

IBM WebSphere Commerce Payments for
Multiplatforms



Cassette for VisaNet Supplement

Version 3.1

IBM WebSphere Commerce Payments for
Multiplatforms



Cassette for VisaNet Supplement

Version 3.1

Note

Before using this information and the product it supports, be sure to read the general information under Appendix C, "Notices" on page 71.

Third Edition (July, 2002)

This edition applies to Version 3.1.3 of the Cassette for VisaNet for IBM® WebSphere® Commerce Payments for Multiplatforms and to all subsequent releases and modifications until otherwise indicated in new editions.

©Copyright IBM Corporation 2000, 2002. All rights reserved.

Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	v
Conventions in this book	v
Terminology	v
Additional information	vii
Chapter 1. Overview of VisaNet	1
Internet Commerce with VisaNet	2
VisaNet Merchant Registration	2
IBM WebSphere Commerce Payments Cassette for VisaNet	3
Chapter 2. VisaNet and WebSphere Commerce Payments Concepts	5
Purchasing cards	5
Levels of purchasing card data supported	6
Purchasing card data processing.	6
A VisaNet purchase example	6
WebSphere Commerce Payments object model implementation	8
Administration objects	8
Financial objects.	9
Cassette-specific characteristics and behaviors	9
Understanding retry parameters	9
Cassette for VisaNet payment command summary.	10
Summary of state changes	11
Protecting sensitive data	12
Chapter 3. Installing the Cassette for VisaNet	13
Before installing Cassette for VisaNet	13
Installing Cassette for VisaNet (Windows® and UNIX platforms)	13
WebSphere Commerce Payments directory structure	14
Installing Cassette for VisaNet (iSeries)	14
Installing the cassette	14
Adding a Cassette for VisaNet to a WebSphere Commerce Payments Instance	15
Uninstalling Cassette for VisaNet on Windows NT	15
Uninstalling Cassette for VisaNet on Windows 2000	15
Uninstalling Cassette for VisaNet on Solaris	16
Uninstalling Cassette for VisaNet on AIX	16
Uninstalling Cassette for VisaNet on Linux.	16
Removing Cassette for VisaNet from a WebSphere Commerce Payments Instance for iSeries	17
Uninstalling Cassette for VisaNet for iSeries	17
Chapter 4. Getting Started	19
Cassette for VisaNet tutorial	19
Before starting this tutorial.	20
Starting the WebSphere Commerce Payments User Interface.	20
Configuring the Cassette	20
Configuring the Cassette for SOCKS (iSeries)	21
Selecting a WebSphere Commerce Payments merchant and authorizing a cassette	21
Configuring a merchant.	22
Creating an account	23
Logging in as the merchant administrator	23
Managing payment processing	23

Creating orders using the Sample Checkout	24
Approving orders with the Sale Function	25
Settling batches	27
Issuing a credit	29
Viewing batch totals	29
Separate approvals and deposits	30
Approve	30
Deposit.	30
Chapter 5. Cassette for VisaNet Cashier Profiles	33
Chapter 6. Using WebSphere Commerce Payments commands	35
Cassette for VisaNet Commands	35
AcceptPayment.	35
Approve	37
ApproveReversal	37
BatchOpen	37
BatchPurge	37
BatchClose	37
CassetteControl	38
CloseOrder	38
CreateAccount	38
CreatePaySystem	39
DeleteBatch	39
Deposit.	39
DepositReversal	40
ModifyAccount	40
ModifyCassette	41
ModifyPaySystem	41
ReceivePayment	42
Refund	42
RefundReversal	42
Chapter 7. Using VisaNet objects	43
Financial objects used by Cassette for VisaNet	43
VisaNet Order	43
VisaNet Payment	45
VisaNet Credit	47
VisaNet Batch	48
Administration objects used by Cassette for VisaNet	49
CassetteAdmin	49
AccountAdmin	50
PaySystemAdmin	51
Address Verification Service (AVS) result codes	52
Appendix A. VisaNet return codes	55
Appendix B. Cassette for VisaNet Messages	61
Appendix C. Notices	71
Trademarks	72
Glossary	73
Index	87

Preface

This book is for users and administrators of the Cassette for VisaNet, who are responsible for installation and implementation. This information will help you understand what you need to use the Cassette for VisaNet. Programmers who are responsible for developing applications to manage IBM WebSphere Commerce Payments will find cassette-specific information regarding parameter requirements and cassette-specific states useful. This book serves as a supplement to the *IBM WebSphere Commerce Payments for Multiplatforms Administrator's Guide* as well as the *IBM WebSphere Commerce Payments for Multiplatforms Programmer's Guide and Reference*.

Note:

IBM WebSphere Commerce Payments for Multiplatforms (hereafter called WebSphere Commerce Payments) was previously known as IBM WebSphere Payment Manager for Multiplatforms. Starting with version 3.1.3, the payments application was renamed to WebSphere Commerce Payments and references to the product were changed throughout this document. References to the former product may still appear in this document and apply to earlier releases of the product.

Conventions in this book

Table 1. Conventions used in this book.

Boldface	Indicates the name of the item you need to select, the name of a field, or a string you must enter.
<i>Italics</i>	Indicates book titles or variable information that must be replaced by an actual value.
Monospace	Indicates an example, a portion of a file, or a previously entered value.

Terminology

Enabling electronic shopping requires different participants and software components:

acquirer

An organization that provides card authorization and payment capture services for merchants. A merchant normally wants to accept more than one credit card brand, but does not want to deal with multiple bankcard associations and so uses acquirer services. These services include verbal or electronic telephone authorization support and , electronic payment transfer to the merchant's account. Acquirer services are paid for by the merchant in the form of a small percentage charge on each transaction.

card processor

An agent for an acquirer to whom merchants send their transaction requests. Provides much of the administrative and organizational infrastructure by which merchants process their transactions.

cardholder

A person with a valid payment card account who uses a browser to shop for goods and services on the Internet.

financial network

The aggregate of card processors, acquirers, card issuers and other institutions or organizations, through which payment card transaction processing is traditionally performed.

gateway

A gateway represents an acquirer or card processor acting as a "translator" between the messages used by the merchant and those used on the financial network. (Note that merchants never communicate directly with a gateway.) Once a transaction request is received from a merchant, some type of request is sent to the processor via the gateway. When the gateway receives the message, it translates the request into the format used on the financial network and then forwards that message to the processor. Responses from the processor are translated and sent back in a form that the merchant understands.

issuer The financial institution that provides the cardholder with the credit card. Ultimately, it is the issuer who is responsible for the cardholder debt payment. For example, the issuer balances the risk of a cardholder defaulting against the income from interest payments. The cardholder need not have a relationship with the issuer except for the credit card account, but in practice, most cardholders have at least one credit card from the bank that holds their checking account.

leased line

A phone line leased from the phone company by the customer that connects the customer terminal to a dedicated port on the network.

merchant

A person or organization that has goods or services to sell to the cardholder.

merchant bank

Also known as the Acquiring Financial institution. It acquires merchant business by supplying the merchant with the means to accept credit cards for payment. The financial institution charges the merchant a fee for providing these services.

merchant server

The merchant server displays products for sale to a global market over the Internet. Shoppers can browse through catalogs, adding items to their shopping cart as they go along, and make their purchases when convenient. Merchants can customize their electronic stores, providing discounts for quantity purchases and seasonal merchandise or targeting consumer groups, such as frequent shoppers. They can also track demographic information from data provided by the shoppers and use it for marketing strategies.

payment server

A payment server handles and stores payment information. The payment server accepts payments from the cardholder via the Internet and passes this information along to financial institutions for approval. The payment server also maintains records of all transactions. The IBM WebSphere Commerce Payments is a payment server.

You should also be familiar with terms used in the credit card industry, including:

Authorize

The cardholder is given permission to make a purchase by the financial

institution and the merchant has some guarantee that it will receive funds. It is the validation of the cardholder for a given purchase. The process involves assessing transaction risk, confirming that a given transaction does not raise the account holder's debt above the account credit limit, and reserving the specified amount of credit.

Batch A collection of financial transactions grouped for administrative and record-keeping purposes.

Capture

Funds can be moved or deposited to the merchant's account.

Credit The merchant needs to return money to the cardholder following a valid capture transaction. For example, if goods are returned or are defective, the cardholder receives credit.

Additional information

The WebSphere Commerce Payments product package includes information that describes the cassette-independent functions of WebSphere Commerce Payments:

- The *IBM WebSphere Commerce Payments for Multiplatforms Installation Guide Version 3.1* (WebSphere Commerce Payments Installation Guide) provides information for installing, migrating, and starting WebSphere Commerce Payments.
- The *IBM WebSphere Commerce Payments for Multiplatforms Programmer's Guide and Reference, Version 3.1* (WebSphere Commerce Payments Programmer's Guide) provides details about WebSphere Commerce Payments API.
- The *IBM WebSphere Commerce Payments for Multiplatforms Administrator's Guide Version 3.1* (WebSphere Commerce Payments Administrator's Guide) contains conceptual information and shows how to configure WebSphere Commerce Payments using the user interface.

This book serves as a supplement to the above books, describing cassette-specific programming or administrative considerations and showing how the Cassette for VisaNet implements WebSphere Commerce Payments generic commands.

All documents are available on the WebSphere Commerce Payments CD-ROMs in Portable Document Format (PDF). For the latest Acrobat reader, see: <http://www.adobe.com>. On iSeries™ systems, the documentation is compressed into a save file and is only available after WebSphere Commerce Payments has been installed.

VisaNet and the Vital Processing Services are described in the following documents, all of which are available at <http://www.vitalps.com>:

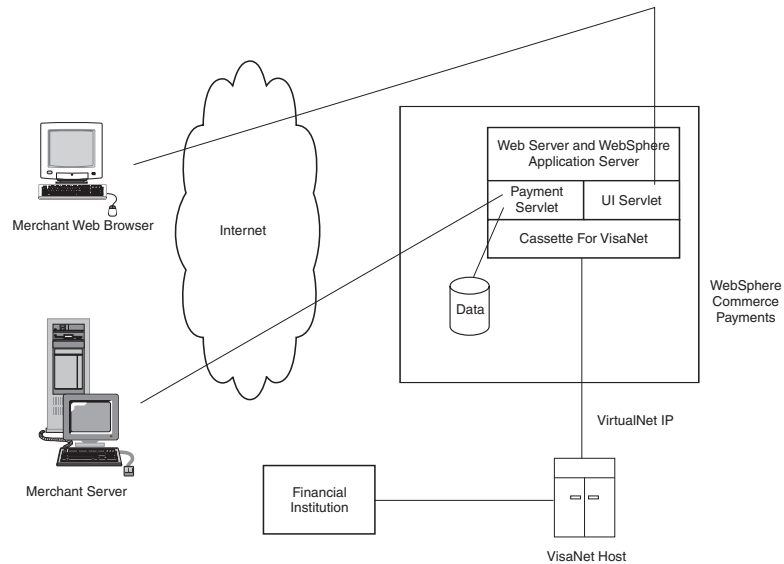
- *EIS 1081 Data Capture Record Formats (version 6.2)*: Contains information on second generation data capture record formats.
- *EIS 1080 Authorization Record Formats (version 6.2)*: Contains information on second generation authorization record formats.
- *EIS 1051 Authorization Protocol (version 3.0)*: Contains information on second generation authorization link level protocol.
- *EIS 1052 Data Capture Protocol (version 3.0)*: Contains information on second generation data capture link level protocol.
- *Transaction Processing Interface Specification (version 9908)*: Vital's connectivity options to VisaNet.

In addition to this manual, you can reference the following related Web sites:

- **<http://www.ibm.com/payment/>** provides more information on the IBM WebSphere Commerce payment processing products.
- **<http://www.software.ibm.com/commerce/payment/support/serv/index.html>** provides current WebSphere Commerce Payments technical information and links to the latest softcopy views of all WebSphere Commerce Payments documentation.
- **<http://www.ibm.com/software/webservers/appserv/library.html>** provides documentation links for IBM WebSphere Application Server.
- **<http://www.ibm.com/software/data/db2/library/>** provides documentation links for IBM Universal Database.

Chapter 1. Overview of VisaNet

The VisaNet system is the largest and most sophisticated consumer financial transaction processing system in the world. It provides worldwide telecommunications and payment data processing, authorizes and settles payments, and offers a range of value-added services such as risk management and fraud control services. Visa Second Generation (Visa II), is the required format for all transactions processed by VisaNet.



Internet Commerce with VisaNet

With the explosion of Internet Commerce, merchants are continually coming online. Brick and mortar merchants use devices like POS terminals to perform real time payment transactions. Internet merchants can not afford to take a step back and perform payment transactions offline. They need the ability to receive and process credit card transactions in real time. Key to their business is the ability to perform on-demand authorization and data capture. In essence, Internet merchants need direct connectivity into VisaNet's high capacity transaction network.

Vital Processing Services is a recognized leader in technology-based commerce enabling services. Vital solutions provide transaction processing capabilities that enable acquirers to strengthen their relationship with both traditional brick and mortar merchants and Web-based merchants operating in the virtual environment of internet commerce. Vital Processing Services provides connectivity to VisaNet through its VirtualNet gateways. VirtualNet offers Internet Service Providers (ISP), Commerce Service Providers (CSP) and merchants two connectivity options:

- IP Gateway
- SSL Gateway

Note

The existing VisaNet cassette only supports VirtualNet IP which requires a leased line connection to Vital Processing Services.

VisaNet Merchant Registration

Vital does not directly perform merchant registration. Merchants must contact a processing bank that uses VisaNet as a processor. Merchants can contact their existing bank or local banks and ask for their merchant services department and then ask if they support VisaNet. The processing bank sets up the account and then works with Vital to complete the registration. If a merchant requires special test accounts, they should work with the processing bank to get that established. The output of the registration is a merchant profile. The merchant profile contains several values for use in processing VisaNet transactions. Those values include the following:

Acquirer BIN

Identifies the merchant's acquiring financial institution (a VisaNet member bank).

Agent Bank Number

Identifies the agent of the acquirer which signed the merchant. This value is provided to the merchant by the acquirer.

Agent Chain Number

Identifies a specific chain of an agent organization. Assigned by the merchant's bank or processor.

Store Number

A number assigned by the signing member, processor, or merchant to identify a specific merchant store within the VisaNet system.

Terminal Number

A number assigned by the signing member, processor, or merchant to identify a unique terminal within a merchant location.

Merchant Name

The merchant name is provided by the signing member or processor.

Merchant Number

Merchant's account number with acquirer (assigned by the merchant's bank or processor).

Merchant Category Code

The merchant industry classification.

American Express CID Support

A merchant does not automatically get Amex CID verification by setting \$CARDVERIFYCODE. In addition to setting this parameter, the acquiring bank must enable CID verification, for the merchant, by contacting American Express merchant services.

IBM WebSphere Commerce Payments Cassette for VisaNet

Payment cassettes are software applications that conform to the data flow and control conventions of the WebSphere Commerce Payments Framework. Each payment cassette contains the implementation of specific payment methods and protocols. The cassette for VisaNet provides merchants with the ability to send real time Internet credit card transactions to the VisaNet system for processing. The cassette enables merchants to utilize the E-Commerce credit card segment of the VisaNet's 6.0 Authorization and Data Capture services. Authorization messages are formatted in accordance with the EIS 1080 Version 6.0 specification. Data Capture messages are formatted in accordance with the EIS 1081 Version 6.0 specification.

All VisaNet transactions are sent to the **VirtualNet IP Gateway**, a VirtualNet Internet Commerce Gateway, for processing. As already noted, the VirtualNet IP Gateway provides connectivity via a private circuit. The private circuit is provided by the Internet Service Provider, Commerce Service Provider or the merchant. The cassette for VisaNet does not provide additional security because transactions occur over a Virtual Private Network directly to VisaNet and not over the Internet.

Address Verification Service

The Address Verification Service (AVS) is a payment card fraud prevention tool that enables mail order and electronic commerce merchants to verify U.S. cardholder shipping addresses against the address that the payment card issuer has on file for the consumer.

The cardholder's billing address, specifically street address and zip code, are sent in the electronic authorization request message to the issuer. The issuer compares the street address and zip code to those it has on file and returns an AVS response code to advise you of the comparison status. This information enables decision making that limits risks when shipping merchandise. Risk reduction for the financial institution can result in reduced transaction fees for the merchant.

The WebSphere Commerce Payments cassette for VisaNet allows the use of this tool by its merchants. It is up to the merchant to decide what risks are allowable if AVS data does not compare favorably.

If you are interested in additional information regarding AVS and merchant chargeback liabilities, contact your acquiring financial institution. See the Vital Processing Services, Authorization Record Formats EIS 1080 Version 6.0 specification, and "Address Verification Service (AVS) result codes" on page 52 for more information on possible AVS codes.

Chapter 2. VisaNet and WebSphere Commerce Payments Concepts

WebSphere Commerce Payments provides a unified interface through which merchants can use multiple payment protocols in a common way. Each WebSphere Commerce Payments cassette attempts to extract protocol-specific differences so that merchants can ignore disparities between protocols.

This section describes how the Cassette for VisaNet presents the VisaNet services through the WebSphere Commerce Payments object model and API set. In addition, cassette-specific behaviors and requirements are discussed.

The Cassette for VisaNet implements the payment commands and the payment processing model of the WebSphere Commerce Payments Framework, using the processing services of VisaNet. This implementation supports:

- AcceptPayment creation of orders only. Wallet-driven purchases are not supported.
- Traditional payment oriented commands.
- Multiple batches for each merchant.
- Multiple batches for an account, one for each currency.

Purchasing cards

Purchasing cards (or "procurement" cards) are credit cards that a business can offer its departments or employees to allow them to buy business related items. Typically a business will make arrangements with the card issuer to govern the purchases that card holders can make. For example, maximum limits can be imposed and the cards can be restricted to allow purchases of certain items only (for example, only stationery goods). Purchasing cards can also have pre-programmed limits for purchase amounts. Purchase-related details (such as the tax amount and merchant category code) and the details of the items being ordered through a purchasing card are passed to the financial network so that the authorization of the purchase can be influenced by the details of the goods being ordered. Purchasing cards are a form of payment commonly used by many businesses because it streamlines the corporate purchasing process.

For example, typically, companies use a purchase order process to receive goods or services. The process usually works like this:

1. An employee requests/creates a purchase order (PO), including a list of goods or services to be purchased.
2. The manager approves the PO.
3. The purchasing department sends the PO to the supplier and files a copy.
4. The receiving department receives the goods and invoice from the supplier.
5. The employee receives goods from receiving department.
6. Accounts Payable receives the invoice from the supplier.
7. Accounts Payable matches the PO to the invoice and then approves the payment.
8. Accounts Payable makes the payment to the supplier.
9. Accounts Payable reconciles the purchase activity.

The use of purchasing cards can remove several of these steps. For example:

1. Employee makes purchase directly from Supplier using Purchasing Card (and logs details of transaction if required by company policy)
2. The employee receives a monthly purchasing card statement, verifies the charges, and send it on to accounts payable.
3. Accounts Payable reviews and approves the statement. The statement may include general purchase information (such as purchase date, amount, commodity code, merchant/supplier info, tax info). It may also include details about line items associated with the purchase (such as the item quantity, unit of measure, part numbers, description, unit cost, tax info). This purchase information is known in the e-commerce industry as Level I, Level II, and Level III information.

The benefits of using purchasing cards include:

- Accounts Payable costs are reduced by not requiring costly EDI implementation.
- Bills are consolidated into a single corporate purchasing card billing statement.
- Check processing is eliminated.
- Suppliers can receive immediate payment and potentially qualify for the low interchange rates by meeting card association requirements for collecting enhanced data.
- Purchases can be paid for electronically and the overall purchase process is more automated.
- Purchasing cards can contain additional authorization controls such as limits for purchase amounts, and where and what types of purchases can be made.

Levels of purchasing card data supported

The Cassette for VisaNet supports two levels of purchasing card information as follows:

Level I data

The standard commercial transaction data for the purchase.

Level II data

Additional data to Level I data about each purchase which includes: sales/local tax and customer reference number.

Purchasing card data processing

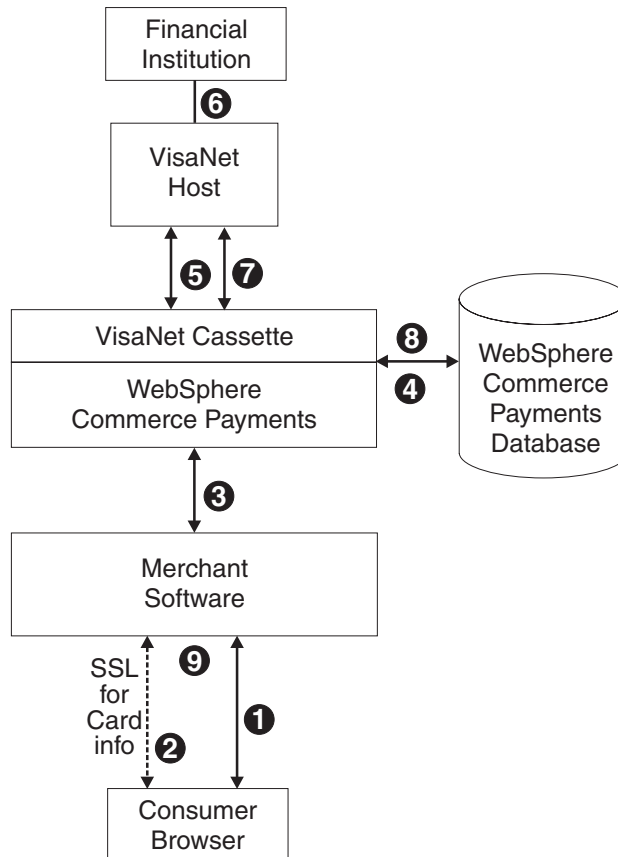
Purchasing card data can be passed in on the AcceptPayment and/or Deposit commands only:

- When purchasing card data is passed in on the AcceptPayment command, the \$NUMPAYMENTS protocol data must either not be specified, or, if specified, must be "1".
- Purchasing card data is then sent to the backend processor during the Batch settlement.
- If purchasing card data is sent on the Deposit command, then it overrides any purchasing card data sent on the AcceptPayment command.

