sent to GlobalPayments and Vital
pg_cvv_code	L	Single-letter response that indicates the outcome of a CVV verification for credit card transactions. Use the following values: • M = Match • N = No Match • E = Error (Unrecognized or Unknown Response) • I = Invalid or Null • P = Not Processed • S = Service Not Supported • U = Issuer Unable to Process • X = No Response	Always present for credit card transactions

Working with Convenience Fees

Overview

Merchant accounts approved to process convenience fee transactions should include additional fields in their transaction requests. Consequently, an additional field is returned in the transaction response.

Convenience Fee Request Fields

The following table displays the existing field (shaded) and the new field merchants can use to capture convenience fees:

Group	Field Name	Description	Type	Req
Convenience Fee	pg_total_amount	The total amount, including the transaction fee to be charged	M	M
	pg_convenience_fee	The amount of the convenience fee	M	M

Forte uses these fields to calculate the original amount (total amount – convenience fee = original amount) and check it against the convenience fee information in the database for the merchant account specified in the pg_merchant_id field. If the convenience fee is incorrect, Forte declines the transaction with the following response:

```
pg_response_code=U28
pre_response_description=CONV FEE INCORRECT
```

Convenience Fee Response Fields

The following field displays in the transaction response after the merchant has specified a convenience fee in the request:

Group	Field Name	Description	Type	Req
Convenience Fee	pg_convenience_fee	The amount of the convenience fee	M	M

Adding Line Items

Overview

Merchants can include up to 100 line items by passing the line item fields in their transaction message(s).

Line Item Fields

The following table displays the fields for line items in a transaction message. Header and item lines allow for a maximum of 8000 characters.

Group	Field Name	Description	Type	Req
Line Items	`pg_line_item_header`	Description of the data elements contained within each line item. This header will be displayed when viewing transaction details within the Virtual Terminal.	A256	O
	`pg_line_item_[1-100]`	Contents of the line item formatted according to `pg_line_item_header`.	M	M

Line Item Syntax

The example below displays the syntax for line items in a transaction message.

```
pg_line_item_header=col1,col2,col3
pg_line_item_1=value1,value2,value3
pg_line_item_2=value1,value2,value3
pg_line_item_3=value1,value2,value3
```

The example below displays how this syntax looks with data:

```
pg_line_item_header=SKU,Price,Qty
pg_line_item_1=021000021,45.00,2
pg_line_item_2=021000022,36.99,10
pg_line_item_3=021000023,27.50,7
```

Using Transaction Type Codes in Messages

Overview

Every transaction that Forte processes must be assigned a transaction type via the `pg_transaction_type` field. This chapter lists these fields.

Transaction Type Codes

The following table displays the fields for line items in a transaction message. Header and item lines allow for a maximum of 8000 characters.

Group	Type	Name	Description	Comments
Credit Card	10	SALE	Customer's card is charged and will be automatically settled at the end of the day.	Customer is charged
	11	AUTH ONLY	Customer's card is charged but will not be settled until a CAPTURE message is completed.	Authorization only, CAPTURE transaction required
	12	CAPTURE	Completes an AUTH ONLY transaction. The original charge will be settled at the end of the day.	Completes AUTH ONLY transaction
	13	CREDIT	Customer's card is credited and will be automatically settled at the end of the day.	Customer is credited
	14	VOID	If the target transaction has not be settled, it will be voided (and never be settled). Attempts to void a settled transaction will be declined (with an appropriate response code).	Cancels non-settled transactions
	15	PRE-AUTH	The customer's card is charged using a merchant-supplied authorization code (received from the card processor directly). This is sometimes referred to as a FORCE transaction.	Customer charge approved from other source
	18	BALANCE INQUIRY	For merchant accounts approved to process partial authorization transactions, this field requests the available balance from a card. For this transaction type, do not include the `pg_total_amount` field. Supported authorizing vendors: GlobalPayments and FirstData.	Requests the available balance from a card
EFT	20	SALE	Transaction is completed and the funds will be captured at the end of the day.	Customer is charged
	21	AUTH ONLY	Transaction is authorized, but the funds are not captured until a CAPTURE message is completed.	Authorization only, CAPTURE transaction required
	22	CAPTURE	Completes an AUTH ONLY transaction. The funds for the original transaction will be captured at the end of the day.	Completes AUTH ONLY transaction
	23	CREDIT	Transaction is completed and the funds will be transferred at the end of the day.	Customer is credited
	24	VOID	If the target transaction has not been settled, it will be voided (and will never be settled). Attempts to void a settled transaction will be declined (with an appropriate response code).	Cancels non-settled transactions

Continued

Using Transaction Type Codes in Messages (Cont'd)