A VisaNet purchase example

The following is an example of how a typical purchase using the Cassette for VisaNet would be processed through the overall system, including the WebSphere Commerce Payments and the Cassette for VisaNet. This example assumes the use of the AutoApprove option of the AcceptPayment command.

Note: Other commands result in different messages being sent to the VisaNet Host, but the same general flow through the overall system still applies.



1. A consumer has been shopping online at a merchant Web site. After choosing several items to purchase, the consumer initiates a purchase, typically by pressing a "Buy" button on the shopping page.
2. The merchant software then requests card information from the consumer over a secure (typically SSL-protected) channel. This information includes the credit card number, the card expiration date, the card brand and possibly the cardholder's address.
3. Once this cardholder information has been received, the merchant software invokes the WebSphere Commerce Payments AcceptPayment command with card information and parameters, requesting that the purchase be approved immediately.
4. The WebSphere Commerce Payments and the Cassette for VisaNet record the information they need to execute payment transactions.
5. The Cassette for VisaNet sends an approve request to the VisaNet Host.
6. The VisaNet Host forwards this request to the financial institution which processes the request and responds to the VisaNet Host.
7. The VisaNet Host records the result, and sends a success response to the cassette.
8. The Cassette for VisaNet, along with the WebSphere Commerce Payments update status in the database and return the success response to the merchant.

9. The merchant software replies to the consumer with an indication that the order is accepted.

WebSphere Commerce Payments object model implementation

This section describes how the Cassette for VisaNet supports the administrative and financial object models that the WebSphere Commerce Payments Framework provides.

Administration objects

The WebSphere Commerce Payments administration objects are the entities that comprise the system and merchant configuration under which all financial transactions will be performed. Refer to the *IBM WebSphere Commerce Payments Administrator's Guide*, for a complete description of the WebSphere Commerce Payments administration objects. The Cassette for VisaNet augments three of the Framework administration objects with its own attributes. VisaNet Administration objects are described in detail in Chapter 7, "Using VisaNet objects" on page 43.

Cassette Admin object

The CassetteAdmin object represents the cassette itself and contains attributes that apply globally across the cassette. The Cassette for VisaNet extends this object with attributes that tell the cassette how to connect to the VisaNet Host.

Account Admin object

In the WebSphere Commerce Payments object model, the AccountAdmin object represents a relationship between a given merchant and a given financial institution. This is exactly the type of relationship that each VisaNet merchant account represents. The cassette extends the WebSphere Commerce Payments AccountAdmin object with attributes that identify and describe the corresponding AccountAdmin merchant account.

PaySystem Admin object

Each PaySystemAdmin object represents configuration data that is different for each merchant, but common across all accounts for the given merchant. This term is synonymous with Merchant Cassette Settings.

The Cassette for VisaNet allows the merchant to define multiple accounts in the PaySystemAdmin object.

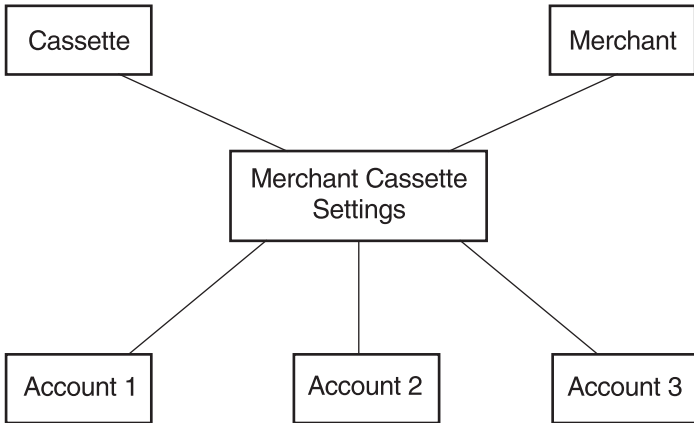


Figure 1. VisaNet PaySystem

Financial objects

The WebSphere Commerce Payments financial objects are used to represent the financial transactions executed by merchants. The Cassette for VisaNet provides extensions for each of these financial objects:

- Order objects
- Payment objects
- Credit objects
- Batch objects

For details on how the Cassette for VisaNet extends these payment objects, see Chapter 7, “Using VisaNet objects” on page 43. For descriptions of the payment objects and for programming information, see the *IBM WebSphere Commerce Payments Programmer’s Guide and Reference*.

Cassette-specific characteristics and behaviors

This section discusses characteristics of communication parameters and the WebSphere Commerce Payments command set that are unique to the Cassette for VisaNet.

Understanding retry parameters

The Cassette for VisaNet extends the WebSphere Commerce Payments Cassette object with several parameters related to communicating with the VisaNet host. Two of these parameters control the attempts of the cassette to recover after failed communications with the VisaNet host. These parameters appear on the VisaNet Cassette Settings screen as follows:

- Attempt Interval
- Max Attempts

You can modify any of the Cassette Settings values through the user interface (select Cassettes under the navigation frame, then select the VisaNet cassette icon, then select Advanced Settings) or through the ModifyCassette API command. For more information on the ModifyCassette command, see Chapter 6, “Using WebSphere Commerce Payments commands” on page 35.

Under direction of these parameters, the cassette will attempt to recover as follows:

- For each command that requires communication with the VisaNet host, a socket connection must be established between the WebSphere Commerce Payments and the VisaNet host. The cassette will attempt to connect to the VisaNet host in accordance with the VisaNet specifications (EIS 1051 and EIS 1052), which dictate the immediate timeout value and the number of immediate retries to attempt before declaring a communications failure. If the cassette is unable to establish this connection, it will not retry the operation and will return a communication error.
- If a connection is established with the VisaNet host, but some other communications error occurs (such as a socket read failure or no response from the VisaNet host), then the cassette will enter "delayed retry" logic. Specifically, delayed retries work as follows:
 - The cassette will return a return code that indicates the operation is pending (PRC_OPERATION_PENDING)
 - The request message is queued and waits a predetermined amount of time as specified by the cassette setting called **Attempt Interval**.
 - Once the attempt interval expires, the request is removed from the internal queue and is retried.
 - The process of queuing the request and retrying the operation is repeated until the request is completed or until the maximum number of communication attempts is reached. The maximum number of communication attempts is specified by the **Max Attempts** value in the cassette settings.

Cassette for VisaNet payment command summary

Table 2 summarizes the way the Cassette for VisaNet handles each of the WebSphere Commerce Payments payment commands (that is the commands that carry out financial transactions). Specifically, for each payment command, the table shows:

- Which payment card function will be performed by the command (using terminology more common to the payment card industry)
- How the cassette processes the command:
 - "Not supported by cassette" means the cassette does not support that particular command. These commands will always receive return codes PRC_CASSETTE_ERROR, RC_COMMAND_NOT_SUPPORTED.
 - "Handled by WebSphere Commerce Payments; no message sent" means that the command is processed completely within the WebSphere Commerce Payments without communicating with a VisaNet host.
 - In any other case, the primary VisaNet command (or commands) used to accomplish the function will be shown.

Table 2. Cassette for VisaNet. Summary of Payment API Commands

API Command	Payment Card Functions	VisaNet Message
AcceptPayment	No comparable function	Handled by WebSphere Commerce Payments; no message sent
AcceptPayment with AutoApprove	Authorize	Authorization
AcceptPayment with AutoApprove and AutoDeposit	Authorize	Authorization
Approve	Authorize	Authorization
Approve with AutoDeposit	Authorize	Authorization

Table 2. Cassette for VisaNet (continued). Summary of Payment API Commands

ApproveReversal	Authorize reversal	Authorization with appropriate reversal information
BatchClose	Close an existing batch	Data Capture
BatchOpen	Open a new batch	Not supported by cassette (Cassette opens batches internally as needed.)
BatchPurge	Purge an existing batch	Handled by WebSphere Commerce Payments; no message sent
CancelOrder	No comparable function	Handled by WebSphere Commerce Payments; no message sent
CloseOrder	No comparable function	Handled by WebSphere Commerce Payments; no message sent
DeleteBatch	No comparable function	Handled by WebSphere Commerce Payments; no message sent
Deposit	No comparable function	Handled by WebSphere Commerce Payments; no message sent
DepositReversal	No comparable function	Handled by WebSphere Commerce Payments; no message sent
ReceivePayment	No comparable function	Not supported by cassette
Refund	No comparable function	Handled by WebSphere Commerce Payments; no message sent
RefundReversal	No comparable function	Handled by WebSphere Commerce Payments; no message sent

Summary of state changes

The following table summarizes the state changes that Order, Payment, Credit and Batch objects undergo as a result of successful completion of each payment command. Only those objects whose states actually change as a result of the given operation are shown. Any other existing object states remain unchanged.

API Command	Object-State
AcceptPayment	ORDER_REFUNDABLE
AcceptPayment with AutoApprove	ORDER_REFUNDABLE PAYMENT_APPROVED
AcceptPayment with AutoApprove and AutoDeposit	ORDER_REFUNDABLE PAYMENT_DEPOSITED
Approve	PAYMENT_APPROVED
Approve with AutoDeposit	PAYMENT_DEPOSITED
ApproveReversal, amount is non-0	PAYMENT_APPROVED
ApproveReversal, amount=0	PAYMENT_VOID
CancelOrder	ORDER_CANCELED
CloseOrder	ORDER_CLOSED
Deposit	PAYMENT_DEPOSITED
DepositReversal	PAYMENT_APPROVED
Refund	CREDIT_REFUNDED

RefundReversal	CREDIT_VOID
BatchClose	BATCH_CLOSED PAYMENT_CLOSED CREDIT_CLOSED ORDER_REFUNDABLE
DeleteBatch	Deletes the batch

Protecting sensitive data

As an option, you can prevent sensitive financial data such as credit card numbers and expiry dates from being returned in query results when users enter query commands. A Payment Servlet parameter called `wpm.MinSensitiveAccessRole` can be specified to define the minimum access role a user must have to view sensitive data returned in query command results. (Refer to the *IBM WebSphere Commerce Payments Administrator's Guide* for information on how to set this and other Payment Servlet initialization parameters.)

When a user enters a query through a query command, WebSphere Commerce Payments checks the user's role against the minimum role specified for the `wpm.MinSensitiveAccessRole` parameter and determines whether sensitive data should be returned in full view or masked out. The following table lists the data elements that are considered sensitive by the Cassette for VisaNet:

Table 3. . Sensitive data processed by Cassette for VisaNet

Data	How data is protected
\$PAN	Cardholder's card number. All but the last 4 digits of the card number are masked with asterisks.
\$EXPIRY	Card expiration date. The entire value is masked with asterisks.
\$CARDVERIFYCODE	Verification code for the payment card. The entire value is masked with asterisks.

Supported minimum sensitive access role values are: clerk, supervisor, merchant administrator, payments administrator, or none. If the `wpm.MinSensitiveAccessRole` parameter is not specified, an access role of clerk is assumed, which allows all users to see sensitive data. If the user's role matches or exceeds the role value, the actual values are displayed for the sensitive data.

For more information about query commands, refer to the *IBM WebSphere Commerce Payments Programmer's Guide and Reference*.

Chapter 3. Installing the Cassette for VisaNet

The WebSphere Commerce Payments must be installed before the Cassette for VisaNet can be installed. The WebSphere Commerce Payments installation will ensure that all prerequisite products are available. The minimum framework level supported by the cassette is 3.1. For detailed information on the WebSphere Commerce Payments Framework, including hardware and software prerequisites, refer to the *IBM WebSphere Commerce Payments for Multiplatforms Installation Guide*, Version 3.1.

Before installing Cassette for VisaNet

- Read the latest README file, *readme.VisaNet.html*, accessed through documentation links on the WebSphere Commerce Payments Web site: <http://www.ibm.com/software/commerce/payment/support/index.html> and on the Cassette for VisaNet CD-ROM.
- WebSphere Commerce Payments should *not* be running at cassette installation. WebSphere Application Server *should be* running at cassette installation.

Note: A prior version of the cassette cannot be installed on top of the WebSphere Commerce Payments Version 3.1.1 Framework. If you currently use a prior version of the Cassette for VisaNet, you must install the WebSphere Commerce Payments Version 3.1.1 Cassette for VisaNet software for your cassette data to be migrated and compatible with the WebSphere Commerce Payments Version 3.1.1 Framework.

- Starting with Payment Manager Version 3.1.1, changes were made to WebSphere Commerce Payments directory structure and configuration of WebSphere Application Server relative to that of earlier releases of Payment Manager. For more information about the installed directory structure, see: “WebSphere Commerce Payments directory structure” on page 14.

Note: iSeries does not require that WebSphere Commerce Payments or WebSphere Application Server be ended during installation.

Installing Cassette for VisaNet (Windows® and UNIX platforms)

This section describes the procedure for installing the Cassette for VisaNet on Windows NT®, Windows 2000, Solaris, AIX® and Linux®. Before installing the Cassette for VisaNet software, you should stop the WebSphere Commerce Payments Application Server from the WebSphere Application Server administrative console. This ensures that the configuration files for WebSphere Commerce Payments will be freed to enable the cassette installation program to update the files. (If you are installing more than one type of payment cassette, you must stop the WebSphere Commerce Payments Application Server before installing each cassette).

- On AIX, Linux and Solaris, you must logon as **root**.
 - On Windows NT and Windows 2000, you must logon as a user who is a member of the administrator group.
1. Once logged in, insert the CD-ROM containing WebSphere Commerce Payments Cassette for VisaNet.
 - For Windows NT, go to the *nt* directory.
 - For Windows 2000, go to the *nt* directory.
 - For Solaris, go to the *solaris* directory.

- For Linux, go to the *linux* directory.
 - For AIX, go to the *aix* directory.
2. Enter **InstallVisaNetCassette** to start the installation. Enter the information requested on the installation screens.
 3. The *IBM Cassette for VisaNet Readme* screen indicates that the configuration of the Cassette for VisaNet has successfully completed and lets you display the README, if desired.

WebSphere Commerce Payments directory structure

Starting with Payment Manager Version 3.1.1, some changes were made to the WebSphere Commerce Payments directory structure and configuration of WebSphere Application Server relative to that of Payment Manager Version 3.1.0 and earlier releases. Under WebSphere Application Server Version 4.0, WebSphere Commerce Payments now makes use of Web archive (WAR) and enterprise archive (EAR) files. After the WebSphere Commerce Payments framework is installed, a `WPMApplication.ear` file representing the WebSphere Commerce Payments application is found in the `<Payments_installdir>` deployable subdirectory. The EAR file has a subdirectory structure containing a `Payments.war` file, Web files, and other files used to configure WebSphere Application Server. Some files that you may have seen in the WebSphere Commerce Payments installation directory prior to version 3.1.1 are now moved into the EAR file structure and deployed to a WebSphere Application Server directory. In WebSphere Application Server, the WebSphere Commerce Payments EAR file becomes a subdirectory under the `installedApps` subdirectory (for example, on Windows: `<WAS_DIR>\installedApps\IBM_Payments.ear`).

The following Cassette for VisaNet files are moved into the EAR file (or EAR directory for iSeries):

```
<Payments_installdir>\eTillVisaNetClasses.zip
<Payments_installdir>\web\*
```

Under this revised directory structure, if you need to make changes to the WebSphere Commerce Payments (Payment Servlet) initialization parameters, you should refer to the *IBM WebSphere Commerce Payments Administrator's Guide* for instructions on using the WebSphere Application Server Application administrative console.

Installing Cassette for VisaNet (iSeries)

This section describes the procedure for installing the Cassette for VisaNet for iSeries and adding the cassette to a WebSphere Commerce Payments instance.

Installing the cassette

- Use the Restore License Program (**RSTLICPGM**) CL command to install the Cassette for VisaNet option of the IBM WebSphere Commerce Payments for iSeries product.
- Specify the WebSphere Commerce Payments product number, option 3 for the Cassette for VisaNet and the device from which the product is to be installed. For example:

```
RSTLICPGM LICPGM(5733PY3) DEV(OPT01) OPTION(3).
```


Adding a Cassette for VisaNet to a WebSphere Commerce Payments Instance

After installing the Cassette for VisaNet, you need to add the VisaNet cassette to a WebSphere Commerce Payments instance. This process enables the cassette to be used by that instance. Be sure none of the following are running when you begin adding the Cassette for VisaNet to a WebSphere Commerce Payments instance:

- The WebSphere Commerce Payments instance
- The HTTP Server that processes payment requests for the WebSphere Commerce Payments instance
- The Websphere Application Server that processes payment requests for the WebSphere Commerce Payments instance

To add the Cassette for VisaNet to a WebSphere Commerce Payments instance:

- Access the iSeries Tasks page at **http://system-name:2001** where the system-name is the TCP/IP host name of the iSeries system.
- Select the **WebSphere Commerce Payments** icon.
- Select the WebSphere Commerce Payments instance from the drop-down menu.
- Select the **Work Cassettes** menu.
- Select **VisaNet** from the cassette list and press the **Add** button to add the cassette.

Note: Alternately, you can use the (**ADDPYMCSS**) CL command.

When the process for adding the cassette to a WebSphere Commerce Payments instance has completed, the cassette-specific tables are added to the WebSphere Commerce Payments instance database collection. You are now ready to start the WebSphere Commerce Payments and configure the VisaNet cassette.

Uninstalling Cassette for VisaNet on Windows NT

On Windows NT, you can remove the Cassette for VisaNet using the following steps:

1. Use the WebSphere Application Server administrative console to stop the WebSphere Commerce Payments Application Server.
2. Go to the Windows NT Control Panel.
3. Click the **Add/Remove Programs** icon.
4. Select the **IBM WebSphere WebSphere Commerce Payments Cassette for VisaNet**.
5. Click **Add/Remove**

Note: This process removes all of the Cassette for VisaNet tables that were installed on your system, including those containing financial transaction data.

Uninstalling Cassette for VisaNet on Windows 2000

On Windows 2000, you can remove the Cassette for VisaNet using the following steps:

1. Use the WebSphere Application Server administrative console to stop the WebSphere Commerce Payments Application Server.
2. Go to the Windows 2000 Control Panel.
3. Click the **Add/Remove Programs** icon.

4. Select the **IBM WebSphere Commerce Payments Cassette for VisaNet**.
5. Click **Add/Remove**.

Note: This process removes all of the Cassette for VisaNet tables that were installed on your system, including those containing financial transaction data.

Uninstalling Cassette for VisaNet on Solaris

On Solaris, you can remove the Cassette for VisaNet using the following steps:

1. Set your display and xhost. From a command prompt, enter:

```
export DISPLAY <machine_name:0.0>  
xhost + <machine_name>
```
2. Use the WebSphere Application Server administrative console to stop the WebSphere Commerce Payments Application Server.
3. Logon as **root**.
4. Change to the WebSphere Commerce Payments install directory. By default, WebSphere Commerce Payments is installed to `/opt/Payments`.
5. From the command prompt, run this script:

```
./UninstallVisaNet.sh
```

Note: This process removes all of the Cassette for VisaNet tables that were installed on your system, including those containing financial transaction data.

Uninstalling Cassette for VisaNet on AIX

On AIX, you can remove the Cassette for VisaNet using the following steps:

1. Use the WebSphere Application Server administrative console to stop the WebSphere Commerce Payments Application Server.
2. Enter **smit** to display the SMIT System Management menu (SMIT main menu).
3. Navigate to the dialog panel that allows you to remove software products.
4. Display your installed software.
5. Select all components that begin with **WebSphere.Commerce.Payments.Cassette.for.VisaNet** and click **OK**.
6. Make sure the Preview Only field is set to **No**. Select **OK** to remove the Cassette for VisaNet.

Note: This process removes all of the Cassette for VisaNet tables that were installed on your system, including those containing financial transaction data.

Uninstalling Cassette for VisaNet on Linux

On Linux, you can remove the Cassette for VisaNet using the following steps:

1. Set your display and xhost. From a command prompt, enter:

```
export DISPLAY <machine_name:0.0>  
xhost + <machine_name>
```
2. Use the WebSphere Application Server administrative console to stop the WebSphere Commerce Payments Application Server.
3. Logon as **root**.
4. Change to the WebSphere Commerce Payments install directory. By default, WebSphere Commerce Payments is installed to `/opt/Payments`.

5. From the command prompt, run this script:

```
./UninstallVisaNet.sh
```

Note: This process removes all of the Cassette for VisaNet tables that were installed on your system, including those containing financial transaction data.