Transaction Type Codes

Group	Type	Name	Description	Comments
EFT	25	FORCE	Transaction is completed and the funds will be captured at the end of the day. Verification checks are skipped for this type of transaction. You must contact Forte Customer Service to put an EFT FORCE into place.	Customer charged (no validation checks)
	26	VERIFY ONLY	Transaction is verified but no authorization is obtained and it cannot be settled. This is for use with Forte Verify. For these transactions, the "Customer/Order Information" fields are all optional except for amount.	Verification only, no customer charge
Recurring	40	SUSPEND	The (active) recurring transaction is put into a suspended state. No more transactions will be generated on its behalf until it is reactivated.	Suspends a recurring transaction
	41	ACTIVATE	The (suspended) recurring transaction is returned to an active state. Transactions will again be generated on its behalf.	Reactivates a recurring transaction
	42	DELETE	The recurring transaction will be deleted permanently.	Deletes a recurring transaction

Setting Up Credit Card Messages

Overview

The following chapter provides notes and helpful hints for setting up credit card messages.

Using Templates

When setting up messages for SALE, AUTH ONLY, CREDIT, and PRE-AUTH transaction messages (types 10, 11, 13, and 15, respectively), use the fields of the Credit Card Transaction group of the Transaction Message Template.

When creating CAPTURE and VOID transaction messages (types 12 and 14), use the fields of the Administrative Message group of the Transaction Message Template.

Understanding Fields

The AVS field can be used with SALE and AUTH ONLY transactions (because credits do not require preauthorization, and PRE-AUTH transactions have already been authorized). The miscellaneous "PREAUTH" fields are not used in credit card messages.

Recurring fields can be used only for SALE and CREDIT messages.

Procurement card transactions must have the pg_procurement_card field set to TRUE and require the sales tax and customer account codes.

Magstripe data (track one or two) may be included in the swipe data field. Mail order and phone order transaction must include the pg_mail_phone_order field set to TRUE.

Understanding Settlement

SALE, CREDIT, and FORCE transactions settle at the end of the day. AUTH ONLY transactions settle at the end of the day that their corresponding CAPTURE message is approved.

Only unsettled transactions may be voided. Voided transactions are never settled. CAPTURE and VOID messages require the original authorization code in addition to the original trace number.

Continued

Setting Up Credit Card Messages (Cont'd)

Understanding Credit Card Transaction Qualification

"Downgrading" can occur when a portion of information is missing from the credit card authorizing request. This may result in a higher fee for the offending transaction. Downgrading typically (although not always) occurs when a card has to be manually keyed into the system and some information required by the credit provider is omitted.

To qualify as swipe transactions, retail transactions typically require information in the swipe data fields. Contact your credit card provider for specific rules. Each credit provider has different rules about what data elements constitute a fully-qualified transaction.

Forte is not responsible for transaction downgrading. The merchant is responsible for contacting his/her credit providers, learning which data elements are required, and ensuring his/her messages contain that data.

Verifying Your Transactions

Forte recommends that immediately after go-live, you contact your credit providers to ensure that the information you are sending them meets their standards for a fully-qualified transaction. Contacting them early can help you avoid needless expense. Some customers have waited until receiving a statement from their provider, only to discover that they were neglecting to enter a key piece of data, resulting in thousands of downgraded transactions.

The following tips may help you prevent your credit card transactions from being downgraded:

1. Non-swipe transaction should include street address and zip code information so an AVS check can be performed. Forte also recommends keying in the CVV/CVV2/CIV/CID data.
2. Include invoice, ticket, or P.O. numbers in the `ecom_consumerorderid` field.
3. When keying in purchase and/or corporate cards, use the `pg_sales_tax_amount` and `pg_customer_acct_code` fields.

Setting Up EFT Messages

Overview	The following section provides notes and helpful hints for setting up electronic funds transfer (EFT) messages.
Using Templates	The SALE, AUTH ONLY, CREDIT, and FORCE messages (types 20, 21, 23, and 25, respectively) use the fields of the EFT Transactions group of the Transaction Message Template. The CAPTURE and VOID messages (types 22 and 24) use the fields of the Administrative Message group of the Transaction Message Template.
Understanding Fields	The AVS field can be used with SALE and AUTH ONLY transactions. Only SALE, CREDIT, and FORCE messages can include the recurring fields. The check number field is only used for point-of-sale transactions.
Understanding Settlement	SALE, CREDIT, and FORCE transactions settle at the end of the day. AUTH ONLY transactions settle at the end of the day that their corresponding CAPTURE message is approved. Only unsettled transactions may be voided. Voided transactions are never settled. CAPTURE and VOID messages require the original authorization code in addition to the original trace number.

Using Recurring Transaction Admin Messages

Overview

You use recurring transaction admin messages to activate, suspend, and delete existing recurring transactions. These message use the Admin Message Template of the Transaction Message Template and differ only in their transaction type. Deleted recurring transactions cannot be activated or suspended. The trace number returned by the original transaction is required by these messages.

Setting up and using recurring transactions can be confusing to some end users because recurring payments can be set up in different ways. The following sections detail common scenarios using recurring payments.