Removing Cassette for VisaNet from a WebSphere Commerce Payments Instance for iSeries

Removing the Cassette for VisaNet from a WebSphere Commerce Payments instance will remove all VisaNet configuration and transaction data from that instance. Be sure none of the following are running when you begin removing the Cassette for VisaNet from a WebSphere Commerce Payments instance:

- The WebSphere Commerce Payments instance
- The HTTP Server that processes payment requests for the WebSphere Commerce Payments instance
- The Websphere Application Server that processes payment requests for the WebSphere Commerce Payments instance

To remove the Cassette for VisaNet to a WebSphere Commerce Payments instance:

- Access the iSeries Tasks page at <http://system-name:2001> where the system-name is the TCP/IP host name of the iSeries system.
- Select the **WebSphere Commerce Payments** icon.
- Select the WebSphere Commerce Payments instance from the drop-down menu.
- Select the **Work Cassettes** menu.
- Select **VisaNet** from the cassette list and press the **Remove** button to add the cassette.

Note: Alternately, you can use the **(RMVPYMCSS)** CL command.

Uninstalling Cassette for VisaNet for iSeries

To uninstall the Cassette for VisaNet option from the system, use the Delete License Program **(DLTLICPGM)** CL command to delete option 3 from the WebSphere Commerce Payments product. For example:

```
DLTLICPGM LICPGM(5733PY3) OPTION(3).
```

Note: All WebSphere Commerce Payments instances must be ended before uninstalling the Cassette for VisaNet.

Chapter 4. Getting Started

Use the information here to configure the Cassette for VisaNet. At this point, you should have completed the following:

- For Windows and UNIX platforms:
 - Installed and configured the WebSphere Commerce Payments Framework.
 - Installed the Cassette for VisaNet.
 - Started the WebSphere Application Server and Web Server (if necessary).
 - Defined a WebSphere Commerce Payments user with administrative authority.
 - Created a merchant and Merchant Administrator for the merchant.
- For iSeries:
 - Installed and configured the WebSphere Commerce Payments Framework.
 - Created a WebSphere Commerce Payments instance and configured the WebSphere Commerce Payments Framework.
 - Added the Cassette for VisaNet to the WebSphere Commerce Payments instance.
 - Started the Instance.
 - Started the WebSphere Application Server and Web Server (if necessary).
 - Defined a WebSphere Commerce Payments user with administrative authority.
 - Created a merchant and Merchant Administrator for the merchant.
 - Assigned a password to the qpyadm (WebSphere Commerce Payments Administrator) user profile.

To configure a cassette, you must logon to WebSphere Commerce Payments as a Merchant administrator. For information on performing these tasks, see the *WebSphere Commerce Payments Administrator's Guide for Multiplatforms, Version 3.1*.

Cassette for VisaNet tutorial

After installing the Cassette for VisaNet, you must configure the cassette before you can process customer transactions. This tutorial will show you how to configure the Cassette for VisaNet. For detailed information on administration, configuration, and payment functions, see the Online Help for the WebSphere Commerce Payments user interface.

This chapter demonstrates everything you must do to achieve a fully functioning Cassette for VisaNet. This information walks you through fictitious scenarios that simulate real-world functions. And while you need not complete the entire walk-through, it is important that you complete these tasks to become familiar with the common Cassette for VisaNet tasks:

1. Configure the Cassette.
2. Specify Merchant Cassette Settings.
3. Create an account.

In addition to the required configuration tasks above, we will walk through common payment processing tasks.

Before starting this tutorial

There are a number of configuration steps that require information from your VisaNet merchant account or your financial institution. Refer to the Merchant registration process section in Chapter 2 for information on establishing production and test accounts.

Starting the WebSphere Commerce Payments User Interface

Our first task is enabling a merchant to use the Cassette for VisaNet. This must be done by a user with Payments Administrator access.

To start the WebSphere Commerce Payments user interface:

1. Point your browser to **http://<hostname>/webapp/PaymentManager/**, where <hostname> is the machine where the WebSphere Commerce Payments is installed.
2. On the WebSphere Commerce Payments Logon window, type the Payments Administrator's user ID and password and click **Logon**.

Configuring the Cassette

During this tutorial, your WebSphere Commerce Payments will attempt to send messages to the VisaNet System. If your computer is behind a firewall and must use a SOCKS server to access sites outside of your internal network, then specify a Socks Host Name and Port Number to enable the Cassette for VisaNet for SOCKS.

1. Click on **Cassettes** in the navigation frame.
2. Click on the **VisaNet** cassette icon. At the next window, you will see several entry fields that allow you to tailor the way the Cassette for VisaNet communicates with the VisaNet Host. Enter the following:

Host Name	The TCP/IP host name to access the VisaNet Host.
Port Number	The TCP/IP port number to access the VisaNet Host.
SOCKS Host Name	TCP Host address for SOCKS server.
SOCKS Port Number	TCP Port Number for SOCKS server.

3. Click on **Update** to update your cassette configuration.
4. Stop the Cassette.
5. Start the Cassette so the settings will take effect.

Configuring the Cassette for SOCKS (iSeries)

During this tutorial, WebSphere Commerce Payments will attempt to send messages through the internet to the VisaNet System. If your computer is behind a firewall and must use a SOCKS server to access sites outside of your internal network, then you may have to perform certain steps to enable the Cassette for VisaNet for SOCKS.

If you have already configured the iSeries TCP/IP support to use a SOCKS server, the iSeries will automatically route connections to the SOCKS server. Thus, you will not have to change the configuration for the VisaNet Cassette.

For more information about configuring the iSeries TCP/IP SOCKS support, see OS/400 TCP/IP Configuration and Reference, Version 4.0

You can access this manual from the iSeries on-line library at:
<http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm>.

1. Select **TCP/IP**.
2. Select **Getting Started**.

If you choose not to configure SOCKS within iSeries TCP/IP support, then perform the following steps to enable the Cassette for VisaNet for SOCKS.

1. Click on **Cassettes** in the navigation frame.
2. Click on the **VisaNet** cassette icon. At the next window, you will see several entry fields that allow you to tailor the way the Cassette for VisaNet communicates with the VisaNet Host. Enter the following:

Host Name	The TCP/IP host name to access the VisaNet Host.
Port Number	The TCP/IP port number to access the VisaNet Host.
SOCKS Host Name	TCP Host address for SOCKS server.
SOCKS Port Number	TCP Port Number for SOCKS server.

3. Click on **Update** to update your cassette configuration.
4. Stop the Cassette.
5. Start the Cassette so the settings will take effect.

Selecting a WebSphere Commerce Payments merchant and authorizing a cassette

If you haven't already created a merchant, you must do that first and authorize that merchant to use a payment cassette. To create a merchant, you must log into the WebSphere Commerce Payments as an administrator:

1. From the navigation frame, click **Merchant Settings** under the Administration section.
2. From the Merchant Settings window, select the Test store merchant created during the WebSphere Commerce Payments tutorial (See the *IBM WebSphere Commerce Payments Administrator's Guide*, for details on the WebSphere Commerce Payments tutorial), or create a new merchant with merchant number 123456789.

Note: If there are more than 500 merchants in the WebSphere Commerce Payments database when you access the Merchant Settings window, you are prompted to search for a specific merchant or merchants. If you see this prompt, enter 123456789.

3. At the next window, you will be allowed to authorize use of the Cassette for VisaNet.

Authorized cassettes	Check the box next to <i>VisaNet</i> . Checking this box authorizes the merchant to use the Cassette for VisaNet.
-----------------------------	---

4. When you have entered the requested information, click **Update (or CreateMerchant)** to save the merchant configuration.
5. You will also have to give the user ID Merchant Administrator authority for this merchant. For instructions on assigning roles, see the *IBM WebSphere Commerce Payments Administrator's Guide*.

Configuring a merchant

After you have enabled the Test Store to use the Cassette for VisaNet, you will need to create the settings for that merchant.

To enter the merchant settings:

1. Click **Merchant Settings** on the navigation frame of the WebSphere Commerce Payments user interface.
2. From the Merchant Settings window, click the **Cassette for VisaNet** icon in the Test Store.
3. From the Cassette for VisaNet window, click **Merchant Cassette Settings**.
4. At the next window, you will be prompted to enter the following information:

Country Code	This is the number that was assigned by the signing member or processor to identify the merchant's location country (e.g., 840 for U.S.).
City Code	Within the U.S., this should be the 5 or 9 digit zipcode of the address of the store location.
VisaNet Merchant Number	This is the 1–12 digit merchant number that was assigned by the merchant's bank or processor.
VisaNet Merchant Name	This is the name that was provided by the signing member or processor.
Time Zone Differential	This is a code that is used to calculate the local time within the VisaNet authorization system. This code is assigned by the signing agent.
Merchant Category Code	This is the number that was assigned by the signing member or processor to identify a merchant industry classification.
Merchant Location	This is the 11 character merchant location. For Direct Marketing merchants, this field will contain a customer service phone number in xxx-xxxxxxx format. The dash is required.
Merchant State	The 2 character field containing the merchant's state or province code provided by the signing member or processor.
VNumber	This is an optional number that was assigned by the signing member or processor to identify a POS device tracking number to be used during Settlement.

Creating an account

So far, you have enabled one merchant, the Test Store, to use the Cassette for VisaNet and you have entered the merchant cassette settings. Now, you need to establish an *account* for the Cassette for VisaNet.

An account is a relationship between the merchant and the financial institution which processes transactions for that merchant. There can be multiple accounts for each payment cassette. But for the purposes of this tutorial, you will create one account for the Cassette for VisaNet.

To create an account:

1. Click **Merchant Settings** on the navigation frame of the WebSphere Commerce Payments user interface.

- From the Merchant Settings window, click the **Cassette for VisaNet** icon in the Test Store.
- From the Cassette for VisaNet window, click **Accounts**.
- Click **Add an Account** on the Accounts window.
- At the next window, you will be prompted to enter the following information (note that the italicized text *must* be entered in these fields for the tutorial):

Account name	Enter <i>VisaNet Account</i> . This is the name that you assign to the account. Its only function is to provide display information in the user interface.
Account number	Enter <i>1</i> . This is a number that you (that is, the hosting service provider or the merchant administrator) assign to uniquely identify this account in all transaction data.
Financial Institution name	Enter <i>VisaNet Bank</i> . This is the name of the financial institution with which you hold this account. Its only function is to provide display information in the user interface.
Acquirer BIN	This is the Visa-assigned Bank Identification Number (BIN) issued by the merchant's member bank or processor.
Agent Bank Number	This value is provided to the merchant by the acquirer.
Agent Chain Number	This value is assigned by the merchant's bank or processor.
Store Number	This is a number assigned by the signing member, processor, or merchant to identify a specific merchant store within the VisaNet system.
Terminal Number	This is a number assigned by the signing member, processor, or merchant to identify a unique terminal within a merchant location.

- Click **Create Account** to create the new account.

Logging in as the merchant administrator

To logoff and login again:

- Click **Logoff admin** (for Windows and UNIX platforms) or **Logoff qpymadm** (for iSeries) on the navigation frame of the WebSphere Commerce Payments user interface. You will return to the main WebSphere Commerce Payments Logon window. From the main WebSphere Commerce Payments Logon window, type the user ID (with Merchant Administrator authority for the Test Store merchant) and the password and click **OK**.

You are now logged in to the WebSphere Commerce Payments user interface with Merchant administrator authority for the Test Store merchant. For the remainder of the tutorial, you will act as the Merchant administrator. Notice that your view of the WebSphere Commerce Payments user interface is now limited to merchant administration functions, whereas as the Payments administrator, you had a global view of both *merchant* and *WebSphere Commerce Payments* administration.

Managing payment processing

As the Merchant administrator, you have global *merchant* authority, which means that you can perform:

- Merchant-specific administration functions
- All payment processing functions

In a real business scenario, you may choose to delegate payment processing tasks to other merchant-defined users who possess limited payment processing authorities (such as, supervisor and clerk). In this tutorial, you, as the Merchant administrator, will perform these tasks.

Having completed all of the WebSphere Commerce Payments and Merchant administration tasks necessary to begin payment processing, you are now ready to start:

- Approving orders
- Depositing payments
- Settling batches
- Issuing credits
- Viewing daily batch totals

For the purposes of this tutorial, you will use the VisaNet Cassette Sample Checkout to create three orders for use in payment processing.

Creating orders using the Sample Checkout

A real business environment features a customer who creates orders using a merchant's Internet storefront and a merchant who processes payments for those orders using the WebSphere Commerce Payments. In order for you to walk through the WebSphere Commerce Payments payment processing functions, you need to create orders that require payment processing. To simulate a merchant's Internet storefront and facilitate order creation, WebSphere Commerce Payments supplies a Sample Checkout. To access the WebSphere Commerce Payments Sample Checkout and create orders:

1. Point your browser to **http://<hostname>/webapp/PaymentManager/SampleCheckout/**, where **<hostname>** is the machine where the WebSphere Commerce Payments is installed.
2. At the Sample Checkout window that appears, you will be prompted to enter the following (note that the italicized text *must* be entered in these fields for the tutorial):

Merchant number	Enter "123456789", the number used when creating the merchant, to represent a Merchant number.
Order number	Enter any unique number to represent an Order number.
Amount	Enter \$25 to represent the total numeric amount of the order.
Currency	Select <i>US dollar</i> . The currency used to place this order.
Payment method	Choose <i>VisaNet</i> as the payment method.
Card number	Use the test credit card obtained from the Merchant registration process.
Expiration date	Highlight the expiration month and year for your credit card. Note: You can choose any future month and year combination for this tutorial.
Card verification value	Enter a 3–4 digit number to represent the verification code printed on the signature panel of the card.
Street address	This field must include your street address.
Zip code	This field must include your zip code. This must be a 5 or 9 digit number.

3. Click **Buy**.

Repeat these steps two more times so that you have three orders for which to process payments.

Approving orders with the Sale Function

The sale function allows you to approve and deposit payments with one command. The Cassette for VisaNet also supports doing this separately. We will discuss this later in the tutorial.

Once you have created three orders using the Sample Checkout, return to the browser window, where the WebSphere Commerce Payments user interface is displayed and log in as the Merchant Administrator for the Test Store merchant.

Note: If you used the same browser window to access the Sample Checkout, you will need to point your browser once again to the WebSphere Commerce Payments URL (that is, **http://<hostname>/webapp/PaymentManager/** and login as the merchant administrator.

Follow these steps to approve and deposit an order:

1. From the navigation frame, click **Approve** under the Payment Processing section.
2. From the Approve window, check the box next to the order that you want to approve and deposit (select only one order for this exercise) and click **Sale Selected**.
3. At the Approve Results window, you will see the status of your sale request. When processing is complete, success or failure status will appear next to each order submitted for sale.
4. When your sale is complete, click **Return to the Approve Screen**.

Two orders are still awaiting sale. You could have approved them all at once (for their full amounts), by clicking **Sale All** from the Approve window. Instead, you will work with each order individually to better demonstrate the many facets of the Approve function.

Approving orders from the Order window

In this section, you will approve and deposit an order from the Order window (rather than from the Approve window), but you will approve only *part* of the total order amount.

1. From the Approve window, click the **Order number** for one of the remaining orders awaiting approval.
2. From the Order window, you can view order details. Click **Sale** to approve and deposit this order.
3. The Order Approve window displays the following information:

Currency	The type of currency used to place this order. This is a read-only field.
Order Amount	The total amount of the order expressed in the currency used to place the order. This is a read-only field.
Approved Amount	This field displays zeros since no amount of the order has yet been approved. This is a read-only field.
Deposited Amount	This field displays zeros since no amount has yet been approved or deposited. This is a read-only field.

Sale Amount	This is an entry field that currently contains the total amount of the order.
--------------------	---

Change the sale amount to **3.00** and click **Sale** to approve and deposit this order.

When sale processing is complete, the Order window will display again, with some indication of the success or failure of the sale. You will notice that approval and deposit amounts in the Order window details have been updated to reflect the \$3.00 sale that you have just completed. In addition, you will notice at the bottom of the screen that a new payment is now listed under the Payments section. This is the payment that you just approved and deposited. To view details of the payment, click on the payment number in the Payments section. On the Payment Detail screen, you will see the following information (all fields on this screen are read-only):

Merchant	The name of the merchant.
State	The current state of the Payment (Deposited).
Currency	The currency used for this payment (US Dollars).
Approved Amount	The amount currently approved (3.00).
Deposited Amount	The amount currently deposited (3.00).
Batch Number	The WebSphere Commerce Payments batch into which this payment has been placed.
Account	The account under which this order is being processed (VisaNet Account).
Order URL	A link to a fictitious order description. In a real merchant's online shopping system, this field might be filled in to point to a corresponding entry in the order entry database.
Time Approved	The time that this payment was approved.
Reference Number	The approval code received from the acquirer when the payment was approved.
Payment Type	The payment type used for this order (VisaNet).
blank line	Denotes the end of the WebSphere Commerce Payments generic attributes for this payment. All of the remaining attributes are unique to the Cassette for VisaNet.
Approval Code	If request is approved, the Authorization code contains the approval code. It is returned in Authorization Response and resubmitted during settlement. It is also resubmitted for an Authorization Reversal Request.
Auth Response Code	Status of authorization request. An AuthResponseCode of "00" indicates approval. An AuthResponseCode of "85" indicates successful card verification and approval. All other codes indicate non-approval.
Auth Response Text	Auth response message. This message contains additional information about the authorization result.
Auth Source Code	Source of the authorization code.
CVV2 Result Code	CVV2 Result code returned in the Authorization Response.
Local Transaction Date	Date calculated by the authorizing system using the time zone differential.
Local Transaction Time	Time calculated by the authorizing system.

Transaction ID	May contain a Visa Transaction Identifier or MasterCard Reference Number.
Validation Code	May contain specific information generated by the card issuer.
Returned ACI	Returned Authorization Characteristics Indicator. This value provides information concerning the transaction's CPS qualification status.
AVS Result Code	Address Verification result code.
Common AVS Code	Common Address Verification Service result code. For more information, see "Address Verification Service (AVS) result codes" on page 52.
Payment Sequence Number	The sequence number that this payment represents in the total count of installment payments being made.
Transaction Sequence Number	Cassette-generated transaction sequence number.

Approving multiple orders at one time

Once you have finished viewing the Payment details, return to the Approve window by clicking on **Approve** in the navigation frame. Since you only approved and deposited a portion of the order in the previous step, there are still two order entries in this window; the one that has been partially approved and the one that is still awaiting approval. In this exercise, you will approve and deposit the complete unapproved purchase amount for each of these in one operation. To do this:

1. From the navigation frame, click **Approve** under the Payment Processing section.
2. Click **Sale All** in the Approve window.
3. In the Approve Results window, you will see a progress bar indicating the status of your sale request. When processing is complete, success or failure status will appear next to each order submitted for sale. Upon successful completion of this request, the order which you partially approved and deposited earlier contains a second payment (for the remaining amount). The third order contains one payment for the entire order amount.
4. When this step is complete, click **Return to the Approve Screen**.

Settling batches

In this exercise, you will *settle* the batch that contains the payments you have created so far. The procedure for settling batches is as follows:

1. From the navigation frame, click **Batch Search** under the Payment Processing section.
2. At the Batch Search window, you will be prompted to enter the following information (note that for the purposes of this tutorial, you can fill in either the payment type: *VisaNet* or the account):

Merchant	The name of the merchant whose batch you are searching for. Note: If there are fewer than 500 merchants in the WebSphere Commerce Payments database, select the merchant name from the drop-down list. If there are more than 500 merchants in the WebSphere Commerce Payments database, enter the merchant number.
Batch Number	The number that uniquely identifies the batch within the merchant. Assigned when the payment is deposited.

State	The state of the batch: <ul style="list-style-type: none"> • Open • Closed
Status	The balance status of this batch: <ul style="list-style-type: none"> • Balanced: the batch has been successfully balanced (that is, all totals agree). • Out of balance: an unsuccessful attempt has been made to balance this batch (that is, all totals do not agree).
Payment Type	Identifies the payment type, or protocol, used to place the order. Select VisaNet .
Batch Open Date	Use the <i>after</i> and <i>before</i> fields below to search for batches opened during the specified range in time: <ul style="list-style-type: none"> • After: Specify a date to search for all batches opened on and after this date. • Before: Specify a date to search for all batches opened on and before this date.
Batch Closed Date	Use the <i>before</i> and <i>after</i> fields below to search for batches closed during the specified range in time: <ul style="list-style-type: none"> • After: Specify a date to search for all batches closed on and after this date. • Before: Specify a date to search for all batches closed on and before this date.
Account	The account under which this order is being processed (VisaNet Account).

3. Click **Search** to initiate a batch search.
4. Click the batch number to view information about the batch.
5. From the Batch window, you can view useful batch information, including the total number and amount of both payments and credits in the batch. Click **Batch Details** to see a detailed listing of all payments and credits in this batch. You will see the four payments you just created and no credits.
6. Click **Settle** to settle the batch. When processing is complete, you will return to the Batch window, which will contain a success or failure indication for the settlement request.

Issuing a credit

Credits are issued against orders and can be given for any amount. To issue a credit, you need to find the order against which you are issuing the credit:

1. From the navigation frame, click **Order Search** under the Payment Processing section.
2. At the Order Search window, you will be prompted to enter the following information (note that for the purposes of this tutorial, you will not be entering any parameter information in the fields to narrow your search):

Merchant	The name of the merchant whose order you are searching for.
Order Number	A number assigned by the merchant that uniquely identifies the order.