Reviewing the Basics

All recurring transactions are based on the original transaction. At a minimum, recurring transactions must include the following fields:

Field	Description
pg_total_amount	Amount to be charged on the date of the sale
pg_schedule_quantity	Number of payments including the original payment (e.g., if this value is 12, then 11 more payments will be made after the initial payment on the date of the sale). **NOTE:** This value must equal the total number of payments.
pg_schedule_frequency	How often the payment will recur (see the Transaction Message Template for a list of valid codes)

Recurring transactions with the three fields listed above have the simplest setup:

- The original date of the sale is the same day of the month any future payments will be made.
- All payments will be the same amount, which is the amount charged on the day of the sale.

For example, if the recurring transaction included the following information,

```
pg_total_amount=100.00
pg_schedule_frequency=20
pg_schedule_quantity=12
```

then the transaction would include a total of 12 payments (including the initial payment)starting monthly from date of the sale and

1 x $100 = $100 Equals the initial transaction on the submission date
11 x $100 = $1100 Equals the remaining payments in the recurring schedule

with a Grand Total = $1200.00 between a total of 12 payments.

Continued

Using Recurring Transaction Admin Messages (Cont'd)

Varying the Payment Amount

To create transactions where the amount on the original sale date is different from the amount of subsequent payments, use the `pg_total_amount` field to specify the amount to be processed on the date of sale and the `pg_schedule_recurring_amount` field to specify the amount of recurring payments.

For example, given

```
pg_total_amount=53.20
pg_schedule_quantity=12
pg_schedule_frequency=20
pg_schedule_recurring_amount=100.00
```

The first payment is $53.20. In subsequent months, on the same day of the month as the first payment, $100.00 will be deducted from the customer's account until a total of 13 payments have been deducted (including the first payment). With these 13 payments, the total amount paid equals:

12 x $100 = $1200.00
1 x $53.20 = $53.20
$1200.00 + $53.20 = **$1253.20**

Varying the Date of Payments

Most customers want their payments to fall on a specific day of the month. Use the `pg_schedule_start_date` field to accomplish this.

To create transaction where the date of the recurring payments is different from that of the original sale date, use the `pg_schedule_start_date` field to specify the date when recurring payments should begin. If the start date is on or before the day the initial transaction is processed, the next start date will be the following day.

For example, given

```
pg_total_amount=53.20
pg_schedule_quantity=12
pg_schedule_frequency=20
pg_schedule_recurring_amount=100.00
```

The second and subsequent payments are deducted on the first day of every month. With these 13 payments, the total amount paid equals:

12 x $100 = $1200.00	Deducted on the first of every month
1 x $53.20 = $53.20	Taken on the date of sale

$1200.00 + $53.20 = **$1253.20**

Continued

Using Recurring Transaction Admin Messages (Cont'd)

Creating a Recurring Transaction without a Payment on the Sale Date

You can create recurring transactions without a payment on the date of sale through one of three methods that Forte recommends:

Method 1: Do not submit the transaction until you want the payments to begin. This manual solution requires that you hold transaction records and submit them on the exact date to be used for all future payments. You would submit the payment on that date, which would then be used for all future subsequent payments.

For example, if a customer wanted to purchase a new chair for $1000 and wants to pay for it over a 10-month period beginning on the first of the next month, you would need to hold the sales slip until the first of the month and then submit the transaction.

Method 2: Submit a small payment on the day of the transaction (e.g., a dollar or more) and then set up the remainder of the payments at the corrected amount.

For example, to set up the purchase of a $1000 chair over 10 months beginning June 1, 2006, use the following code:

```
pg_total_amount=1.00
pg_schedule_quantity=10
pg_schedule_frequency=20
pg_schedule_recurring_amount=99.90
pg_schedule_start_date=6/1/2006
```

In this example, the customer pays $1.00 on the purchase date and pays $99.90 per month every month for the next 10 months beginning on June 1, 2006.

This method can also be used even if the purchase amount is not evenly divisible by the number of payments. For example, if the chair including sales tax cost $1003.38, the merchant could charge $3.38 to the customer's card on the day of the sale and then set up the remaining 10 payments of $100 each, as shown below:

```
pg_total_amount=3.38
pg_schedule_quantity=10
pg_schedule_frequency=20
pg_schedule_recurring_amount=100.00
pg_schedule_start_date=6/1/2006
```

Continued

Using Recurring Transaction Admin Messages (Cont'd)

Creating a Recurring Transaction without a Payment on the Sale Date

Method 3: Set up and run the recurring transaction, then void the original transaction, resulting in recurring payments but no payment on the date of sale. To use this method, you would set up the transaction as described in method 2 above. After obtaining confirmation that the transaction is accepted, you must immediately void the transaction. The first payment (e.g., the $1.00 payment) will be authorized but voided, resulting in no actual charge to the customer's bank account. The recurring transaction information has been submitted and authorized and future payment will occur as scheduled.