State	The state of the order: <ul style="list-style-type: none"> • Requested • Ordered • Refundable • Canceled • Closed
Payment Type	Identifies the payment type, or protocol, used to place the order. Select VisaNet .
Order Date	Use the <i>after</i> and <i>before</i> fields below to search for orders opened during the specified range in time: <ul style="list-style-type: none"> • After: Specify a date to search for all orders opened on and after this date. • Before: Specify a date to search for all orders opened on and before this date.
Order Amount	<ul style="list-style-type: none"> • Currency: The currency used to place this order. Select the currency type from the drop-down list. • Greater than: Specify a value to retrieve all orders with order amounts that are greater than or equal to the value you specify. • Less than: Specify a value to retrieve all orders with order amounts that are less than or equal to the value you specify.
Account	The account under which this order is being processed (VisaNet Account).

3. Click **Search** to initiate an order search.
4. From the next window, click an order number for an order in Refundable state, to view the details of that order.
5. From the Order window, click **Credit** to create a credit against this order.
6. At the Create Credit window, the following information displays:

Currency	The type of currency used to place this order. This is a read-only field.
Order Amount	The total amount of the order expressed in the currency used to place the order. This is a read-only field.
Approved Amount	This field displays the approved amount. This is a read-only field.
Deposited Amount	This field displays the deposited amount. This is a read-only field.
Credit Amount	This is the credit amount of the order.

Enter the credit amount and click **Credit**.

When credit processing has completed, you will be returned to the Order window and notified of credit success or failure. You will notice on the Order window that the newly created credit appears under **Credits** at the bottom of the window. If you view details, you will see the Transaction Date and Transaction Time.

Viewing batch totals

The last exercise in this tutorial is viewing daily batch totals. The WebSphere Commerce Payments Reports function allows you to view *daily totals* for batches in Closed state. (See the *WebSphere Commerce Payments Administrator's Guide* for more information on batch states.) To generate a daily batch totals report:

1. From the navigation frame, click **Reports** under the Payment Processing section.
2. From the Reports window, click **Daily Batch Totals**.
3. At the Batch Totals Report window, you will be prompted to enter the date for which you would like a batch totals report. WebSphere Commerce Payments will compute the batch totals for the date entered and generate a report. *Leave this field blank to generate a report for the current date.*
4. Click **Search** to generate the batch totals report.

The Daily Batch Totals report computes the totals for all batches that were closed on the date specified on the Search screen. These totals will be computed on a per-currency basis, so there will be one line per currency. Note that these totals cover all payments and credits made for all payment types (not just those made through the Cassette for VisaNet).

Assuming that you have not closed any other batches for US Dollars today, you should see one line that indicates you deposited four payments totalling \$75.00. Note that the number of credits in this report is zero, since the batch which contains credits you just created has not yet been closed.

Note: If you have stepped through other tutorials or have closed other batches using the Test Store today, then the totals you see will not match those described above.

You have just completed a day in the life of a Payments administrator and a Merchant administrator. While individual business models may vary, this tutorial outlines the basic path to establishing a working WebSphere Commerce Payments and demonstrates fundamental payment processing implemented through the Cassette for VisaNet. For more information on specific fields in the WebSphere Commerce Payments user interface, see the online Help.

Separate approvals and deposits

VisaNet Cassette allows you to do approvals and deposits separately. A brief description of these actions follows.

Approve

Approval without deposit is performed through the same windows as the Sale function (that is, the Approve or Order windows). Instead of choosing the **Sale**, **Sale Selected**, or **Sale All** buttons as described, use the **Approve**, **Approve Selected** or **Approve All** buttons.

Deposit

Once a Payment has been created and approved via the Approve function, you must use the Deposit function to actually place the payment in the batch. As demonstrated in “Approving multiple orders at one time” on page 27, multiple payments can be associated with a single order. Therefore, you may see the same order number appear multiple times in the same list, each time with different payment information. To deposit a payment that has previously been approved:

1. From the navigation frame, click **Deposit** under the Payment Processing section.
2. Check the box next to each of the listed payments that you want to deposit and then click **Deposit Selected**.

3. In the Deposit Results window, you will see a progress bar indicating the status of your deposit request. When processing is complete, success or failure status will appear next to each order submitted for deposit.
4. When this step is complete, click **Return to the Deposit Screen**.

Note that a **Deposit All** button is also available in the Deposit screen, should you want to deposit the full approval amount of all non-deposited payments. This operates much like the **Sale All** and **Approve All** buttons that you have already seen.

You may deposit only *part* of a payment, in much the same way that you can approve or sale only part of an order:

1. From the Deposit window, click the payment number for the payment that you want to partially deposit.
2. The Payment window appears, as described in “Approving orders from the Order window” on page 25. Click the **Deposit** button at the bottom of this screen to deposit all or part of the approved amount.
3. On the Deposit Payment screen, change the deposit amount to a value less than the full approval amount and click **Deposit**.
4. When the deposit has been processed, you will return to the Payment window, which will be updated with the new deposit amount.

Chapter 5. Cassette for VisaNet Cashier Profiles

The Cashier is WebSphere Commerce Payments software that can be invoked by client applications - such as merchant software - to simplify the process of creating WebSphere Commerce Payments orders and payments. The Cashier uses XML documents called profiles that describe how orders should be created for a given cassette. This allows the client code writer to concentrate on integrating with the WebSphere Commerce Payments in a generic way rather than having to write code that deals with cassette-specific information.

It is still possible to create WebSphere Commerce Payments orders without using the Cashier; programs can use the client access library or the HTTP/XML interface to use the AcceptPayment API. However, the use of the Cashier is preferred since it allows the potential for new cassettes to be introduced to the system without the need for rewriting any code. For more information on the Cashier, see the *WebSphere Commerce Payments Programmer's Guide and Reference for Multiplatforms*.

A Cashier profile represents a description of how WebSphere Commerce Payments orders should be created for a particular payment method. Profiles are XML documents that contain all the information needed by the Cashier to create WebSphere Commerce Payments API requests to create orders for a cassette supporting that payment method. All profiles must include the following data:

- An indication of whether a wallet is used (this flag will be used to determine whether the Cashier should use the AcceptPayment or ReceivePayment command)
- Required WebSphere Commerce Payments parameters
- Required cassette parameters
- Specifications for how the Cashier should supply values for each of the above parameters

In addition, profiles may also contain the following optional data:

- An indication of which WebSphere Commerce Payments instance to use for each profile
- Optional WebSphere Commerce Payments parameters
- Optional cassette parameters
- Buy page information that specifies how client code should build buy pages to collect buyer information. For example, the buy page information might contain an HTML form that collects credit card information required by a specific cassette
- An indication of whether diagnostic information is to be enabled for the profile

Cashier profiles allow parameter values to be specified in four different ways:

1. Hard-coded as constants in the profile
2. Passed as an environment variable on the *CollectPayment()* call
3. Specified as originating from a relational database field
4. Specified as being calculated by Cashier extension code

The Cassette for VisaNet Cashier profile is in the directory: <<<**directory where WebSphere Commerce Payments is installed**>>>/profiles/. For details on designing and tailoring profiles, please see Chapter 3 of the *WebSphere Commerce Payments Programmer's Guide and Reference*, Version 3.1. The following is the

Cassette for VisaNet Cashier profile:

Table 4. Cassette for VisaNet Cashier Profile

VisaNet Cashier profile	Function
SampleCheckoutVisaNet.profile	Uses full AVS information

Note: For iSeries: The Cassette for VisaNet Cashier profile is in the directory:
/QIBM/UserData/PymSvr/profiles/

Chapter 6. Using WebSphere Commerce Payments commands

This chapter describes for each included WebSphere Commerce Payments API command:

- All VisaNet-specific protocol parameters
- Any special notes related to the Cassette for VisaNet handling of Framework parameters.

Note: For any Framework commands, that are not listed here, there are no specific VisaNet parameters or unique behaviors. See the *IBM WebSphere Commerce Payments Programmer's Guide and Reference*, for a complete listing of generic Framework commands.

Cassette for VisaNet Commands

The following section outlines information specific to the VisaNet protocol for the parameters on WebSphere Commerce Payments commands. This information serves as a supplement to the command information contained in the *IBM WebSphere Commerce Payments Programmer's Guide and Reference*.

AcceptPayment

The AcceptPayment command causes a generic order and a VisaNet cassette order to be created. The ApproveFlag for AcceptPayment can be set to "0", "1", or "2". The default setting is "0". An ApproveFlag of "0" indicates that the transaction should not be approved. An ApproveFlag of "1" indicates that the transaction should be approved automatically. An ApproveFlag of "2" indicates that the transaction should be approved asynchronously. If the DepositFlag is set to "1", the Payment is added to the currently open batch. If a batch is not currently open, one will be created implicitly. See the *IBM WebSphere Commerce Payments Programmer's Guide and Reference* for more information on Asynchronous Auto Approve.

Required keywords for AcceptPayment command

Required Keywords	Value
\$EXPIRY	Specifies card expiration date. The value is specified as a 6 character string, in the form of "YYYYMM." For example: "200207" (for July 2002.)
\$PAN	Specifies card account number. The value is specified as a 5–22 digit numeric string. For example: "2222333344445555."
\$ACCOUNTNUMBER	Specifies the account number associated with the request.
\$AVS.STREETADDRESS	Specifies the cardholder's street address for the Address Verification Service. The value is specified as a 1–24 character string.
\$AVS.POSTALCODE	Specifies the cardholder's 5- or 9-digit postal code for the Address Verification Service. The value is specified as a string. For example: "37614".

Optional keywords for AcceptPayment command

Optional Keywords	Value
-------------------	-------

Optional keywords for AcceptPayment command

\$PURCHASEORDERNUMBER	Optional Purchase Order Number assigned by the merchant. A 1–25 character value.
\$CARDVERIFYCODE	3–4 character Card Verification Value used to assist in authenticating the physical presence of a card. This value appears as additional characters embossed on the credit card signature line following the credit card account number. If the \$CARDVERIFYCODE represents an American Express CID, a separate result code will not be returned. In the case of the CID not matching, American Express will not authorize the transaction (i.e., a non-approval response code on the Approve).
\$NUMBEROFPAYMENTS	Specifies the total number of payments or installments associated with this order. The default is 1.
\$SECURECONNECTION	Specifies if the connection between the consumer and the merchant is secure (e.g. SSL). A value of "0" (the default) means that the connection is not secure; a value of "1" means that the connection is secure. The setting of this field affects the ECI (electronic commerce indicator) that is passed in the authorization request.
\$PCARD.LOCALTAXAMOUNT	If specified, contains the local tax amount for the order. VisaNet only supports sending one tax amount, thus, it is not valid to send both a NationalTaxAmount and a LocalTaxAmount. The cassette will detect this condition and throw an exception if this occurs.
\$PCARD.NATIONALTAXAMOUNT	If specified, contains the national sales tax amount for the order. VisaNet only supports sending one tax amount, thus, it is not valid to send both a NationalTaxAmount and a LocalTaxAmount. The cassette will detect this condition and throw an exception if this occurs.
\$PCARD.TAXEXEMPTINDICATOR	If specified, indicates if the order is tax exempt. Supported values are: 0 - indicates that the order is not tax exempt 1 - indicates that the order is tax exempt.
\$PCARD.CUSTOMERREFERENCENUMBER	A 16 or 17 character field indicating a customer reference ID.

Approve

The Approve command causes a generic payment and VisaNet cassette payment to be created, and an Authorization message to be sent to the associated financial institution. If the DepositFlag is set to "1", the Payment is added to the currently open batch. If a batch is not currently open, one will be created implicitly.

The following table presents cassette-specific processing of Framework parameters.

Cassette-specific processing of parameters for the Approve command

Optional Keywords	Value
BATCHNUMBER	Not allowed. Must not be specified because all batches are opened implicitly.

ApproveReversal

VisaNet supports both full reversals and partial reversals. The ApproveReversal command causes the specified payment to be retrieved and reversed. If the AMOUNT is "0", a full reversal is performed and the payment moves into Void state. If the Amount is non-"0", a partial reversal is performed and the old payment is voided and a new payment is created and approved with the amount passed in on the API . In either situation, an AuthorizationReversal message is sent to the financial institution. Due to the fact that the ApproveReversal is not supported by some brands, the ApproveReversal API will always result in a voided payment (if full reversal) and good return codes.

BatchOpen

This command is not supported. All batches are opened implicitly. If this command is issued with PAYMENTTYPE set to "VisaNet" the command will fail with the following return codes:

- PRC_COMMAND_NOT_SUPPORTED
- RC_NONE

BatchPurge

The BatchPurge command is not supported by the acquirer, thus it is a local operation only. The BatchPurge causes all the associated payments and credits to be removed from the batch, with payments returning to the Approved state and credits returning to the Void state. In addition, the batch is returned to Open state. Batch Purge is valid when a batch close has been attempted but the batch did not reconcile and the batch is in open state.

Note: Even though the batch is re-opened, it is considered inactive. No new transactions will be added to it.

BatchClose

The BatchClose command causes a Data Capture message to be sent to the financial institution. If the operation is successful, all associated payments and credits are moved to the Closed state. In addition, the batch is placed in the Closed state. In the VisaNet cassette, batch total comparisons are performed by the acquirer, not the merchant. If a bad response is received from the financial institution, the associated payments and credits will stay in their current state. In this situation, reconciliation is done manually and offline. When a BatchClose

command is received, the current batch is put in Closing state and a new batch is implicitly opened to ensure that new transactions will not be added to the batch we are trying to close.

This command is fully supported as documented in the *IBM WebSphere Commerce Payments Programmer's Guide and Reference* .

CassetteControl

The CassetteControl command is not supported. This command will fail with the following return codes:

- PRC_COMMAND_NOT_SUPPORTED
- RC_NONE.

CloseOrder

The Delete option may be used only if every Batch containing one or more of the Payments or Credits has already been Closed.

CreateAccount

Required keywords for CreateAccount command

Required Keywords	Value
\$ACQUIRERBIN	The 6–digit field containing the Visa assigned Bank Identification Number (BIN) issued by the merchant's member bank or processor.
\$AGENTBANKNUMBER	The 1–6 digit field that identifies the agent of the acquirer which signed the merchant. This value is provided to the merchant by the acquirer.
\$AGENTCHAINNUMBER	The 1–6 digit field containing a specific chain of an agent organization. Assigned by the merchant's bank or processor.
\$STORENUMBER	The 1–4 digit field containing a number assigned by the signing member, processor, or merchant to identify a specific merchant store within the VisaNet system.
\$TERMINALNUMBER	The 1–4 digit field containing a number assigned by the signing member, processor, or merchant to identify a unique terminal within a merchant's location.

CreatePaySystem

Required keywords for CreatePaySystem command

Required Keywords	Value
\$VISANETMERCHANTNUMBER	The 1–12 digit Merchant number assigned by the merchant's bank or processor.
\$COUNTRYCODE	A 3-digit number assigned by the signing member or processor to identify a merchant's location country.

Required keywords for CreatePaySystem command

\$CITYCODE	Used to further identify the merchant location. Within the U.S., the 5 or 9 digit zip code of the address of the store location is used. Outside the U.S., this field will be assigned by the signing member or processor.
\$TIMEZONEDIFFERENTIAL	A 3-digit code used to calculate the local time within the VisaNet authorization system. The differential is calculated by the signing member or processor, providing the standard local time zone differential from GMT.
\$MERCHANTCATEGORYCODE	A 1–4 digit field containing a number assigned by the signing member or processor to identify a merchant’s industry classification.
\$VISANETMERCHANTNAME	A 1-25 character string. The merchant name provided by the signing member or processor.
\$MERCHANTLOCATION	An 11–character merchant location. This is a field that contains a customer service phone number in xxx-xxxxxx format. The dash is required.
\$MERCHANTSTATE	A 2 character field containing the merchant state or province code provided by the signing member or processor.

Optional keyword for CreatePaySystem command

Optional Keywords	Value
\$VNUMBER	A 1–8 digit number assigned by the signing member or processor to identify a POS device tracking number to be used during Settlement. Defaults to "00000001" if not specified.

DeleteBatch

The DeleteBatch command removes the specified Batch from the database. A Batch can be deleted only if the Batch is in Closed state.

Deposit

The Deposit command causes the specified payment to be added to the currently open batch. If a batch is not currently open, one is created. This command does not cause a message to be sent to the financial institution. It is a local operation only. If the operation is successful, the payment moves from Approved state to Deposited state.

Optional keywords for Deposits command

Optional Keywords	Value
-------------------	-------

Optional keywords for Deposits command

\$PCARD.LOCALTAXAMOUNT	If specified, contains the local tax amount for the order. VisaNet only supports sending one tax amount, thus, it is not valid to send both a NationalTaxAmount and a LocalTaxAmount. The cassette will detect this condition and throw an exception if this occurs.
\$PCARD.NATIONALTAXAMOUNT	If specified, contains the national sales tax amount for the order. VisaNet only supports sending one tax amount, thus, it is not valid to send both a NationalTaxAmount and a LocalTaxAmount. The cassette will detect this condition and throw an exception if this occurs.
\$PCARD.TAXEXEMPTINDICATOR	If specified, indicates if the order is tax exempt. Supported values are: 0 - indicates that the order is not tax exempt 1 - indicates that the order is tax exempt.
\$PCARD.CUSTOMERREFERENCENUMBER	A 16 or 17 character field indicating a customer reference ID.

DepositReversal

The DepositReversal command causes the specified payment to be removed from the currently open batch. This command is a local operation only and does not cause a message to be sent to the financial institution. This command is valid for payments in Deposited state. If the command is successful, the payment moves to Approved state.

ModifyAccount

Optional keywords for ModifyAccount command

Optional Keywords	Value
\$ACQUIRERBIN	The 6-digit field containing the Visa assigned Bank Identification Number (BIN) issued by the merchant's member bank or processor.
\$AGENTBANKNUMBER	Identifies the agent of the acquirer which signed the merchant. This is a 1–6 digit value and it is provided to the merchant by the acquirer.
\$AGENTCHAINNUMBER	The 1–6 digit field containing a specific chain of an agent organization. Assigned by the merchant's bank or processor.
\$STORENUMBER	The 1–4 digit field containing a number assigned by the signing member, processor, or merchant to identify a specific merchant store within the VisaNet system.
\$TERMINALNUMBER	The 1–4 digit field containing a number assigned by the signing member, processor, or merchant to identify a unique terminal within a merchant location.

ModifyCassette

Optional keywords for ModifyCassette command

Optional Keywords	Value
\$ATTEMPTINTERVAL	Number of seconds to wait until trying the next set of retries. An integer between 0 and 2147483647. If not specified, the default is "60" (1 minute).
\$MAXATTEMPTS	Maximum number of retry sets. An integer between 0 and 2147483647. If not specified, the default is "3."
\$SOCKSHOSTNAME	TCP Host Address for socks server.
\$SOCKSPORTNUMBER	TCP Port Number for socks server.
\$HOSTNAME	The TCP/IP host name to access the VisaNet Host.
\$PORTNUMBER	The TCP/IP port number to access the VisaNet Host.

ModifyPaySystem

Optional keywords for ModifyPaySystem command

Optional Keywords	Value
\$VISANETMERCHANTNUMBER	The 1–12 digit Merchant number assigned by the merchant's bank or processor.
\$COUNTRYCODE	A 3-digit number assigned by the signing member or processor to identify a merchant's location country.
\$CITYCODE	Used to further identify the merchant location. Within the U.S., the 5 or 9 digit zip code of the address of the store location is used. Outside the U.S., this field will be assigned by the signing member or processor.
\$TIMEZONEDIFFERENTIAL	The 3-digit code used to calculate the local time within the VisaNetauthorization system. The differential is calculated by the signing member or processor, providing the standard local time zone differential from GMT.
\$MERCHANTCATEGORYCODE	The 1–4 digit field containing a number assigned by the signing member or processor to identify a merchant's industry classification.
\$VISANETMERCHANTNAME	The 1–25 character merchant name.
\$MERCHANTLOCATION	The 11 character merchant location. For Direct Marketing merchants, this field will contain a customer service phone number in xxx-xxxxxxx format. The dash is required.
\$MERCHANTSTATE	The 2 character field containing the merchant's state or province code provided by the signing member or processor.

ReceivePayment

This command is not supported because the cassette does not support order creation through a wallet. If this command is issued with PAYMENTTYPE set to "VisaNet" the command fails with the following return codes:

- PRC_COMMAND_NOT_SUPPORTED.
- RC_NONE.

Refund

The Refund command causes a generic Credit object and a VisaNet cassette Credit object to be created and added to the currently open batch. If a batch is not currently open, one is created. The Refund command is valid only if the associated Order is in the Refundable state. If the operation is successful, the Credit is put in Refunded state. When the Refund command is issued, a message is not sent to the financial institution.

Cassette-specific processing of parameters for the Refund command.

Required Keyword	Value
BATCHNUMBER	Is not allowed. Must not be specified because all batches are opened implicitly.

RefundReversal

The RefundReversal command causes the specified credit to be removed from the currently open batch. This command does not cause a message to be sent to the financial institution. The RefundReversal command is valid for credits in the Refunded state only. If the RefundReversal command is successful, the credit moves to the Void state.

Chapter 7. Using VisaNet objects

The object model of the Cassette for VisaNet closely reflects the generic model of the WebSphere Commerce Payments. This section describes each of the cassette extensions to the various Framework objects, as well as new objects defined exclusively by the cassette.

The WebSphere Commerce Payments query command set allows merchant software to search for and retrieve the data objects maintained in the WebSphere Commerce Payments database. The results of each query call are returned in the form of an XML PSApiResult document. Cassette for VisaNet object extensions appear in these documents as extensions to the generic objects of the Framework.

Financial objects used by Cassette for VisaNet

Each of the Framework's generic financial objects is extended by the Cassette for VisaNet.