NOTES:

1. Any time that `pg_schedule_recurring_amount` is present, the total number of payment will be `pg_schedule_quantity` + 1. This takes into account that no transactions have been voided.
2. A message containing a basic set of recurring fields will yield `pg_schedule_quantity` payments. This takes into account that no transactions have been voided.

Understanding Response Messages

Overview

A transaction message can result in one of three possible outcomes: approved, declined, or error. The three response fields (type, code, and description) give different levels of detail in all cases.

Appendix A lists all the possible response codes for a transaction message, which can be classified into three groups: results, formatting errors, and exceptions.

The **results** group (codes beginning with "U" or "A") represent successfully processed transactions (approved or declined). The **formatting** errors group (codes beginning with "F") represent messages not processed because of one or more errors in the message formatting. The response description field will list the offending fields and the original message is archived to assist in technical support. The **exceptions** group (codes beginning with "E") represent codes for messages encountering some fatal condition preventing further processing (e.g., bad merchant ID, security error, or communications timeout).

The `preauth` fields are responses for EFT transactions utilizing the Forte Verify product. The `pg_preauth_result` field indicates the status of the account in question (see Appendix B for more information on Forte Verify processing).

Testing

Overview

The following section details what information you need to set up, test, certify, and bring your system up live with messages. When testing on the Forte platform, keep in mind that

- Our servers are available 24/7, every day
- You can build and test at your own pace
- We provide examples of code
- We provide preset responses for some message so you will know what response indicates a successful test

After completing testing, your messages will be ready to be placed in a live production environment and your system is ready for operation.

Preparing for Testing

Once you've composed your messages, you will need a system sign-on, which consists of a Merchant ID and processing password that should have been included in your new merchant approval letter/package.

URLs

The following table displays the test URLs for the RAW HTTP POST and SOAP POST methods:

Delivery Method	URL
HTTP POST	https://www.paymentsgateway.net/cgi-bin/posttest.pl
SOAP POST	https://ws.paymentsgateway.net/pgtest/paymentsgateway.asmx

Differences Between Test and Live Servers

The test and live servers are virtually the same except for the following differences:

- 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
- Test Forte Verify transactions are run against an internal StarChek test bed

Going Live

Overview

"Going Live" occurs when your system is ready to work in a production environment. This step requires a minor change to your delivery method and a call to Forte to set up your live account.

Setting Up a Live Account

To set up a live account contact Forte's Technical Support team. The following table shows the parameters for live transactions:

Delivery Method	URL
HTTP POST	https://www.paymentsgateway.net/cgi-bin/postauth.pl
SOAP POST	https://ws.paymentsgateway.net/pg/paymentsgateway.asmx

Best Practices

Overview

The following section summarizes best practices for integrating and maintaining Forte's platform.

Tools Available to Help You

Forte maintains the following online tools to help you through your integration:

Tool	Description	URL/Email
Knowledge Base	Post your questions and our experts will promptly answer them.	https://www.forte.net/developers
Integration Support	Includes support for customers currently undergoing integration or needing assistance with integration or testing issues	integration@forte.net
Developter Site	Includes sample code, a developer forum, and updated documentation	http://www.forte.net/devdocs/

Central Point-of-Contact

Forte recommends you route all communications through a designated central point-of-contact in your organization. This one person will have a full understanding of the business relationship and will keep Forte updated on important issues. Forte must be kept informed about any changes to messages, schedules, or issues that relate to Forte's platform.

Obtaining Help from Forte

When contacting Forte Technical Support, provide the following information to expedite your issue:

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

Managing the Reconciliation Process

The reconciliation process is your responsibility. Failing to properly oversee this process can be costly.

With credit card transactions, you know immediately whether or not the transaction will be paid. However, with ACH and EFT transactions, settlement typically does not occur for minimum of 3–4 days and chargebacks can occur for up to 90 days.

To mitigate the risk inherent in this long settlement time, you must reconcile your settlement information with your authorizations on a regular basis. You can obtain your settlement information from Forte and compare it to your authorizers. If you are pulling down EFT settlement information, a match against transaction results is a good way to ensure accuracy.

You can download settlement files from Forte's web site. For more information see the Batch Transmission File Specification available on the developer's documentation library at https://www.forte.net/devdocs/ or via the Virtual Terminal.

Continued

Best Practices (Cont'd)

Maintaining Documentation

Carefully documenting your integration helps make maintaining your system easier. Forte recommends you document the following:

- **Delivery Method**: Document why you chose that delivery method, who approved the delivery method, and the date you obtained the approval.
- **Messages**: Document the business purpose of each message, any alternate drafts considered, who approved the message, and the date you obtained the approval.
- **Testing**: Document the test methods you use (including any test scripts), who participated in the tests, who approved the test results, and the date you obtained the approval.
- **Certification**: If any changes are made to messages as a result of certification testing, be sure to adjust the documentation. If staff training is delivery 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 occasionally misunderstandings during training can occur).

If a problem occurs in a few months, this documentation will help you determine if the problem is new or was encountered when the system was integrated, tested, or put into production. Additionally, thorough documentation will help you determine whether or not users were ever trained on that topic.

Appendix A: Response Codes

Overview

The following section describes transaction response codes returned in the `pg_response_code` parameter of the response message.

Approved and Declined Responses

The following 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	Test Parameters
A01	APPROVED	Transaction approved/completed	Example transaction messages available in Appendix D
A03	PARTIAL AUTHORIZATION	Transaction approved for a partial authorization (Credit Card only)	Not available
U01	MERCH AUTH REVOKED	Merchant not allowed to access customer account (EFT only)	Not available
U02	ACCOUNT NOT APPROVED	Customer account is in the Forte "Known Bad Accounts" list (EFT only)	Send echeck sale transaction with the following data: • Routing Number: 021000021 • Account Number: 987654321
	TRN NOT APPROVED	Routing number passes checksum test but is not valid for ACH	Send echeck sale transaction with the following data: • Routing Number: 064000101 • Account Number: Any account number
U03	DAILY TRANS LIMIT	Merchant daily limit exceeded (EFT only)	Not available
U04	MONTHLY TRANS LIMIT	Merchant monthly limit exceeded (EFT only)	Not available
UO5	AVS FAILURE ZIP CODE	AVS State/Zip Code check failed	Set `pg_avs_method = 00200` and send a state and zip code that do not match
U06	AVS FAILURE AREA CODE	AVS State/Area Code check failed	Set `pg_avs_method = 00020` and send a state and area code that do not match
U07	AVS FAILURE EMAIL	AVS anonymous email check failed	Set `pg_avs_method = 00002` and send an email address for Hotmail.com
U10	DUPLICATE TRANSACTION	Transaction has the same attributes as another transaction within the time set by the merchant	Send the same transaction twice within five minutes

Continued

Appendix A: Response Codes (Cont'd)

Approved and Declined Responses

Code	Description	Comments	Test Parameters
U11	RECUR TRANS NOT FOUND	Transaction types 40–42 only	Not available
U12	UPDATE NOT ALLOWED	Original transaction not voidable or capture-able	Send a void transaction for a declined transaction
U13	ORIG TRANS NOT FOUND	Transaction to be voided or captured was not found	Send void transaction for the following trace number: 00000000-0000-0000-0000-000000000000
U14	BAD TYPE FOR ORIG TRANS	Void/Capture and original transaction types do not agree (CC/EFT)	Send a void credit card transaction for an echeck transaction
U15	ALREADY VOIDED ALREADY CAPTURED	Transaction was previously voided or captured	Void the same transaction twice
U18	UPDATE FAILED	Void or Capture failed	Send a transaction for $19.18 or $1918
U19	INVALID TRN	Account ABA number if invalid	Send echeck transaction with TRN of 123456789
U20	INVALID CREDIT CARD NUMBER	Credit card number is invalid	Send a credit card transaction with a card number of 1111111111111111
U21	BAD START DATE	Date is malformed	Send a transaction with scheduling data but a start date of 13/1/2008 or 1/1/2001
U22	SWIPE DATA FAILURE	Swipe data is malformed	Send Credit Card transaction with pg_cc_swipe_data=ABC123
U23	INVALID EXPIRATION DATE	Malformed expiration date	Send Credit Card transaction with Ecom_Payment_Card_ExpDate_Month=13
U25	INVALID AMOUNT	Negative amount	Send a transaction for a negative amount (-$1.00)
U26	INVALID DATA**	Invalid data present in transaction	Send a void transaction with pg_original_authorization_code=.
U27	CONV FEE NOT ALLOWED	Merchant configured for convenience fee but did not send one	For merchants configured to accept a convenience fee, send a transaction with an incorrect convenience fee in pg_convenience_fee
U28	CONV FEE INCORRECT	Merchant configured for convenience fee but did not send one	For merchants configured to accept a convenience fee, send a transaction with an incorrect convenience fee in pg_convenience_fee

Continued

Appendix A: Response Codes (Cont'd)