VisaNet Order

Cassette properties that belong to a POrder Object

Field Name	Description
expiry	The Credit or Debit card's expiration date in the form YYYYMM as specified on the AcceptPayment command. This value is always present.
pan	The Credit or Debit card number as specified on the AcceptPayment command. This value is always present.
numberOfPayments	The number of payments associated with this order. This field is used only if the order has recurring payments or installments associated with it.
avsPostalCode	Specifies the cardholder's 5- or 9-digit postal code for the Address Verification Service.
purchaseOrderNumber	The purchase order number as specified on the AcceptPayment command.
avsStreet	Specifies the cardholder's street address for the Address Verification Service. The combination of street address and postal code cannot exceed a length of 24 characters for VisaNet.
cvv2	Card Verification Value that is used to assist in authenticating the physical presence of a card.
taxExemptIndicator	Indicates that the order is tax exempt, i.e., no taxes are applied to the order.
localTaxAmount	The total amount of local tax applied to the order.
nationalTaxAmount	The total amount of national tax (sales or VAT) applied to the order.
customReferenceNumber	A reference number assigned to the order by the Cardholder.
secureConnection	Returns value of the \$SECURECONNECTION parameter.

VisaNet Order object XML example

This XML example shows an Order object and its cassette extensions:

```

<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="3" primaryRC="0" secondaryRC="0">
<OrderCollection size="3" withCredits="0" withPayments="0">
<PSOrder ID="0:123456789:33" amount="7500" amountExp10="-2"
approvesAllowed="1" currency="840" merchantAccount="1"
merchantNumber="123456789" merchantOriginated="1"numberOfCredits="0"
numberOfPayments="0" orderNumber="33" paymentType="VisaNet"
state="order_refundable" timeStampCreated="968460959000"
timeStampModified="968461231000"unapprovedAmount="0">
<CassetteExtensionObject>
<CassetteProperty propertyId="numberOfPayments" value="1">
</CassetteProperty>
<CassetteProperty propertyId="expiry" value="200411">
</CassetteProperty>
<CassetteProperty propertyId="pan" value="4111111111111111">
</CassetteProperty>
<CassetteProperty propertyId="avsPostalCode" value="27511">
</CassetteProperty>
<CassetteProperty propertyId="avsStreet" value="123 Memory Lane">
</CassetteProperty>
<CassetteProperty propertyId="cvv2" value="529">
</CassetteProperty>
</CassetteExtensionObject>
</PSOrder>
<PSOrder ID="0:123456789:34" amount="4399" amountExp10="-2"
approvesAllowed="1" currency="840" merchantAccount="1"
merchantNumber="123456789" merchantOriginated="1"numberOfCredits="0"
numberOfPayments="0" orderNumber="34" paymentType="VisaNet"
state="order_refundable" timeStampCreated="968461024000"
timeStampModified="968461279000" unapprovedAmount="0">
<CassetteExtensionObject>
<CassetteProperty propertyId="numberOfPayments" value="1">
</CassetteProperty>
<CassetteProperty propertyId="expiry" value="200207">
</CassetteProperty>
<CassetteProperty propertyId="pan" value="4060667755430966">
</CassetteProperty>
<CassetteProperty propertyId="avsPostalCode" value="23411">
<CassetteProperty>
<CassetteProperty propertyId="avsStreet" value="111 Main Street">
</CassetteProperty>
<CassetteProperty propertyId="cvv2" value="703">
</CassetteProperty>
</CassetteExtensionObject>
</PSOrder>
<PSOrder ID="0:123456789:35" amount="15688" amountExp10="-2"
approvesAllowed="1" currency="840" merchantAccount="1"
merchantNumber="123456789" merchantOriginated="1" numberOfCredits="0"
numberOfPayments="0" orderNumber="35" paymentType="VisaNet"
state="order_refundable" timeStampCreated="968461103000"
timeStampModified="968461103000" unapprovedAmount="15688">
<CassetteExtensionObject>
<CassetteProperty propertyId="numberOfPayments" value="1">
</CassetteProperty>
<CassetteProperty propertyId="expiry" value="200309">
</CassetteProperty>
<CassetteProperty propertyId="pan" value="7655343434987987834">
</CassetteProperty>
<CassetteProperty propertyId="avsPostalCode" value="29645">
</CassetteProperty>
<CassetteProperty propertyId="avsStreet" value="27 Lombard Street">
</CassetteProperty>
<CassetteProperty propertyId="cvv2" value="1234">
</CassetteProperty>

```

```

</CassetteExtensionObject>
</PSOrder>
</OrderCollection>
</PSApiResult>

```

VisaNet Payment

Cassette properties that belong to a PSPayment Object

Field Name	Description
paySequenceNum	The sequence number that this payment represents in the total count of installment payments being made. The VisaNet Order table contains the total count of installment payments associated with the order.
cvv2ResultCode	CVV2 Result code returned in the Authorization Response.
paymentNumber	The number assigned to this payment.
validationCode	May contain specific information generated by the card issuer.
authResponseCode	The status of the authorization request. An AuthResponseCode of "00" indicates approval. An AuthResponseCode of "85" indicates successful card verification and approval. All other codes indicate non-approval.
avsResultCode	The AVS code received from the acquirer for this payment. This field will be present only when the VisaNet host returns an avs-code value for this payment.
approvalCode	The approval code received from the acquirer for this payment.
localTrxTime	Time calculated by the authorization center using the time zone differential. The time format is HH:MM:SS.
orderNumber	The WebSphere Commerce Payments order number.
trxSequenceNum	Cassette-generated transaction sequence number.
merchantNumber	The WebSphere Commerce Payments merchant number.
retrievalRefNum	The transaction's Retrieval Reference Number.
authResponseText	Authorization response message. This message contains additional information about the authorization result.
localTrxDate	Date calculated by the authorization center using the time zone differential. The date format is MMDDYY.
returnedACI	The Returned Authorization Characteristics Indicator. This value provides information concerning the transaction's CPS qualification status.
authSourceCode	Source of the authorization code.
transactionID	May contain a Visa Transaction Identifier or MasterCard Reference Number.
commonAVSCode	Common Address Verification Service result code. For more information, see "Address Verification Service (AVS) result codes" on page 52.
taxExemptIndicator	Indicates that the order is tax exempt, i.e., no taxes are applied to the order.
localTaxAmount	The total amount of local tax applied to the order.

Cassette properties that belong to a PSPayment Object

nationalTaxAmount	The total amount of national tax (sales or VAT) applied to the order.
customReferenceNumber	A reference number assigned to the order by the Cardholder

Payment object XML Example

This XML example shows a VisaNet Payment object and its cassette extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="2" primaryRC="0" secondaryRC="0">
<PaymentCollection size="2" withOrders="0">
<PSPayment ID="P:123456789:33:1" amountExp10="-2" approveAmount="7500"
batchNumber="4" currency="840" depositAmount="7500" merchantAccount="1"
merchantNumber="123456789" orderNumber="33" paymentNumber="1"
paymentType="VisaNet" referenceNumber="123456789012" state="payment_deposited"
timeStampCreated="96846123000" timeStampModified="968461363000">
<CassetteExtensionObject>
<CassetteProperty propertyId="commonAVSCode" value="0">
</CassetteProperty>
<CassetteProperty propertyId="paySequenceNum" value="1">
</CassetteProperty>
<CassetteProperty propertyId="paymentNumber" value="1">
</CassetteProperty>
<CassetteProperty propertyId="authResponseCode" value="00">
</CassetteProperty>
<CassetteProperty propertyId="avsResultCode" value="X">
</CassetteProperty>
<CassetteProperty propertyId="approvalCode" value="222222">
</CassetteProperty>
<CassetteProperty propertyId="localTrxTime" value="123000">
</CassetteProperty>
<CassetteProperty propertyId="orderNumber" value="33">
</CassetteProperty>
<CassetteProperty propertyId="trxSequenceNum" value="0001">
</CassetteProperty>
<CassetteProperty propertyId="merchantNumber" value="123456789">
<CassetteProperty>
<CassetteProperty propertyId="retrievalRefNum" value="123456789012">
</CassetteProperty>
<CassetteProperty propertyId="authResponseText" value="Order Approved!!">
</CassetteProperty>
<CassetteProperty propertyId="localTrxDate" value="070400">
</CassetteProperty>
<CassetteProperty propertyId="returnedACI" value="A">
</CassetteProperty>
<CassetteProperty propertyId="returnedACI" value="A">
</CassetteProperty>
<CassetteProperty propertyId="authSourceCode" value="9">
</CassetteProperty>
</CassetteExtensionObject>
</PSPayment>
<PSPayment ID="P:123456789:34:1" amountExp10="-2" approveAmount="4399"
currency="840" depositAmount="0" merchantAccount="1"
merchantNumber="123456789" orderNumber="34" paymentNumber="1"
paymentType="VisaNet" referenceNumber="123456789012"
state="payment_approved" timeStampCreated="968461249000"
timeStampModified="968461279000">
<CassetteExtensionObject>
<CassetteProperty propertyId="commonAVSCode" value="0">
</CassetteProperty>
<CassetteProperty propertyId="paySequenceNum" value="1">
</CassetteProperty>
<CassetteProperty propertyId="paymentNumber" value="1">
</CassetteProperty>
```



```

<CassetteProperty propertyId="authResponseCode" value="00">
</CassetteProperty>
<CassetteProperty propertyId="avsResultCode" value="X">
</CassetteProperty>
<CassetteProperty propertyId="approvalCode" value="222222">
</CassetteProperty>
<CassetteProperty propertyId="localTrxTime" value="123000">
</CassetteProperty>
<CassetteProperty propertyId="orderNumber" value="34">
</CassetteProperty>
<CassetteProperty propertyId="trxSequenceNum" value="0002">
</CassetteProperty>
<CassetteProperty propertyId="merchantNumber" value="123456789">
</CassetteProperty>
<CassetteProperty propertyId="retrievalRefNum" value="123456789012">
</CassetteProperty>
<CassetteProperty propertyId="authResponseText" value="Order Approved!!">
</CassetteProperty>
<CassetteProperty propertyId="localTrxDate" value="070400">
</CassetteProperty>
<CassetteProperty propertyId="returnedACI" value="A">
</CassetteProperty>
<CassetteProperty propertyId="authSourceCode" value="9">
</CassetteProperty>
</CassetteExtensionObject>
</PSPayment>
</PaymentCollection>
</PSApiResult>

```

VisaNet Credit

Cassette properties that belong to a PSCredit Object

Field Name	Description
transactionTime	The local system time when the Credit was created. The format is "HHMMSS."
transactionDate	The local system date when the Credit was created. The format is "MMDD."

VisaNet Credit object XML example

This XML example shows a Credit object and its cassette extensions:

```

<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
<CreditCollection size="1" withOrders="0">
<PSCredit ID="C:123456789:34:1" amount="1500" amountExp10="-2"
batchNumber="4" creditNumber="1" currency="840" merchantAccount="1"
merchantNumber="123456789" orderNumber="34" paymentType="VisaNet"
referenceNumber="0908,210118" state="credit_refunded"
timeStampCreated="968461279000" timeStampModified="968461279000">
<CassetteExtensionObject>
<CassetteProperty propertyId="transactionTime" value="210118">
</CassetteProperty>
<CassetteProperty propertyId="transactionDate" value="0908">
</CassetteProperty>
</CassetteExtensionObject>
</PSCredit>
</CreditCollection>
</PSApiResult>

```

VisaNet Batch

Cassette properties that belong to a PSBatch Object

Field Name	Description
batchErrorData	If a batch response indicates "RB" for rejected batch, this field contains error information that can be useful in identifying and correcting the "RB" condition.
batchRspCode	The status of the batch returned from the financial institution. The responses returned, are: <ul style="list-style-type: none">• "GB" for Good Batch• "QD" for Duplicate Batch• "RB" for Rejected Batch.
batchTransDate	The batch transmission date, in the format of MMDD. This date is generated by the cassette and sent in the settlement message.
vitalBatchNumber	The Cassette generated value in the range of 001-999.
currencyCode	The currency code of the Batch.

Batch object XML example

This XML example shows a Batch object and its cassette extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="2" primaryRC="0" secondaryRC="0">
<BatchCollection size="2" withCredits="0" withPayments="0">
<PSBatch ID="B:123456789:3" batchNumber="3" batchStatus="batch_not_yet_balanced"
forceAllowed="0" merchantAccount="1" merchantControl="1" merchantNumber="123456789"
paymentType="OfflineCard" purgeAllowed="1" state="batch_open" timeStampClosed="0"
timeStampModified="968275527000" timeStampOpened="968275527000">
<BatchTotalCollection size="1">
<PSBatchTotal amountExp10="-2" creditAmount="0" currency="840" numberOfCredits="0"
numberOfPayments="1" paymentAmount="7500">
</PSBatchTotal>
</BatchTotalCollection>
<CassetteExtensionObject>
<CassetteProperty propertyId="Currency" value="840">
</CassetteProperty>
</CassetteExtensionObject>
</PSBatch>
<PSBatch ID="B:123456789:4" batchNumber="4" batchStatus="batch_not_yet_balanced"
forceAllowed="0" merchantAccount="1" merchantControl="1" merchantNumber="123456789"
paymentType="VisaNet" purgeAllowed="1" state="batch_open" timeStampClosed="0"
timeStampModified="968461363000" timeStampOpened="968461278000">
<BatchTotalCollection size="1">
<PSBatchTotal amountExp10="-2" creditAmount="1500" currency="840"
numberOfCredits="1" numberOfPayments="1" paymentAmount="7500">
</PSBatchTotal>
</BatchTotalCollection>
<CassetteExtensionObject>
<CassetteProperty propertyId="currencyCode" value="840">
</CassetteProperty>
</CassetteExtensionObject>
</PSBatch>
</BatchCollection>
</PSApiResult>
```

Administration objects used by Cassette for VisaNet

The Cassette for VisaNet uses and extends these Framework objects for WebSphere Commerce Payments administration:

- CassetteAdmin
- AccountAdmin
- PaySystemAdmin

Each administration object is defined by its attributes, or fields. The field names and field descriptions are shown for each administration object.

CassetteAdmin

Cassette properties that belong to a PSCassette Object

Field	Description
hostName	The TCP/IP host name to access the VisaNet Host.
portNumber	The TCP/IP port number to access the VisaNet Host.
socksHostName	TCP Host Address for SOCKS server.
socksPortNumber	TCP Port Number for SOCKS server.
maxAttempts	The maximum number of retry sets. The default is 3 .
attemptInterval	The number of seconds to wait before trying the next set of retries. The default is 60 seconds (1 minute).

Cassette object XML example

This XML example shows a CassetteAdmin object and its cassette extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
<CassetteCollection>
<PSCassette active="1" cassette="VisaNet" changesPending="0"
  companyPkgName="ibm" enabled="1" traceSetting="-1" valid="1">
<CassetteExtensionObject>
<CassetteProperty propertyId="maxAttempts" value="3">
</CassetteProperty>
<CassetteProperty propertyId="socksHostName" value="">
</CassetteProperty>
<CassetteProperty propertyId="attemptInterval" value="60">
</CassetteProperty>
<CassetteProperty propertyId="hostName" value="localhost">
</CassetteProperty>
<CassetteProperty propertyId="portNumber" value="1234">
</CassetteProperty>
<CassetteProperty propertyId="socksPortNumber" value="0">
</CassetteProperty>
</CassetteExtensionObject>
</PSCassette>
</CassetteCollection>
</PSApiResult>
```

AccountAdmin

Cassette properties that belong to a PSMerchantAccount

Field Name	Description
accountNumber	The WebSphere Commerce Payments account number.
acquirerBin	A 6-digit field containing the Visa assigned Bank Identification Number (BIN) issued by the merchant's member bank or processor.
agentBankNumber	Identifies the agent of the acquirer which signed the merchant. This value is provided to the merchant by the acquirer.
agentChainNumber	A 1–6 digit field containing a specific chain of an agent organization. Assigned by the merchant's bank or processor.
merchantNumber	The WebSphere Commerce Payments merchant number.
storeNumber	A 1–4 digit field containing a number assigned by the signing member, processor, or merchant to identify a specific merchant store within the VisaNet system.
terminalNumber	A 1–4 digit field containing a number assigned by the signing member, processor, or merchant to identify a unique terminal within a merchant location.

This XML example shows a AccountAdmin object and its cassette extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
<MerchantAccountCollection>
<PSMerchantAccount active="1" apApproveFlag="0" apDepositFlag="0" cassette="VisaNet"
  changesPending="0" enabled="1" financialInstName="My Bank" merchantAccount="1"
  merchantAccountName="My Account" merchantNumber="123456789" valid="1">
<CassetteExtensionObject>
<CassetteProperty propertyId="agentChainNumber" value="444545">
</CassetteProperty>
<CassetteProperty propertyId="accountNumber" value="1">
</CassetteProperty>
<CassetteProperty propertyId="acquirerBin" value="222222">
</CassetteProperty>
<CassetteProperty propertyId="terminalNumber" value="5656">
</CassetteProperty>
<CassetteProperty propertyId="agentBankNumber" value="222333">
</CassetteProperty>
<CassetteProperty propertyId="merchantNumber" value="123456789">
</CassetteProperty>
<CassetteProperty propertyId="storeNumber" value="5646">
</CassetteProperty>
<CassetteProperty propertyId="batchSequenceNumber" value="0">
</CassetteProperty>
</CassetteExtensionObject>
</PSMerchantAccount>
</MerchantAccountCollection>
</PSApiResult>
</v>
```

PaySystemAdmin

Each PaySystem represents configuration data that are different for each merchant, but common across all accounts for the given merchant. The following describes the VisaNet PaySystem data:

Cassette properties that belong to PSMerchantCassetteSettings

Field Name	Description
merchantLocation	An 11-character merchant location. This is a field that contains a customer service phone number in xxx-xxxxxxx format. The dash is required.
countryCode	A 3-digit number assigned by the signing member or processor to identify a merchant's location country.
cityCode	Used to further identify the merchant location. Within the U.S., the 5 or 9 digit zip code of the address of the store location is used. Outside the U.S., this field will be assigned by the signing member or processor.
visaNetMerchantNumber	The 1–12 digit Merchant number assigned by the merchant's bank or processor.
industryCode	A 3-digit number assigned by the signing member or processor to identify a merchant's location country.
visaNetMerchantName	A 1-25 character string. The merchant name provided by the signing member or processor.
vNumber	Contains a number assigned by the signing member or processor to identify a POS device tracking number.
timeZoneDifferential	A 3-digit code used to calculate the local time within the VisaNet authorization system. The differential is calculated by the signing member or processor, providing the standard local time zone differential from GMT.
merchantCategoryCode	A 1–4 digit field containing a number assigned by the signing member or processor to identify a merchant's industry classification.