Approved and Declined Responses

Code	Description	Comments	Test Parameters
U29	CONV FEE DECLINED	Convenience fee transaction failed – SplitCharge model only	N/A
U30	PRINCIPAL DECLINED	Principal transaction failed – SplitCharge model only	N/A
U51	MERCHANT STATUS	Merchant is not "live"	Send a transaction for a non-live Merchant ID
U52	TYPE NOT ALLOWED	Merchant not approved for transaction type (Credit Card or EFT)	Send a transaction of a type (Credit card or echeck) that the account is not allowed to process
U53	PER TRANS LIMIT	Transaction amount exceeds merchant's per transaction limit (EFTs only)	Send a transaction that exceeds the merchants echeck limit(s)
U54	INVALID MERCHANT CONFIG	Merchant's configuration requires updating – call Customer Support	Send a transaction for $19.54 or $1954
U80	PREAUTH DECLINE	Transaction was declined due to preauthorization (Forte Verify) result	Send a transaction for $19.80 or $1980
U81	PREAUTH TIMEOUT	Preauthorizer not responding (Verify Only transactions only)	Send a transaction for $19.81 or $1981
U82	PREAUTH ERROR	Preauthorizer error (Verify Only transactions only)	Send a transaction for $19.82 or $1982
U83	AUTH DECLINE*	Transaction was declined due to authorizer declination	Send a transaction for $19.83, $1983, or $1.33
U84	AUTH TIMEOUT	Authorizer not responding	Send a transaction for $19.84 or $1984
U85	AUTH ERROR	Authorizer error	Send a transaction for $19.85 or $1985
U86	AVS FAILURE AUTH	Authorizer AVS check failed	Send a transaction for $19.86 or $1986
U87	AUTH BUSY	Authorizing vendor busy, may be resubmitted (credit card only)	Send a transaction for $19.87 or $1987
U88	PREAUTH BUSY	Verification vendor busy, may be resubmitted (type 26 only)	Send a transaction for $19.88 or $1988
U89	AUTH UNAVAIL	Vendor service unavailable (credit card only)	Send a transaction for $19.89 or $1989
U90	PREAUTH UNAVAIL	Verification service unavailable (type 26 only)	Send a transaction for $19.90 or $1990

*`pg_response_description` will contain the text of the vendor's response.
**`pg_response_description` will contain a more specific message.

Continued

Appendix A: Response Codes (Cont'd)

Formatting Error Responses

The following table displays the codes returned when Forte finds formatting errors. The response description field will actually list all the 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 an "F."

Code	Description	Comments
F01	MANDATORY 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

Fatal Error Responses

The following table displays exceptions that 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 processing password is incorrect
E20	MERCHANT TIMEOUT	Transaction message not received (I/O flush required?)
E55	INVALID TOKEN	Specified token was invalid, could not be located, or may have been deleted
	Client Token Transactions	For client token transactions where neither payment fields nor a payment token were specified, the client record does not have a default payment method matching the transaction type
	Payment Token Transactions	For payment token transaction where no client token is specified, the payment token must be clientless
	Both Client and Payment Tokens Present	For transactions with client and payment tokens, the specified payment token is not associated with the client or is clientless
E90	BAD MERCH IP ADDR	Origination IP not on merchant's approved IP list
E99	INTERNAL ERROR	An unspecified error has occurred

Appendix B: Forte Verify

Overview	Forte Verify is an optional service that provides additional verification of an EFT account number. Forte performs these "preauthorization" searches (also called "checks") automatically for subscribing merchants (the authorization messages require no additional fields).
Verifying Accounts	The Forte Verify service consults the status reported by the bank to see if the customer account is valid and in good standing. The response indicates if the account is open and valid, closed, NSF, or one of the other conditions listed in the table below. Charges are only assessed for transactions involving participating banks (Most Tier I and II banks participate, but some local banks and smaller credit unions may not). Transactions that do not receive a definitive response (such as `POS` or `NEG`) may then be checked against the national negative check database. Note that the status of the account may change between the bank's report and settlement.
Using Forte Verify	Forte Verify searches yield up to four additional response fields (see below). The most important field is `pg_preauth_result`, which indicates the result of the verification. `POS` indicates a positive response from the verification service and `NEG` indicates a negative response. `UNK` indicates that nothing more is known about the account (for various reasons). Use the following response fields when performing Forte Verify searches: • `pg_preauth_result`: The value in this field may cause a transaction to be declined, depending upon the setup. Potential values for this field include the following: `POS`, `NEG`, or `UNK`. • `pg_preauth_description` - This field displays the current state of the account as provided by the verifying agent. • `pg_preauth_neg_report` - This field indicates negative database responses and normally contains the negative report details and (usually) the name and phone number of the reporting entity.

Continued

Appendix B: Forte Verify (Cont'd)

Response Values

Forte returns the following values in the result and description fields listed above. Sandbox normally generates a `POS` result for any account (no participating bank check is performed). The test account numbers below may be used on Sandbox (with any valid ABA number) to force the indicated response.

Result	Description	Test Account #
NEG	P15:HIGH RISK	99915
NEG	P41:NEGATIVE INFO	99941
UNK	P50:NO INFO	99950
POS	P70:VALIDATED	99970
POS	P71:LOW RISK APPROVAL	99971
POS	P73:MEDIUM RISK APPROVAL	99973
UNK	P80:PREAUTH VENDOR BUSY	99980
UNK	P90:PREAUTH VENDOR UNAVAIL	99990
UNK	P91:PREAUTH VENDOR ERROR	99991
UNK	P92:PREAUTH SERVER UNAVAIL	99992

Approval and Forte Verify

Forte Verify applies only to Sales, Auth Only, and Verify Only transactions (Types `20`, `21`, and `26`). Transactions with a `NEG` result are normally declined for that reason. Those with `UNK` and `POS` responses are not declined and may be subject to other checks. Normally, `UNK` and `POS` responses are approved. If the merchant only uses Verify Only transactions, he or she may want to use the `pg_preauth_result` value instead of the `pg_response_type` value for his/her decision making.

Authorization and Forte Verify

Within the Forte platform, merchants can set up multiple levels of account verification for ACH items. If the final response is positive, `UNK`, or the account is not found, the transaction is APPROVED in all cases. If the final response has negative reports, the transaction is DECLINED.

Appendix C: AVS and Other Verification Systems

Overview

Credit card companies typically use Address Verification Systems (AVS) by matching the street address number and the zip code with that of the cardholder. Forte offers more verification checks, including the following:

- State/Zip Code match
- State/Area Code match
- Anonymous Email check

The `pg_avs_method` field specifies what checks to perform and whether or not to decline a transaction if they fail. The `pg_avs_results` field indicates which checks passed, failed, or were not performed. Both fields consists of five digits, each representing one of the checks mentioned above. The table below lists the checks represented by each digit position.

$X_1\ X_2\ X_3\ X_4\ X_5$ is the value specified by one of the AVS fields.

Digit	Description
X_1	Credit Card Account/Zip Code Check
X_2	Credit Card Account/Street Number Check
X_3	State/Zip Code Check
X_4	State/Area Code Check
X_5	Anonymous Email Check

The digits represent different values depending on the field.

Digit	`pg_avs_method`	Digit	`pg_avs_result`
0	Do not perform check	0	Check not performed
1	Check only; do not decline on fail	3	Passed
2	Check and decline on fail	4	Failed

Credit Card Account Checks (Positions 1 and 2)

Forte only performs Credit Card Account checks for credit card transactions and ignores EFT transactions. The merchant's assigned authorizing agent (e.g., Nova) performs the checks. The authorizer authorizes the transaction if the AVS checks fail and there is no other reason to decline. If the `pg_avs_method` position representing a failed check is set to `2`, the processor declines the transaction and it will not settle.

NOTE: Forte recommends that merchants at least have these checks performed (positions 1 and 2 set to 1) as they will usually get a better rate from the authorizer.

Continued

Appendix C: AVS and Other Verification Systems (Cont'd)

State/Zip Code and State/Area Code Checks (Positions 3 and 4)

The State/Zip Code and State/Area Code checks compare the customer's billing address state (official two-character abbreviation) and zip code or area code.

Anonymous Email Check (Position 5)

The Anonymous Email check compares the customer's specified email address against a list of known anonymous email services (e.g., Hotmail). The following table displays examples of Anonymous Email checks.

Method	Results	Declined?	Description
22000	34000	Yes	Declined because the Credit Card Street Number check failed
11000	34000	No	The Credit Card Street Number check failed, but the action was a "Check only"
22001	33004	No	The Anonymous Email check failed, but the action was a "Check only"
00222	00334	Yes	The Anonymous Email check failed and was a "Decline on fail" action

NOTE: If the required data for a requested AVS check (value of "1" or "2") is missing, Forte sends an `F01` (mandatory field missing) response code along with the missing field(s).

Implicit AVS Checks

Forte still performs an AVS check even if the merchant specifies the zip code and/or street address fields for a credit card transaction without specifying the `pg_avs_method` field. Forte performs this implicit AVS check silently without returning the `pg_avs_result` field.

Appendix D: Example Messages

Overview

The following sections illustrate various message types. Each message is displayed as newline delimited and the merchant ID and password values have been omitted. A newline character is present after the `endofdata` tag in each of the messaging examples.

Credit Card Sale Transaction

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=10
pg_total_amount=1.13
ecom_billto_postal_name_first=John
ecom_billto_postal_name_last=Smith
ecom_billto_postal_street_line1=123 Any Street
ecom_billto_postal_city=San Diego
ecom_billto_postal_stateprov=CA
ecom_billto_postal_postalcode=92104
ecom_payment_card_name=John Smith
ecom_payment_card_type=Visa
ecom_payment_card_number=<Credit Card Account Number>
ecom_payment_card_expdate_month=<Credit Card Expiration Month>
ecom_payment_card_expdate_year=<Credit Card Expiration Year>
pg_avs_method=22000
endofdata
```

Credit Card Capture Transaction

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=12
pg_original_authorization_code=42344
pg_original_trace_number=c3dce169-ad27-11d5-a75c-0050da8def0f
endofdata
```