This XML example shows a PaySystemAdmin object and its cassette extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
<MerchantCassetteSettingsCollection>
<PSMerchantCassetteSettings active="1" cassette="VisaNet" changesPending="0"
  enabled="1" merchantNumber="123456789" valid="1">
<CassetteExtensionObject>
<CassetteProperty propertyId="visaNetMerchantNumber" value="132168461534">
</CassetteProperty>
<CassetteProperty propertyId="merchantLocation" value="213-1231234">
</CassetteProperty>
<CassetteProperty propertyId="countryCode" value="840">
</CassetteProperty>
<CassetteProperty propertyId="visaNetMerchantName" value="myMerchant">
</CassetteProperty>
```

```

<CassetteProperty propertyId="timeZoneDifferential" value="705">
</CassetteProperty>
<CassetteProperty propertyId="cityCode" value="90210">
</CassetteProperty>
<CassetteProperty propertyId="merchantState" value="CA">
</CassetteProperty>
<CassetteProperty propertyId="industryCode" value="D">
</CassetteProperty>
<CassetteProperty propertyId="vNumber" value="00000009">
</CassetteProperty>
<CassetteProperty propertyId="merchantNumber" value="123456789">
</CassetteProperty>
<CassetteProperty propertyId="merchantCategoryCode" value="9876">
</CassetteProperty>
</CassetteExtensionObject>
</PSMerchantCassetteSettings>
</MerchantCassetteSettingsCollection>
</PSApiResult>

```

Address Verification Service (AVS) result codes

When Address Verification Services (AVS) are requested on an `AcceptPayment` or a `ReceivePayment` command, subsequent approvals will reflect the results of the AVS check by storing the associated AVS result code in the Payment object. Since other credit card-oriented cassettes also support AVS, but may use different result codes, the Payment Server Framework provides a set of common AVS result codes that can be used by any cassette that supports AVS. These common codes relieve merchant software from having to be aware of which cassette is being used. For more information on Address Verification Services, see Chapter 1, “Overview of VisaNet” on page 1.

The following table illustrates the way the Cassette for VisaNet maps the VisaNet-specific AVS result codes to the Payment Server Framework’s common AVS codes. Note that the VisaNet AVS result codes (defined in the left-most column) are sometimes returned as two letters (for example, both the letter X and the letter Y can be returned on a complete AVS match).

Table 5. Payment Server common AVS result codes mapped to Cassette for VisaNet AVS result codes

VisaNet AVS result code (returned from acquirer)	Payment Server common AVS result code	Explanations
X	0	Both the postal code (that is, the AVS 5-digit and 9-digit) and the street address match.
Y	0	Both the postal code (that is, the AVS 5-digit and 9-digit) and the street address match.
A	1	The street address matches, but the postal code does not match.
W	2	The 5-digit or 9-digit postal codes matches, but the street address does not match.
Z	2	The 5-digit or 9-digit postal codes matches, but the street address does not match.

Table 5. Payment Server common AVS result codes mapped to Cassette for VisaNet AVS result codes (continued)

VisaNet AVS result code (returned from acquirer)	Payment Server common AVS result code	Explanations
N	3	Neither the street address nor the postal code matches.
R	4	This constant maps: <ul style="list-style-type: none"> • Address information unavailable • System unavailable (maybe due to timeout) • Card type not supported • Transaction ineligible AVS return codes
S	4	This constant maps: <ul style="list-style-type: none"> • Address information unavailable • System unavailable (maybe due to timeout) • Card type not supported • Transaction ineligible AVS return codes
U	4	This constant maps: <ul style="list-style-type: none"> • Address information unavailable • System unavailable (maybe due to timeout) • Card type not supported • Transaction ineligible AVS return codes
E	4	This constant maps: <ul style="list-style-type: none"> • Address information unavailable • System unavailable (maybe due to timeout) • Card type not supported • Transaction ineligible AVS return codes
G	4	Non-US card issuer that does not participate in AVS.

Note: The query command results will contain both the common as well as the cassette-specific codes.

Appendix A. VisaNet return codes

Almost all of the error conditions raised by the Cassette for VisaNet are reported exclusively through primary and secondary return codes:

- **Primary Return Codes:** Only Framework-defined primary return codes are used. Refer to the *IBM WebSphere Commerce Payments Programmer's Guide and Reference*, for this listing.
- **Secondary Return Codes:** The majority of the secondary return codes generated by the Cassette for VisaNet are defined by the Framework (See the *IBM WebSphere Commerce Payments Programmer's Guide and Reference*, for a listing.) The following table lists VisaNet-specific errors and their definitions.

SRC_CASSETTE_PURCHASEORDERNUMBER	10001	Response refers to the \$PURCHASEORDERNUMBER parameter.
SRC_CASSETTE_NUMBEROFPAYMENTS	10002	Response refers to the \$NUMBEROFPAYMENTS parameter.
SRC_CASSETTE_SECURECONNECTION	10003	Refers to \$SECURECONNECTION
SRC_CASSETTE_ATTEMPTINTERVAL	10005	Response refers to the \$ATTEMPTINTERVAL parameter.
SRC_CASSETTE_MAXATTEMPTS	10006	Response refers to the \$MAXATTEMPTS parameter.
SRC_CASSETTE_INDUSTRYCODE	10007	Response refers to the \$INDUSTRYCODE parameter.
SRC_CASSETTE_CITYCODE	10008	Response refers to the \$CITYCODE parameter.
SRC_CASSETTE_VISANETMERCHANTNUMBER	10009	Response refers to the \$VISANETMERCHANTNUMBER parameter.
SRC_CASSETTE_VISANETMERCHANTNAME	10010	Response refers to the \$VISAMERCHANTNAME parameter.
SRC_CASSETTE_MERCHANTLOCATION	10011	Response refers to the \$MERCHANTLOCATION parameter.
SRC_CASSETTE_MERCHANTSTATE	10012	Response refers to the \$MERCHANTSTATE parameter.
SRC_CASSETTE_TIMEZONEDIFFERENTIAL	10013	Response refers to the \$TIMEZONEDIFFERENTIAL parameter.
SRC_CASSETTE_MERCHANTCATEGORYCODE	10014	Response refers to the \$MERCHANTCATEGORYCODE parameter.
SRC_CASSETTE_COUNTRYCODE	10015	Response refers to the \$COUNTRYCODE parameter.
SRC_CASSETTE_ACQUIRERBIN	10016	Response refers to the \$ACQUIRERBIN parameter.

SRC_CASSETTE_AGENTBANKNUMBER	10017	Response refers to the \$AGENTBANKNUMBER parameter.
SRC_CASSETTE_AGENTCHAINNUMBER	10018	Response refers to the \$AGENTCHAINNUMBER parameter.
SRC_CASSETTE_STORENUMBER	10019	Response refers to the \$STORENUMBER parameter.
SRC_CASSETTE_TERMINALNUMBER	10020	Response refers to the \$TERMINALNUMBER parameter.
SRC_CASSETTE_CARDVERIFYCODE	10021	Response refers to the \$CARDVERIFYCODE parameter.
SRC_CASSETTE_ACCOUNTNUMBER	10022	Response refers to the \$ACCOUNTNUMBER parameter.
SRC_CASSETTE_VNUMBER	10023	Response refers to the \$VNUMBER parameter.
SRC_CASSETTE_HOSTNAME	10024	Response refers to the \$HOSTNAME parameter.
SRC_CASSETTE_PORTNUMBER	10025	Response refers to the \$PORTNUMBER parameter.
SRC_CASSETTE_SOCKSHOSTNAME	10026	Response refers to the \$SOCKSHOSTNAME parameter.
SRC_CASSETTE_SOCKSPORTNUMBER	10027	Response refers to the \$SOCKSPORTNUMBER parameter.
SRC_DEPOSIT_AMOUNT	10028	Response refers to the \$DEPOSITAMOUNT parameter
SRC_METHOD_NOT_IMPLEMENTED	20000	The Vital cassette does not support the framework method (e.g., CreateProtocolRequest).
SRC_CASSETTE_BUNDLE_ID_MISMATCH	20001	Response refers to \$BUNDLEIDMISMATCH parameter.
SRC_ACCOUNT_SELECT_SQL_FAILURE	21000	Database error occurred while attempting to retrieve Vital accounts.
SRC_ACCOUNT_CREATE_SQL_FAILURE	21001	Database error occurred while attempting to create Vital account.
SRC_ACCOUNT_UPDATE_SQL_FAILURE	21002	Database error occurred while attempting to update a Vital account.
SRC_ACCOUNT_DELETE_SQL_FAILURE	21003	Database error occurred while attempting to delete a Vital account.
SRC_ACCOUNT_MORE_THAN_ONE_BATCH	21004	There is more than one batch open for this account.
SRC_BATCH_SELECT_BATCH_MISSING	22001	The batch cannot be resurrected since it is not in the database.
SRC_BATCH_SELECT_SQL_FAILURE	22002	Database error occurred while attempting to retrieve a Vital batch

SRC_BATCH_CREATE_SQL_FAILURE	22003	Database error occurred while attempting to create a Vital batch
SRC_BATCH_UPDATE_SQL_FAILURE	22004	Database error occurred while attempting to update a Vital batch
SRC_BATCH_DELETE_SQL_FAILURE	22005	Database error occurred while attempting to delete a Vital batch
SRC_BATCH_NULL_ORDER_FOR_PAYMENT	22006	Framework associated with the Vital Batch's payment cannot be retrieved
SRC_BATCH_NULL_PAYMENT	22007	Vital Batch payment cannot be retrieved.
SRC_BATCH_BAD_BATCH_IN_PAYMENT	22008	Vital cassette batch number for payment and framework's batch number for payment are not the same.
SRC_BATCH_NULL_ORDER_FOR_CREDIT	22009	Framework associated with the Vital Batch's credit cannot be retrieved
SRC_BATCH_NULL_CREDIT	22010	Vital Batch credit cannot be retrieved.
SRC_BATCH_BAD_BATCH_IN_CREDIT	22011	Vital cassette batch number for credit and framework's batch number for credit are not the same.
SRC_BATCH_PURGE_INCOMPLETE	22012	Unable to complete BatchPurge operation
SRC_BATCH_DUPLICATE	22013	The batch is a duplicate batch
SRC_BATCH_MUST_BE_OUT_OF_BALANCE	22014	The batch purge cannot be completed since the batch is not out of balance
SRC_ORDER_SELECT_ORDER_MISSING	23000	The order cannot be resurrected since it is not in the database.
SRC_ORDER_SELECT_SQL_FAILURE	23001	Database error occurred while attempting to retrieve a Vital order
SRC_ORDER_CREATE_SQL_FAILURE	23002	Database error occurred while attempting to create a Vital order
SRC_ORDER_UPDATE_SQL_FAILURE	23003	Database error occurred while attempting to update a Vital order
SRC_ORDER_DELETE_SQL_FAILURE	23004	Database error occurred while attempting to delete a Vital order
SRC_ORDER_SELECT_CLOSE_FAILURE	23005	Database error occurred while attempting to end a query
SRC_ORDER_UNABLE_TO_OBTAIN_CRYPTO_KEY	23006	Encryption failure
SRC_ORDER_MULTIPLE_PCARD_TAXAMOUNTS_SPECIFIED	23007	Only one purchase card tax may be specified
SRC_PAYMENT_SELECT_PAYMENT_MISSING	24000	The payment cannot be resurrected since it is not in the database.
SRC_PAYMENT_SELECT_SQL_FAILURE	24001	Database error occurred while attempting to retrieve a Vital payment

SRC_PAYMENT_CREATE_SQL_FAILURE	24002	Database error occurred while attempting to create a Vital payment
SRC_PAYMENT_UPDATE_SQL_FAILURE	24003	Database error occurred while attempting to update a Vital payment
SRC_PAYMENT_DELETE_SQL_FAILURE	24004	Database error occurred while attempting to delete a Vital payment
SRC_PAYMENT_SELECT_CLOSE_FAILURE	24005	Database error occurred while attempting to end a query
SRC_PAYMENT_NUMPAYMENTS_EXCEEDED	24006	The number of payments allowed for the order has been exceeded
SRC_PAYMENT_MULTIPLE_PCARD_TAXAMOUNTS_SPECIFIED	24007	Only one purchase card tax may be specified
SRC_CREDIT_SELECT_CREDIT_MISSING	25000	The credit cannot be resurrected since it is not in the database.
SRC_CREDIT_SELECT_SQL_FAILURE	25001	Database error occurred while attempting to retrieve a Vital credit
SRC_CREDIT_CREATE_SQL_FAILURE	25002	Database error occurred while attempting to create a Vital credit
SRC_CREDIT_UPDATE_SQL_FAILURE	25003	Database error occurred while attempting to update a Vital credit
SRC_CREDIT_DELETE_SQL_FAILURE	25004	Database error occurred while attempting to delete a Vital credit
SRC_QUERY_ORD_SELECT_SQL_FAILURE	26000	A database error occurred while processing the result set for a query on cassette orders.
SRC_QUERY_PAY_SELECT_SQL_FAILURE	26001	A database error occurred while processing the result set for a query on cassette payments.
SRC_QUERY_CRE_SELECT_SQL_FAILURE	26002	A database error occurred while processing the result set for a query on cassette credits.
SRC_QUERY_BAT_SELECT_SQL_FAILURE	26003	A database error occurred while processing the result set for a query on cassette batches.
SRC_QUERY_ACC_SELECT_SQL_FAILURE	26004	A database error occurred while processing the result set for a query on cassette accounts.
SRC_QUERY_CASSETTE_SELECT_SQL_FAILURE	26005	A database error occurred while processing the result set for a query on cassette configuration.
SRC_QUERY_PAYSYS_SELECT_SQL_FAILURE	26006	A database error occurred while processing the result set for a query on cassette pay systems.
SRC_CASSETTE_ADMIN_SELECT_SQL_FAILURE	27000	Database error occurred while attempting to retrieve the CassetteAdmin data.
SRC_CASSETTE_ADMIN_UPDATE_SQL_FAILURE	27001	Database error occurred while attempting to update the CassetteAdmin data.

SRC_PAYSYS_NOT_VALID	28000	PaymentSystem is not in a valid state. It is missing required protocol data.
SRC_INVALID_TEST_PROTOCOL_DATA	29000	Invalid \$TEST protocol data (only applies when TestMode is turned on).
SRC_NO_RESPONSE_TO_REQUEST_RETRIES_EXHAUSTED	30000	No authorization response received by VisaNet host and internal recovery attempts have been exhausted.
SRC_CONNECT_FAILED_TO_VISANET_HOST	30001	Unable to open a socket with the configured VisaNet host.
SRC_WRITE_FAILED_TO_VISANET_HOST	30002	Socket error trying to send data to VisaNet host.
SRC_SOCKET_READ_FAILED	30003	Socket error trying to read VisaNet host data.
SRC_TOO_MANY_OUTSTANDING_AUTHS	30004	Unable to send authorization request to VisaNet host because the maximum outstanding authorization requests limit has been reached. This request should be retried.
SRC_OPERATION_INTERRUPTED_BY_CASSETTE_SHUTDOWN	30005	Unexpected internal error
SRC_VISANETLLP_NETWORKING_IS_STOPPED	30006	Internal error because all request preceding the cassette start request will be returned with cassette not in right state src.
SRC_VISANETLLP_PROTOCOL_UNEXPECTED_BYTE_VALUE_RECEIVED	30007	Error in link protocol between the cassette and the VisaNet host. The cassette received an unexpected link protocol byte from the VisaNet host.
SRC_VISANETLLP_PROTOCOL_READ_TIMEOUT_EXPIRED	30008	Timed out waiting for VisaNet host to respond with a control character
SRC_VISANETLLP_READ_PACKET_TOO_LARGE	30009	Batch response received from the VisaNet host was too large.
SRC_VISANET_PACKET_FORMAT_ERROR	30010	Format error in the response received from the VisaNet host.
SRC_BATCH_RECORD_RETRIES_EXHAUSTED	30013	The VisaNet host failed either the LCR or parity check on a settlement record set by the cassette. The cassette has exhausted the re-transmission limit defined by the batch link specification.
SRC_BATCH_RESPONSE_RETRIES_EXHAUSTED	30014	The received batch response failed the LRC or parity check. The cassette has exhausted the request for re-transmission limit defined by the batch link specification.

Appendix B. Cassette for VisaNet Messages

This chapter contains the Cassette for VisaNet-specific messages.

CEPVisaNet1000 This method has not been implemented yet. Method name = *methodName*.

Severity: Information

Explanation:

User Response:

CEPVisaNet1001 The VisaNet Cassette's resource bundle ID does not match the ID passed by the framework. The expected ID = *resourceBundleID*. The ID passed by the framework = *msgID*.

Severity: Error

Explanation: This is an internal cassette error.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet1002 The VisaNet Cassette has started.

Severity: Information

Explanation: The VisaNet Cassette has started and is ready to accept commands.

User Response: None

CEPVisaNet1003 The VisaNet Cassette has shutdown.

Severity: Information

Explanation: The VisaNet Cassette is no longer active.

User Response: None

CEPVisaNet2000 An SQL exception was caught while selecting existing accounts from the database.

Severity: Error

Explanation: An SQL exception occurred while retrieving a record from the Payment Server database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet2001 Unable to create the account. Account number *acctNum* for merchant number *merchNum* already exists.

Severity: Error

Explanation: The specified merchant already has an account for the specified account number.

User Response: Re-attempt the CreateAccount request, specifying an account number that does not already exist.

CEPVisaNet2002 An SQL exception occurred while creating an account for Merchant *merchNum* and Account *acctNum*.

Severity: Error

Explanation: An SQL exception occurred while creating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet2003 An SQL exception occurred while updating an account for Merchant *merchNum* and Account *acctNum*.

Severity: Error

Explanation: An SQL exception occurred while updating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet2004 An SQL exception occurred while deleting the account for Merchant *merchNum* and Account *acctNum*.

Severity: Error

Explanation: An SQL exception occurred while deleting a record in the Payment Server database. This could be due to an error connecting to the database.

User Response: Check the connection to the database to make sure that there is not a problem with

the communication between the Payment Server machine and the database server.

CEPVisaNet2005 An error occurred while attempting to retrieve a batch for Merchant *merchNum*, Account *acctNum*, Order *orderNum*, and Transaction *transNum*.

Severity: Error

Explanation: An SQL exception occurred while deleting a record in the Payment Server database. This could be due to an error connecting to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet2006 An error occurred while attempting to settle a batch for Merchant *merchNum*, Account *acctNum*, Batch *batchNum*. The Payment that is in error is Payment *paymentNum* associated with Order *orderNum*. The field that is in error is *name*. The value that was passed for that field is *value*.

Severity: Error

Explanation: A bad return code ("RB") was returned in the batch settlement response. The error occurred in a Detail record. Batch reconciliation must occur in order to settle the batch.

User Response: Use the information provided in the message to try to do batch reconciliation.

CEPVisaNet2007 An error occurred while attempting to settle a batch for Merchant *merchNum*, Account *acctNum*, Batch *batchNum*. The Credit that is in error is Credit *creditNum* associated with Order *orderNum*. The field that is in error is *name*. The value that was passed for that field is *value*.

Severity: Error

Explanation: A bad return code ("RB") was returned in the batch settlement response. The error occurred in a Detail record. Batch reconciliation must occur in order to settle the batch.

User Response: Use the information provided in the message to try to do batch reconciliation.

CEPVisaNet2008 An error occurred while attempting to settle a batch for Merchant *merchNum*, Account *acctNum*, Batch *batchNum*. The field that is in error is *name*. The value that was passed for that field is *value*.

Severity: Error

Explanation: A bad return code ("RB") was returned in the batch settlement response. The error occurred in the Header, Parameter, or Trailer record. Batch reconciliation must occur in order to settle the batch.

User Response: Use the information provided in the message to try to do batch reconciliation.

CEPVisaNet2009 An error occurred while attempting to retrieve a batch for Merchant *merchNum*, Account *acctNum*, Order *orderNum*, Payment/Credit *transNum*.

Severity: Error

Explanation: A bad return code ("RB") was returned in the batch settlement response. The error occurred in the Header, Parameter, or Trailer record. Batch reconciliation must occur in order to settle the batch.

User Response: Use the information provided in the message to try to do batch reconciliation.

CEPVisaNet2010 An error occurred while attempting to settle a batch for Merchant *merchNum*, Account *acctNum*, Batch *batchNum*. The VisaNet host has indicated that the batch already exists.

Severity: Error

Explanation: A bad return code ("QD") was returned in the batch settlement response.

User Response: Contact the VisaNet host representative to determine why the batch already exists.

CEPVisaNet2011 Information ignored while attempting to settle a batch for Merchant *merchNum*, Account *acctNum*, Batch *batchNum*. The card used in Order *orderNum* is not a commercial card. The purchase card data specified in Payment *payNum* has been ignored.

Severity: Information

Explanation: The card used in the transaction is not a commercial card. The purchase card data associated with the payment has been ignored.

User Response: Contact cardholder to ensure that

future transactions are made with a valid commercial card.

CEPVisaNet3000 Unable to retrieve batch for Merchant *merchNum* and Batch *batchNum*.

Severity: Error

Explanation: The specified batch for the specified merchant was not found in the Payment Server database.

User Response: Ensure that the specified batch exists for the specified merchant.

CEPVisaNet3001 An SQL exception was caught while selecting existing batches from the database.

Severity: Error

Explanation: An SQL exception occurred while retrieving a record from the Payment Server database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet3002 An SQL exception occurred while creating a batch for Merchant *merchNum* and Batch *batchNum*.

Severity: Error

Explanation: An SQL exception occurred while creating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet3003 An SQL exception occurred while updating a batch for Merchant *merchNum* and Batch *batchNum*.

Severity: Error

Explanation: An SQL exception occurred while updating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with

the communication between the Payment Server machine and the database server.

CEPVisaNet3004 An SQL exception occurred while deleting the batch for Merchant *merchNum* and Batch *batchNum*.

Severity: Error

Explanation: An SQL exception occurred while deleting a record in the Payment Server database. This could be due to an error connecting to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet3005 Unable to retrieve order information for a payment in the batch. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Payment Number = *paymentNum*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3006 Unable to retrieve payment information for a payment in the batch. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Payment Number = *paymentNum*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3007 The batch number in the payment is not the same as the batch number of the Batch in which the payment exists. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Payment Number = *paymentNum*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3008 Unable to retrieve order information for a credit in the batch. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Credit Number = *creditNum*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3009 Unable to retrieve credit information for a credit in the batch. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Credit Number = *creditNum*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3010 The batch number in the credit is not the same as the batch number of the Batch in which the credit exists. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Credit Number = *creditNum*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3011 An error occurred while attempting to reverse a deposit during the BatchPurge operation. Merchant Number = *merchNum*. Account Number = *acctNum*. Order Number = *orderNum*. Payment Number = *paymentNum*. Primary Return Code = *prc*. Secondary Return Code = *src*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3012 An error occurred while attempting to reverse a refund during the BatchPurge operation. Merchant Number = *merchNum*. Account Number = *acctNum*. Order Number = *orderNum*. Credit Number = *creditNum*. Primary Return Code = *prc*. Secondary Return Code = *src*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3013 An error occurred while attempting to retrieve an order for a payment in the batch during the BatchPurge operation. Merchant Number = *merchNum*. Account Number = *acctNum*. Order Number = *orderNum*. Payment Number = *paymentNum*. Primary Return Code = *prc*. Secondary Return Code = *src*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet3014 An error occurred while attempting to retrieve an order for a credit in the batch during the BatchPurge operation. Merchant Number = *merchNum*. Account Number = *acctNum*. Order Number = *orderNum*. Credit Number = *creditNum*. Primary Return Code = *prc*. Secondary Return Code = *src*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet4000 Unable to retrieve order for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: The specified order for the specified merchant was not found in the Payment Server database.

User Response: Ensure that the specified order exists for the specified merchant.

CEPVisaNet4001 An SQL exception was caught while selecting existing orders from the database.

Severity: Error

Explanation: An SQL exception occurred while retrieving a record from the Payment Server database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet4002 An SQL exception was caught while closing a SELECT statement.

Severity: Error

Explanation: An SQL exception occurred while attempting to close a SQL SELECT statement.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet4003 An SQL exception occurred while creating an order for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while creating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet4004 An SQL exception occurred while updating an order for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while updating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet4005 An SQL exception occurred while deleting an order for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while deleting a record in the Payment Server database. This could be due to an error connecting to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet4006 An error occurred while attempting to close a payment in an order, due to the fact that the order is in the incorrect state. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Payment Number = *paymentNum*. Current State = *curState*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet4007 An error occurred while attempting to close a credit in an order, due to the fact that the order is in the incorrect state. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Credit Number = *creditNum*. Current State = *curState*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet5000 Unable to retrieve payment for Merchant *merchNum*, Order *orderNum* and Payment *paymentNum*.

Severity: Error

Explanation: The specified payment was not found in the Payment Server database.

User Response: Ensure that the specified payment exists for the specified merchant and the specified order.

CEPVisaNet5001 An SQL exception was caught while selecting existing payments from the database for Merchant *merchNum*, Order *orderNum* and Payment *paymentNum*.

Severity: Error

Explanation: An SQL exception occurred while retrieving a record from the Payment Server database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet5002 An SQL exception was caught while closing a SELECT statement for Merchant *merchNum*, Order *orderNum* and Payment *paymentNum*.

Severity: Error

Explanation: An SQL exception occurred while attempting to close a SQL SELECT statement.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet5003 An SQL exception occurred while creating Payment *payNum* for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while creating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet5004 An SQL exception occurred while updating Payment *payNum* for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while updating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet5005 An SQL exception occurred while deleting Payment *payNum* for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while deleting a record in the Payment Server database. This could be due to an error connecting to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet5006 An error occurred while attempting to close a payment due to the fact that the payment is in the incorrect state. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Payment Number = *payNum*. Current State = *curState*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet6000 An SQL exception was caught while selecting existing credits from the database for Merchant *merchNum*, Order *orderNum* and Credit *creditNum*.