Credit Card Authorize Only Transaction

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=11
pg_total_amount=1.00
ecom_billto_postal_name_first=John
ecom_billto_postal_name_last=Smith
ecom_billto_postal_street_line1=123 Any Street
ecom_billto_postal_city=San Diego
ecom_billto_postal_stateprov=CA
ecom_billto_postal_postalcode=92104
ecom_payment_card_name=John Smith
ecom_payment_card_type=Visa
ecom_payment_card_number=<Credit Card Account Number>
ecom_payment_card_expdate_month=<Credit Card Expiration Month>
ecom_payment_card_expdate_year=<Credit Card Expiration Year>
pg_avs_method=22000
endofdata
```

Continued

Appendix D: Example Messages (Cont'd)

Credit Card Force Transaction

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=15
pg_total_amount=1.13
ecom_billto_postal_name_first=John
ecom_billto_postal_name_last=Smith
ecom_billto_postal_street_line1=123 Any Street
ecom_billto_postal_city=San Diego
ecom_billto_postal_stateprov=CA
ecom_billto_postal_postalcode=92104
ecom_payment_card_name=John Smith
ecom_payment_card_type=Visa
ecom_payment_card_number=<Credit Card Account Number>
ecom_payment_card_expdate_month=<Credit Card Expiration Month>
ecom_payment_card_expdate_year=<Credit Card Expiration Year>
pg_original_authorization_code=42344
endofdata
```

Credit Card Recurring Sale (Initial Charge + 11 Equal Monthly Charges)

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=10
pg_total_amount=9.95
ecom_billto_postal_name_first=John
ecom_billto_postal_name_last=Smith
ecom_billto_postal_street_line1=123 Any Street
ecom_billto_postal_city=San Diego
ecom_billto_postal_stateprov=CA
ecom_billto_postal_postalcode=92104
ecom_payment_card_name=John Smith
ecom_payment_card_type=Visa
ecom_payment_card_number=<Credit Card Account Number>
ecom_payment_card_expdate_month=<Credit Card Expiration Month>
ecom_payment_card_expdate_year=<Credit Card Expiration Year>
pg_avs_method=22000
pg_schedule_frequency=20
pg_schedule_quantity=12
endofdata
```

Continued

Appendix D: Example Messages (Cont'd)

EFT Sale Transaction

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=20
pg_total_amount=10.00
ecom_billto_postal_name_first=John
ecom_billto_postal_name_last=Smith
ecom_billto_postal_street_line1=123 Any Street
ecom_billto_postal_city=San Diego
ecom_billto_postal_stateprov=CA
ecom_billto_postal_postalcode=92104
ecom_payment_check_account_type=S
ecom_payment_check_account=<Account Number>
ecom_payment_check_trn=<Routing Number>
pg_avs_method=00220
endofdata
```

EFT Verify Only Transaction

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=26
pg_total_amount=10.00
ecom_payment_check_account_type=S
ecom_payment_check_account=<Account Number>
ecom_payment_check_trn=<Routing Number>
endofdata
```

EFT Void Transaction

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=24
pg_original_authorization_code=42344
pg_original_trace_number=c3dce169-ad27-11d5-a75c-0050da8def0f
endofdata
```

Recurring Admin Delete Transaction

```
pg_merchant_id=<Merchant ID>
pg_password=<Merchant Password>
pg_transaction_type=42
pg_original_trace_number=ef0fe169-ad27-11d5-a75c-c3dc0050da8d
endofdata
```

Continued

Appendix D: Example Messages (Cont'd)

EFT Sale Transaction Response (with Forte Verify)

```
pg_response_type=A
pg_response_code=A01
pg_response_description=APPROVED
pg_merchant_id=<Merchant ID>
pg_authorization_code=420023
pg_trace_number=f0b77189-4014-11d6-a7fd-0050da8def0f
pg_avs_result=0330
pg_transaction_type=20
pg_total_amount=25.95
ecom_billto_postal_name_first=John
ecom_billto_postal_name_last=Smith
pg_preauth_code=none
pg_preauth_result=POS
pg_preauth_description=P70: VALIDATED
endofdata
```

Glossary

ACH	The 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. The types of ACH payments include the following: • Direct deposits of all types including tax refunds, payroll and government benefits (e.g., Social Security) • Direct payments of bills such as utilities, mortgages, loans, and insurance policies • Federal, state, and local tax payments • Business-to-business payments • eChecks • eCommerce payments
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.
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 if 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.
Decline	A transaction which is not approved by the credit provider/issuer. No authorization is issued.

Continued

Glossary (Cont'd)

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."
Merchant ID	This is the identification number for your organization, used by Forte to identify you in all communications. It is critical that anyone contacting Forte for assistance know this ID number.
Pre Auth	This has the same meaning as Auth Only.
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.
Reversal	If a transaction has already settled and should have been voided, it can be reversed by issuing a credit to correct the error.
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.
SIC	Acronym for Standard Industry Classification, this four-digit code is used to classify types of businesses and industries.
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 keythat is transferred over the SSL connection. Web sites that require an SSL connection have an address that begins with *https://* rather than *http://*.

Continued

Glossary (Cont'd)

Travel and Entertainment Card	These credit cards usually require payment in full every month (e.g., American Express, Diner's Club and Carte Blanche).
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.