Severity: Error

Explanation: An SQL exception occurred while retrieving a record from the Payment Server database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet6001 An SQL exception occurred while creating Credit *creditNum* for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while creating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data

that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet6002 An SQL exception occurred while updating Credit *creditNum* for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while updating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet6003 An SQL exception occurred while deleting Credit *creditNum* for Merchant *merchNum* and Order *orderNum*.

Severity: Error

Explanation: An SQL exception occurred while deleting a record in the Payment Server database. This could be due to an error connecting to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet6004 An error occurred while attempting to close a credit due to the fact that the credit is in the incorrect state. Merchant Number = *merchNum*. Batch Number = *batchNum*. Order Number = *orderNum*. Credit Number = *credNum*. Current State = *curState*.

Severity: Error

Explanation: This is an internal error in the cassette.

User Response: If the problem persists, contact your IBM service representative.

CEPVisaNet6005 Unable to retrieve credit for Merchant *merchNum*, Order *orderNum* and Credit *creditNum* .

Severity: Error

Explanation: The specified credit was not found in the Payment Server database.

User Response: Ensure that the specified credit

exists for the specified merchant and the specified order.

CEPVisaNet7000 An SQL exception was caught while processing the result set for a query on cassette orders.

Severity: Error

Explanation: An SQL exception occurred processing a result set that was obtained by querying a database view. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet7001 An SQL exception was caught while processing the result set for a query on cassette payments.

Severity: Error

Explanation: An SQL exception occurred processing a result set that was obtained by querying a database view. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet7002 An SQL exception was caught while processing the result set for a query on cassette credits.

Severity: Error

Explanation: An SQL exception occurred processing a result set that was obtained by querying a database view. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet7003 An SQL exception was caught while processing the result set for a query on cassette batches.

Severity: Error

Explanation: An SQL exception occurred processing a result set that was obtained by querying a database view. This could be due to an error connecting to or reading from the database, or due to an error in the

content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet7004 An SQL exception was caught while processing the result set for a query on cassette accounts.

Severity: Error

Explanation: An SQL exception occurred processing a result set that was obtained by querying a database view. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet7005 An SQL exception was caught while processing the result set for a query on cassette payment systems.

Severity: Error

Explanation: An SQL exception occurred processing a result set that was obtained by querying a database view. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet7006 An SQL exception was caught while processing the result set for a query on cassette configuration.

Severity: Error

Explanation: An SQL exception occurred processing a result set that was obtained by querying a database view. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet8000 An SQL exception was caught while selecting existing cassette configuration from the database.

Severity: Error

Explanation: An SQL exception occurred while retrieving a record from the Payment Server database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet8001 An SQL exception occurred while updating Cassette Configuratin in the database.

Severity: Error

Explanation: An SQL exception occurred while updating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet9000 An SQL exception was caught while selecting existing payment systems from the database.

Severity: Error

Explanation: An SQL exception occurred while retrieving a record from the Payment Server database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet9001 An SQL exception occurred while creating a Payment System for Merchant *merchNum*.

Severity: Error

Explanation: An SQL exception occurred while creating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the

database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet9002 An SQL exception occurred while updating a Payment System for Merchant *merchNum*.

Severity: Error

Explanation: An SQL exception occurred while updating a record in the Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet9003 An SQL exception occurred while deleting a Payment System for Merchant *merchNum*.

Severity: Error

Explanation: An SQL exception occurred while deleting a record in the Payment Server database. This could be due to an error connecting to the database.

User Response: Check the connection to the database to make sure that there is not a problem with the communication between the Payment Server machine and the database server.

CEPVisaNet9005 An error occurred while attempting to *en_decrypt* the data.

Severity: Error

Explanation: Unable to encrypt/decrypt the PAN or expiration date.

User Response: Contact your service representative.

CEPVisaNet0606 An internal error occurred: *exception text*.

Severity: Error

Explanation: An internal error occurred in the cassette. The exception text will help IBM service identify the location of the problem.

User Response: If some required operation or service is not functioning properly, contact IBM Service.

Appendix C. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Department TL3B/Building 062
PO Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement or any equivalent agreement between us.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- AIX
- DB2
- IBM
- iSeries

Vital Processing Services is a registered trademark of Vital Corporation.

VisaNet is a registered trademark of Visa International Services Organization.

Lotus and Lotus Domino Go Webserver are trademarks of Lotus Development Corporation in the United States and/or other countries.

Microsoft, Windows NT, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This dictionary defines technical terms used in the documentation for Payment Suite products. It includes IBM product terminology and may include selected terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- Internet Request for Comments: 1208, *Glossary of Networking Terms*
- Internet Request for Comments: 1392, *Internet Users' Glossary*
- The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

The most current *IBM Dictionary of Computing* is available on the World Wide Web at <http://www.networking.ibm.com/nsg/nsgmain.htm>.

The following cross-references are used in this dictionary:

Contrast with:

This refers the reader to a term that has an opposed or substantively different meaning.

See:

This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

Obsolete term for:

This indicates that the term should not be used and refers the reader to the preferred term.

A

access control. In computer security, the process of ensuring that the resources of a computer system can be accessed only by authorized users in authorized ways.

account. An account is a relationship between the merchant and the financial institution which processes transactions for that merchant. There can be multiple accounts for each payment cassette.

account name. The name you assign to the account. Its only function is to provide display information in the user interface.

acquirer. In e-commerce, the financial institution (or an agent of the financial institution) that receives from the merchant the financial data relating to a transaction and initiates that data into an interchange system.

Address Verification Service (AVS). Within IBM e-commerce, a credit and debit card scheme used by merchants to authenticate the cardholder. The merchant requests the cardholder's address and uses AVS to confirm that the cardholder is who he says he is.

ADSM. See ADSTAR Distributed Storage Manager.

ADSTAR Distributed Storage Manager (ADSM). An IBM licensed program that provides storage management and data access services in a multi-vendor, distributed computing environment.

applet. An application program, written in the Java programming language, that can be retrieved from a Web server and executed by a Web browser. A reference to an applet appears in the markup for a Web page, in the same way that a reference to a graphics file appears; a browser retrieves an applet in the same way that it retrieves a graphics file. For security reasons, an applet's access rights are limited in two ways: the applet cannot access the file system of the

client upon which it is executing, and the applet's communication across the network is limited to the server from which it was downloaded. Contrast with servlet.

approve. Within IBM e-commerce, a WebSphere Commerce Payments verb. A merchant issues this verb to create a Payment object. For cassettes that implement credit card protocols, this verb will likely map to authorization (see *authorize*). Other cassettes may implement the approval process differently. For IBM WebSphere Commerce Payments Cassette for SET and Cassette for CyberCash, the **approve** verb results in the creation of a Payment object and authorization to ensure that funds are available to cover payment.

approve all. Selects all orders displayed for approval.

approved amount. The amount of the order approved for payment.

approve selected. Selects the orders that you want to create a payment in the approved state for. You must perform a manual deposit on this payment to move it from approved state to deposit state.

asymmetric. In computer security, pertaining to the use of different keys for encryption and decryption.

authentication. (1) In SETCo., the process that seeks to validate identity or to prove the integrity of the information. Authentication in public key systems uses digital signatures. (2) In computer security, verification that a message has not been altered or corrupted. (3) In computer security, a process used to verify the user of an information system or protected resources.

authorization. (1) In SETCo., the process by which a properly appointed person or persons grants permission to perform some action on behalf of an organization. This process assesses transaction risk, confirms that a given transaction does not raise the account holder debt above the account credit limit, and reserves the specified amount of credit. (When a merchant obtains authorization, payment for the authorized amount is guaranteed provided that the merchant followed the rules associated with the authorization process.) (2) In computer security, the right granted to a user to communicate with or make use of a computer system. (T) (3) An access right. (4) The process of granting a user either complete or restricted access to an object, resource, or function.

authorization reversal. In SETCo., a transaction sent when a previous authorization needs to be canceled (that is, a full reversal performed) or decreased (that is, a partial reversal performed). A full reversal will be used when the transaction cannot be completed, such as when the cardholder cancels the order or the merchant discovers that goods are no longer available, as when discontinued. A partial reversal will be used when the authorization was for the entire order and some of the goods cannot be shipped, resulting in a split shipment.

authorize. In the credit card world, a merchant is guaranteed that cardholder funds are available to cover a transaction by first *authorizing* the transaction. The cardholder's issuer (that is, the bank that issued the card) guarantees payment.

B

balance. Within IBM e-commerce, an attribute of a WebSphere Commerce Payments Batch object. Indicates whether the merchant and financial institution agreed on the contents of the batch when it was closed. See 84 for more details.

balanced. Within IBM e-commerce, an attribute of a WebSphere Commerce Payments Batch object. The batch has been successfully balanced. All totals agree.

balance status. Within IBM e-commerce, an attribute of a WebSphere Commerce Payments Batch object. The balance status of a batch can be balanced or out of balance.

baseline. In SETCo., a baseline product is the specific product within an operating system family that is run against the SET Tests. A vendor must designate a distinct baseline product for each unrelated operating system family. Refer to the *SET Testing Policies and Procedures* for a complete explanation.

batch. (1) In the credit card world, a batch is a collection of fund transfer requests that are all done at the same time (that is, *in a batch*). Individual fund transfers are not performed for each individual payment, but, rather, a group of transfers is processed so that both the merchant and the financial institution can agree on the funds that are to be transferred. Before a batch is *closed* (that is, the funds are exchanged) there is usually some type of reconciliation process so that both sides agree on the amounts. A group of records or data processing jobs brought together for processing or transmission. (2) Within IBM e-commerce, one of the fundamental WebSphere Commerce Payments objects is the Batch. A Batch is an object with which Payment and Credit objects are associated. Transfer of funds is to occur when the batch is closed. (3) A group of records or data processing jobs brought together for processing or transmission.

batch close date. One of two numeric search parameters that defines the chronological start of your search. Specify a date that precedes the batch close date for the batch you want to search.

batch number. The number that identifies the batch. WebSphere Commerce Payments assigns a number to the batch when the payment is deposited.

batch open date. One of two numeric search parameters that defines the chronological start of your search. Specify a date that precedes the batch open date for the batch you want to search.

batch number. The number that identifies the batch. The number WebSphere Commerce Payments assigns to the batch when the payment is deposited.

batch search. Search for a single batch or group of batches, based on a defined list of characteristics.

BCD. See binary-coded decimal notation.

big endian. A format for storage or transmission of binary data in which the most significant bit (or byte) is placed first. Contrast with little endian.

binary-coded decimal (BCD) notation. A binary-coded notation in which each of the decimal digits is represented by a binary numeral; for example, in binary-coded decimal notation that uses the weights 8, 4, 2, 1, the number “twenty-three” is represented by 0010 0011 (compare its representation 10111 in the pure binary numeration system). (I) (A)

bitmapped message. A variable-length transaction in which each bit in an array of bits indicates the presence or absence of a data field within the transaction.

brand. Within IBM e-commerce, the Cassette object for all of the WebSphere Commerce Payments cassettes (for example, Cassette for SET and Cassette for CyberCash). Each financial transaction for a WebSphere Commerce Payments cassette is associated with a particular brand (for example, MasterCard or VISA). Each account with a financial institution can be configured to support one or more brands.

browser. See Web browser.

browser plug-in. See Web browser plug-in.

C

CA. See certificate authority.

capture. (1) In SETCo., a transaction sent after the merchant has shipped the goods. This transaction will trigger the movement of funds from the Issuer to the Acquirer and then to the merchant account. (2) In the credit card world, payment is actually made when the funds are *captured*. All payments must be authorized *and* captured (although this work can be done at the same time). Note that all payments are associated with a batch (see “void payment” on page 85) and that the actual capture occurs when the associated batch is closed.

capture reversal. In SETCo., a transaction sent when the information in a previous capture message was incorrect or should never have been sent (such as when the goods were not actually shipped). If the capture reversal is the result of incorrect information, it will be followed by a new capture message with the correct information.

cardholder. In e-commerce, a person who has a valid payment card account and uses software that supports e-commerce.

cardholder application. In SETCo., a cardholder application, sometimes called a wallet, that is run by an online consumer that enables secure payment card transactions over a network. SET Cardholder applications must generate SET protocol messages that can be accepted by SET Merchant, Payment Gateway, and Certificate Authority components.

cascading. In high-availability cluster multiprocessing (HACMP), pertaining to a cluster configuration in which the cluster node with the highest priority for a particular resource acquires the resource if the primary node fails but relinquishes the resource to the primary node upon reintegration of the primary node into the cluster. Contrast with concurrent and rotating.

cassette. (1) In e-commerce, a software component consisting of a collection of Java classes and interfaces that can be easily installed into other software components involved in e-commerce to extend the function of these components. (2) In IBM e-commerce, a WebSphere Commerce Payments concept. The WebSphere Commerce Payments provides a framework that can support many different forms of payment. WebSphere Commerce Payments cassettes are written by IBM or third-party vendors to support different payment protocols (such as, SET and CyberCash) within the WebSphere Commerce Payments Framework. Thus, WebSphere Commerce Payments is an extensible product that can support additional protocols.

cast. In programming languages, an operator that converts the value of its operand to a specified type.

CERN. Conseil Européen pour la Recherche Nucléaire (European Laboratory for Particle Physics). Located in Geneva, Switzerland, CERN initiated the World Wide Web and was the first organization to create a Web server. The CERN Web server is the basis for many commercially available servers.

certificate. (1) In e-commerce, a digital document that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated. A certificate is issued by a certificate authority (CA). (2) In SETCo., a certificate that has been digitally signed by a trusted authority (usually the cardholder financial institution) to identify the user of the public key. SET defines the following certificate types:

- signature
- key encipherment
- certificate signature
- CRL signature

certificate authority (CA). (1) In e-commerce, an organization that issues certificates. The CA

authenticates the certificate owner's identity and the services that the owner is authorized to use, issues new certificates, renews existing certificates, and revokes certificates belonging to users who are no longer authorized to use them. See issuer. (2) In SETCo., certificate authority refers to both the component and to the entity issuing and verifying digital certificates. The component is a product run by a Certificate Authority entity that is authorized to issue and verify digital certificates as requested by Cardholder Wallet components, Merchant Server components, and/or Payment Gateway components over public and private networks.

certificate chain. (1) In SETCo., a hierarchy of digital certificates. The certificate at the top of the hierarchy is called the "root certificate." (2) In SETCo., an ordered grouping of digital certificates, including the root certificate, that are used to validate a specific certificate.

certificate renewal. In SETCo., the process by which a new certificate is created for an existing public key.

certificate revocation. In SETCo., the process of revoking an otherwise valid certificate by the entity that issued the certificate.

certificate revocation list. In SETCo., a list of certificate serial numbers previously issued by a certificate authority that indicate the certificates that are invalid prior to normal expiration due to compromise, disaffiliation, or some other unusual circumstance.

certification. In SETCo., the process of ascertaining that a set of requirements or criteria has been fulfilled and attesting to that fact to others, usually with some written instrument. A system that has been inspected and evaluated as fully compliant with the SET protocol by duly authorized parties and process would be said to have been certified.

certification authority. See certificate authority.

certified. In SETCo., the process of ascertaining that a set of requirements or criteria has been fulfilled and attesting to that fact to others, usually with some written instrument. A system that has been inspected and evaluated as fully compliant with the SET protocol by duly authorized parties and process would be said to have been certified.

CGI. See Common Gateway Interface.

CGI program. A computer program that runs on a Web server and uses the Common Gateway Interface (CGI) to perform tasks that are not usually done by a Web server (for example, database access and form processing). A CGI script is a CGI program that is written in a scripting language such as Perl.

CGI script. See CGI program.

clerk. (1) In IBM e-commerce, this is a WebSphere Commerce Payments concept. The WebSphere Commerce Payments has four different access rights. A clerk is defined on a per-merchant basis and has the lowest level of access. (2) A clerk is a low-level employee.

client. A computer system or process that requests a service of another computer system or process that is typically referred to as a server. Multiple clients may share access to a common server.

closed. An order moves into closed state when its associated payment, or payments, moves from deposited state into closed state (that is, when the batch associated with the payment closes). When an order is in closed state, the financial transaction is complete; monies are deposited, and the order cannot be modified. No commands are permitted for orders in this state.

cluster. In high-availability cluster multiprocessing (HACMP), a set of independent systems (called nodes) that are organized into a network for the purpose of sharing resources and communicating with each other.

cluster node. In high-availability cluster multiprocessing (HACMP), an RS/6000 system that participates in a cluster.

commerce service provider (CSP). An Internet service provider that hosts merchant shopping sites and processes payments for the merchants.

Common Gateway Interface (CGI). A standard for the exchange of information between a Web server and computer programs that are external to it. The external programs can be written in any programming language that is supported by the operating system on which the Web server is running. See CGI program.

concurrent. In high-availability cluster multiprocessing (HACMP), pertaining to a cluster configuration in which all cluster nodes use a resource simultaneously. A cluster node can fail and reintegrate into the cluster without affecting other cluster nodes or the resource. Contrast with cascading and rotating.

confidentiality. In SETCo., the protection of sensitive and personal information from unintentional and intentional attacks and disclosure.

constructor. In programming languages, a method that has the same name as a class and is used to create and initialize objects of that class.

constructor method. See constructor.

conversation. A logical connection between two transaction programs using an LU 6.2 session. Conversations are delimited by brackets to gain exclusive use of a session.

credit. In SETCo., a transaction sent when the merchant needs to return money to the cardholder (via the Acquirer and the Issuer) following a valid capture message, such as when goods have been returned or were defective.

credit reversal. In SETCo., a transaction sent when the information in a previous credit transaction was incorrect or should have never been sent.

cryptographic key. In SETCo., a value which is used to control a cryptographic process, such as encryption or authentication. Knowledge of an appropriate key allows correct decryption or validation of a message.

cryptography. (1) In SETCo., a mathematical process used for encryption or authentication information. (2) The discipline which embodies principles, means, and methods for the transformation of data in order to hide its information content, prevent its undetected modification and unauthorized use, or a combination thereof. (3) The transformation of data to conceal its contents and to prevent one person from forging or modifying another person's messages.

CSP. See commerce service provider.

CyberCash CashRegister. An electronic payment processing service that is provided by CyberCash, Inc. The CyberCash CashRegister enables merchants to accept and process various types of electronic payments for goods or services that are purchased over the Internet.

CyberCash cassette. A payment cassette that provides support for the CyberCash CashRegister.

D

daily batch totals. The Daily Batch Totals report computes the totals for all batches closed on the date specified on the Search window. The totals include all payments and credits made for all payment types.

decryption. In computer security, the process of transforming encoded text or ciphertext into plain text.

derived products. In SETCo., derived products are components that are created from a product that has received a SET Mark license. Derived products must be created from a product that has received the SET Mark, regardless of operating system family. Please refer to the *SET Testing Policies and Procedures* for a complete explanation.

deposit all . Selects all of the order payments displayed for deposit.

deposited amount . The amount deposited for a Payment. The deposited amount can be different than the approved amount.

deposit selected . Selects the order payments that you want to deposit.

digital envelope. (1) In SETCo., a cryptographic technique to encrypt data and send the encryption key along with the data. Generally, a symmetric algorithm is used to encrypt the data and an asymmetric algorithm is used to encrypt the encryption key. (2) In e-commerce, a package of encrypted data and the encryption key.

digital signature. (1) In SETCo., information encrypted with an entity private key, which is appended to a message to assure the recipient of the authenticity and integrity of the message. The digital signature proves that the message was signed by the entity owning, or having access to, the private key. (2) In e-commerce, data that is appended to, or is a cryptographic transformation of, a data unit and that enables the recipient of the data unit to verify the source and integrity of the unit and to recognize potential forgery.

distinguished name. In SET programs, information that uniquely identifies the owner of a certificate.

document type definition (DTD). The rules that specify the structure for a particular class of SGML or XML documents. The DTD defines the structure with elements, attributes, and notations, and it establishes constraints for how each element, attribute, and notation may be used within the particular class of documents. A DTD is analogous to a database schema in that the DTD completely describes the structure for a particular markup language.

DTD. See document type definition.

dual signature. In SETCo., a digital signature that covers two or more data structures by including secure hashes or each data structure in a single encrypted block. Dual signing is done for efficiency, that is, to reduce the number of public key encryption operations.

E

EAR file. An Enterprise Archive file represents a J2EE application that can be deployed in a WebSphere application server. EAR files are standard Java archive files and have the file extension .ear.

e-business. Either (a) the transaction of business over an electronic medium such as the Internet or (b) any organization (for example, commercial, industrial, nonprofit, educational, or governmental) that transacts its business over an electronic medium such as the Internet. An e-business combines the resources of traditional information systems with the vast reach of an electronic medium such as the Internet (including the World Wide Web, intranets, and extranets); it connects critical business systems directly to critical business constituencies--customers, employees, and suppliers. The key to becoming an e-business is building a

transaction-based Web site in which all core business processes (especially all processes that require a dynamic and interactive flow of information) are put online to improve service, cut costs, and sell products.

ECML. See Electronic Commerce Modeling Language.

e-commerce. (1) The exchange of goods and services for payment between the cardholder and merchant when some or all of the transaction is performed via electronic communication. (2) The subset of e-business that involves the exchange of money for goods or services purchased over an electronic medium such as the Internet.

electronic commerce. See e-commerce.

Electronic Commerce Modeling Language (ECML).

In e-commerce, a universal format for wallets that streamlines the collection of electronic data for shipping, billing, and payment on a merchant's Web site and thereby enhances the online shopping experience for consumers and merchants. IBM is one of many companies that are collaborating to develop ECML.

encryption. (1) In SETCo., the process of converting information in order to render it into a form unintelligible to all except holders of a specific cryptographic key. Use of encryption protects information between the encryption process and the decryption process (that is, the inverse of encryption), against unauthorized disclosure. (2) In computer security, the process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process.

event. In the Tivoli environment, any significant change in the state of a system resource, network resource, or network application. An event can be generated for a problem, for the resolution of a problem, or for the successful completion of a task. Examples of events are: the normal starting and stopping of a process, the abnormal termination of a process, and the malfunctioning of a server.

event listener. In IBM e-commerce, a computer program that waits to be informed of events of interest and acts upon them.

expiry. (1) The certificate expiration date assigned when the certificate was obtained. Certificates are specific to payment types (for example, SET or CyberCash.) (2) Specifies the card expiration date. An expiry value is required for SET protocol. The value is specified as a string and is used on the payment initiation message. For example, 199911 is an expiry value.

Extensible Markup Language (XML). A standard metalanguage for defining markup languages that was derived from and is a subset of SGML. XML omits the more complex and less-used parts of SGML and makes it much easier to (a) write applications to handle

document types, (b) author and manage structured information, and (c) transmit and share structured information across diverse computing systems. The use of XML does not require the robust applications and processing that is necessary for SGML. XML is being developed under the auspices of the World Wide Web Consortium (W3C).

F

failover. See fallover.

fallover. In high-availability cluster multiprocessing (HACMP), an active node's acquisition of resources that were previously owned by another cluster node in order to maintain the availability of those resources.

financial institution. (1) In SETCo., an establishment responsible for facilitating customer-initiated transactions or transmissions of funds for the extension of credit or the custody, loan, exchange, or issuance of money, such as a bank or its designate. (2) Within IBM e-commerce, banks, building societies, and credit unions are examples of financial institutions. An institution that provides financial services.

financial network. Within IBM e-commerce, the aggregate of card processors, acquirers, card issuers, and other institutions through which payment card transaction processing is traditionally performed.

firewall. In communication, a functional unit that protects and controls the connection of one network to other networks. The firewall (a) prevents unwanted or unauthorized communication traffic from entering the protected network and (b) allows only selected communication traffic to leave the protected network.

force. Within IBM e-commerce, a WebSphere Commerce Payments verb. An attempt to settle a batch (see 84). If the reconciliation step fails, the batch is still not closed on the WebSphere Commerce Payments (although it may be out of balance or not closed at the financial institution).

FQDN. See fully qualified domain name.

fully qualified domain name (FQDN). In the Internet suite of protocols, the name of a host system that includes all of the subnames of the domain name. An example of a fully qualified domain name is `mycomputer.city.company.com`. See host name.

H

HACMP. See high-availability cluster multiprocessing.

handle. In the AIX operating system, a data structure that is a temporary local identifier for an object. Allocating a handle creates it. Binding a handle makes it identify an object at a specific location.

hardware token. In SETCo., a portable device (for example, smart card, PCMCIA cards) specifically designed to store cryptographic information and possibly perform cryptographic functions in a secure manner.

has been certified. A system that has been inspected and evaluated as fully compliant with the SET protocol by duly authorized parties and process would be said to have been certified.

hash. See root key hash.

heartbeat. In software products, a signal that one entity sends to another to convey that it is still active.

high-availability cluster multiprocessing (HACMP). An application service that enables up to eight RS/6000 servers to access the same data in parallel. This optimizes application execution and scalability and protects against unplanned outages and server downtime.

home page. The initial Web page that is returned by a Web site when a user specifies the uniform resource locator (URL) for the Web site. For example, if a user specifies the URL for the IBM Web site, which is <http://www.ibm.com>, the Web page that is returned is the IBM home page. Essentially, the home page is the entry point for accessing the contents of the Web site. The home page may sometimes be called the "welcome page" or the "front page."

host. To provide the software and services for managing a Web site.

HostCapture/PostAuth. Within IBM e-commerce, this is a CyberCash concept. One of the three processing models supported by the CyberCash CashRegister service. In particular, the AcquirerProfile field of an account may be set to HostCapture/PostAuth = 2, which indicates that the acquirer controls batch processing and a separate deposit request is required to capture the funds after a payment is authorized.

host byte order. The byte order that a central processing unit (CPU) uses to store and process data. This byte order can be big endian or little endian, depending on the particular CPU. Contrast with network byte order.

host name. In the Internet suite of protocols, the name given to a computer. Sometimes, "host name" is used to mean fully qualified domain name; other times, it is used to mean the most specific subname of a fully qualified domain name. For example, if mycomputer.city.company.com is the fully qualified domain name, either of the following may be considered the host name:

- mycomputer.city.company.com
- mycomputer

HTML. See Hypertext Markup Language.

HTTP. See Hypertext Transfer Protocol.

Hypertext Markup Language (HTML). A markup language that conforms to the SGML standard and was designed primarily to support the online display of textual and graphical information that includes hypertext links.

Hypertext Transfer Protocol (HTTP). In the Internet suite of protocols, the protocol that is used to transfer and display hypertext documents.

idempotency. (1) A property of a mathematical operation whereby repeating the operation produces no change in the final result. For example, the operation of deducting \$25.00 from an account balance is not idempotent, but the operation of setting an account balance to \$500.00 is idempotent. (2) In SET Secure Electronic Transaction, a property that enables the sender of a request to repeat the request with a guarantee that the outcome will be the same regardless of whether the request is lost, the response is lost, or the request or response is delayed due to network problems. Idempotency is necessary because the SET protocol works in environments where message delivery is not guaranteed, and when the sender does not receive a response, it cannot determine the cause of the delay. If a SET application does not receive a response in a reasonable amount of time, it resends the message; when the receiving SET application determines that it has already processed that message, it retrieves the previous response and sends that response again.

instrumentation. In application or system software, either (a) monitoring functions that provide performance and other information to a management system or (b) the use of monitoring functions to provide performance and other information to a management system.

identify. To establish the identity.

installment payments. In SETCo., a type of payment transaction negotiated between the merchant and the cardholder which permits the merchant to process multiple authorizations. Cardholder specifies a maximum number of permitted Authorization for paying through installment payments.

integrity. In SETCo., the quality of information or a process that is free from error, whether induced accidentally or intentionally.

interactive. In SETCo., a generic class for a network transport mechanism that is dependent on a logical session being maintained during the message exchange (for example, World Wide Web sessions).

internet. (1) In SETCo., the largest collection of networks in the world, interconnected in such a way as

to allow them to function as a single virtual network. (2) A collection of interconnected networks that use the Internet suite of protocols. The internet that allows universal access is referred to as the Internet (with a capital "I"). An internet that provides restricted access (for example, to a particular enterprise or organization) is frequently called an intranet, whether or not it also connects to the public Internet.

Internet. The worldwide collection of interconnected networks that use the Internet suite of protocols and permit public access.

interoperability. In SETCo., the ability to exchange messages and keys, both manually and in an automated environment, with any other party implementing this standard, provided that both implementations use compatible options of this standard and compatible communications facilities.

interprocess communication (IPC). The process by which programs communicate data to each other and synchronize their activities. Semaphores, signals, and internal message queues are common methods of interprocess communication.

intranet. A private network that integrates Internet standards and applications (such as Web browsers) with an organization's existing computer networking infrastructure.

IP address. The unique 32-bit address that specifies the location of each device or workstation on the Internet. For example, 9.67.97.103 is an IP address.

IPC. See interprocess communication.

issuer. (1) In SETCo., the financial institution or its agent that issues the unique primary account number (PAN) to the cardholder for the payment card brand. (2) In e-commerce, a financial institution that issues payment cards to individuals. An issuer can act as its own certificate authority (CA) or can contract with a third party for the service.

J

J2EE. (Java 2 Enterprise Edition) J2EE is designed to support applications that implement enterprise services for customers, employees, suppliers, partners, and others who make demands on or contributions to the enterprise. This can be a single module or a group of modules packaged into an .ear file with a J2EE application deployment descriptor. J2EE applications are typically engineered to be distributed across multiple computing tiers.

Java. An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

Java Database Connectivity (JDBC). An application programming interface (API) that has the same characteristics as Open Database Connectivity (ODBC) but is specifically designed for use by Java database applications. Also, for databases that do not have a JDBC driver, JDBC includes a JDBC to ODBC bridge, which is a mechanism for converting JDBC to ODBC; it presents the JDBC API to Java database applications and converts this to ODBC. JDBC was developed by Sun Microsystems, Inc. and various partners and vendors.

Java Development Kit (JDK). A software package that can be used to write, compile, debug, and run Java applets and applications.

Java Runtime Environment (JRE). A subset of the Java Development Kit (JDK) that contains the core executables and files that constitute the standard Java platform. The JRE includes the Java Virtual Machine, core classes, and supporting files.

Java Virtual Machine (JVM). A software implementation of a central processing unit (CPU) that runs compiled Java code (applets and applications).

JDBC. See Java Database Connectivity.

JDK. See Java Development Kit.

JRE. See Java Runtime Environment.

JVM. See Java Virtual Machine.

K

keepalive message. In Internet communications, a message sent among nodes when no data traffic has been detected for a given period of time. This communication ensures the vitality of the session by keeping the link "alive."

key. In computer security, a sequence of symbols that is used with a cryptographic algorithm for encrypting or decrypting data. See private key and public key.

key pair. In computer security, a public key and a private key. When the key pair is used for encryption, the sender uses the public key to encrypt the message, and the recipient uses the private key to decrypt the message. When the key pair is used for signing, the signer uses the private key to encrypt a representation of the message, and the recipient uses the public key to decrypt the representation of the message for signature verification. See asymmetric and digital signature.

key ring. In computer security, a file that contains public keys, private keys, trusted roots, and certificates.

L

little endian. A format for storage or transmission of binary data in which the least significant bit (or byte) is placed first. Contrast with big endian.

M

memory leak. A condition in which a computer program allocates memory and does not free (or properly free) this memory. If the program continues to run and is not terminated, it uses large amounts of real memory and eventually runs out of memory.

merchant. In SETCo., a seller of goods, services, and/or other information who accepts payment for these items electronically. The merchant may also provide electronic selling services and/or electronic delivery of items for sale.

merchant chargeback. Within IBM e-commerce, when fraud occurs and a merchant is liable for funds not obtained, a financial institution may issue a merchant chargeback, reclaiming funds previously credited to a merchant's account.

merchant server. (1) In SETCo., a Merchant Server component is a product run by an online merchant to process payment card transactions and authorizations. It communicates with the Cardholder Wallet, Payment Gateway, and Certificate Authority components. (2) In e-commerce, a Web server that offers cataloged shopping.

merchant settings. The settings that a merchant has made for a cassette. In the WebSphere Commerce Payments user interface, the Payment System object displays as Merchant Settings.

MIME. See Multipurpose Internet Mail Extensions.

mirroring. In the AIX operating system, the maintenance of more than one copy of stored data to prevent the loss of data.

Multipurpose Internet Mail Extensions (MIME). An Internet standard for identifying the type of object being transferred across the Internet. MIME types include several variants of audio, graphics, and video.

mutex. A mutual exclusion lock. See mutual exclusion mechanism.

mutual exclusion mechanism. In software, a method for preventing two separately executing pieces of code from interfering with each other's use of a particular data object. For example, if one thread is executing a function that modifies a shared data structure, the application may need to prevent other threads from simultaneously attempting to read the data before the modifications are complete.

N

network. In SETCo., a collection of communication and information processing systems that may be shared among several users.

network byte order. The byte order that a network uses to transmit data. In the Internet, this byte order is always big endian. Contrast with host byte order.

node. See cluster node.

normal mode. In the IBM Payment Gateway, the processing scheme in which a batched SET message is presented in its entirety to the customized exits of the Payment Gateway Application. Contrast with supervisor mode.

number of credits. In SETCo., a credit is a transaction sent when the merchant needs to return money to the cardholder (via the Acquirer and the Issuer) following a valid capture message, such as when goods have been returned or were defective. Credits can be for up to the total amount of all payments associated with an Order. There can be zero or more Credits per Order.

number of payments. In SETCo., a payment is a request by the merchant to the financial institution to approve all or part of an order. In many cases, all the money authorized for collection by the order will be collected in a single payment. Some payment systems may allow the money authorized in one order (that is, one set of payment instructions) to be collected in multiple payments, depending on the business model. There can be zero or more payments per order.

O

ODBC. See Open Database Connectivity.

ODBC bridge. See Java Database Connectivity.

Open Database Connectivity (ODBC). A standard application programming interface (API) for accessing data in both relational and nonrelational database management systems. Using this API, database applications can access data stored in database management systems on a variety of computers even if each database management system uses a different data storage format and programming interface. ODBC is based on the call level interface (CLI) specification of the X/Open SQL Access Group and was developed by Digital Equipment Corporation (DEC), Lotus, Microsoft, and Sybase. Contrast with Java Database Connectivity.

online catalog. In SETCo., shopping on the Internet is simple with online catalogs. Online catalogs are Web pages that display items for sale by an online merchant.

order. An order represents all the instructions and information needed from the consumer (payer) in order for the merchant (payee) to collect money.

order amount. The amount of the order.

order fulfillment. Within IBM e-commerce, merchant systems responsible for shipping or distributing orders for which payment has been received. It is believed that an order fulfillment system would query the WebSphere Commerce Payments to determine what goods are to be shipped.

order search. Search for a single order or group of orders, based on a defined set of characteristics.

out of balance. An unsuccessful attempt was made to balance a batch. All totals do not agree.

P

password. For computer or network security, a specific string of characters entered by a user and authenticated by the system in determining the user's privileges, if any, to access and manipulate the data and operations of the system.

payment. In SETCo., a payment is a request by the merchant to the financial institution to approve all or part of an order. In many cases, all the money authorized for collection by the order will be collected in a single payment. Some payment systems may allow the money authorized in one order (that is, one set of payment instructions) to be collected in multiple payments, depending on the business model.

payment amount. The total payment amount deposited by the merchant for this order.

payment card. (1) In SETCo., a term used to collectively refer to credit cards, debit cards, charge cards, and bank cards issued by a financial institution and which reflects a relationship between the cardholder and the financial institution. (2) In e-commerce, a credit card, debit card, or charge card (a) that is issued by a financial institution and shows a relationship between the cardholder and the financial institution and (b) for which a certificate can be issued from an authenticated certificate authority.

payment cassette. A cassette that implements an electronic payment protocol.

payment gateway. (1) In SETCo., a payment gateway component is a product run by an acquirer or a designated third party that processes merchant authorization and payment messages (including payment instructions from cardholders) and interfaces with private financial networks. (2) In e-commerce, the entity that handles transactions between a merchant and an acquirer.

Payment Gateway Application. In the Payment Gateway Transaction Manager (PGTM), the component that processes SET transactions.

Payment Gateway Transaction Manager (PGTM). In the IBM Payment Gateway, the component that is the application-level routing switch. It provides the protocol-level conversion for managing incoming and outgoing communication, and it provides base services for the intelligent routing of transactions to applications. It manages the communication with merchants and routes transactions among merchants, the Payment Gateway Application, and the acquirer's private financial networks.

payment number. (1) Numeric token. (2) A unique identifier for a particular payment within an order.

payment server. In e-commerce, the electronic equivalent of a cash register that organizes and accepts payment for the goods and services selected for purchase. A payment server uses other components, such as a payment gateway and a payment management system, to complete the financial transactions.

Payment Suite. The brand name for IBM's family of payment products for e-commerce.

PGTM. See Payment Gateway Transaction Manager.

port. In the Internet suite of protocols, a specific logical connector between the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) and a higher-level protocol or application. See well-known port.

port number. In the Internet suite of protocols, the identifier for a logical connector between an application entity and the transport service.

primary account number (PAN). In SETCo., the assigned number that identifies the card issuer and cardholder. This account number is composed of an issuer identification number, an individual account number identification, and an accompanying check digit, as defined by ISO 7812-1985.

private key. (1) In SETCo., a cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity and not made public. This key is used to create digital signatures or to decrypt messages or files. (2) In computer security, a key that is known only to its owner. Contrast with public key. See public key cryptography.

protocol. The meanings of, and the sequencing rules for, requests and responses used for managing a network, transferring data, and synchronizing the states of network components.

public key. (1) In SETCo., a cryptographic key used with a public key cryptographic algorithm, uniquely

associated with an entity publicly available. It is used to verify signatures that were created with the matched private key. Public keys are also used to encrypt messages or files that can only be decrypted using the matched private key. (2) In computer security, a key that is made available to everyone. Contrast with private key. See public key cryptography.

public key cryptography. In computer security, cryptography in which public keys and private keys are used for encryption and decryption.

purge. Within IBM e-commerce, a WebSphere Commerce Payments verb. To remove all associated Payments and Credits from a Batch object, treating it as if it has just been created.

R

random. In SETCo., a value in a set that has equal probability of being selected from the total population of possibilities and is, hence, unpredictable.

realm. In the WebSphere family of products, a database of users, groups, and access control lists. A user must be defined in a realm to access any resource belonging to that realm.

recurring payments. In SETCo., a type of payment transaction initiated by the cardholder that permits the merchant to process multiple authorizations. There are two kinds of recurring payments:

1. multiple payments for a fixed amount
2. repeated billings

refund. Identifies the Credit amount in the smallest denomination of the particular currency used to place the Order.

registration authority. In SETCo., an independent third-party organization that processes payment card applications for multiple payment card brands and forwards applications to the appropriate financial institutions.

reintegration. In high-availability cluster multiprocessing (HACMP), the actions that occur within the cluster when a component that had previously detached from the cluster returns to the cluster. These actions are controlled by the event scripts and when necessary, by manual intervention.

root certificate. In SETCo., the certificate at the top of the certificate hierarchy. See certificate chain.

root key hash. In SET programs, a hexadecimal value that is used to verify the validity of a root certificate. The hash value is published for a consumer to use when the software does not recognize the root certificate.

rotating. In high-availability cluster multiprocessing (HACMP), pertaining to a cluster configuration in which

the cluster node with the highest priority for a particular resource acquires the resource if the primary node fails and retains the resource even upon reintegration of the primary node into the cluster. Contrast with cascading and concurrent.

RS/6000. A family of workstations and servers based on IBM's POWER architecture. They are primarily designed for running multi-user numerical computing applications that use the AIX operating system.

S

sale. (1) In the credit card world, a sale occurs when a transaction is authorized and marked for capture all at once rather than using a two-step process. (2) Within IBM e-commerce, a WebSphere Commerce Payments user interface verb. It means a simultaneous Approve and Deposit.

sale all. Selects all orders displayed to approve and move the associated payment directly into deposited state. The sale function automatically performs an approve and a deposit on your payment.

sale selected. Selects the orders that you want to approve and move the associated payment directly into deposited state. The sale function automatically performs an approve and a deposit on your payment.

sales transaction. In SETCo., a payment authorization transaction that allows a merchant to authorize a transaction and request payment in a single message to the acquirer.

Secure Electronic Transaction. See SET Secure Electronic Transaction.

Secure Sockets Layer (SSL). (1) In SETCo., Secure Socket Layer (SSL) (developed by Netscape Communications Company) is a standard that encrypts data between a Web browser and a Web server. SSL does not specify what data is sent or encrypted. In an SSL session, all data sent is encrypted. (2) A security protocol that provides communication privacy. SSL enables client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. SSL was developed by Netscape Communications Corp. and RSA Data Security, Inc.

server. In SETCo., a functional unit that provides services to one or more clients over a network. Examples include a file server, a print server, and a mail server.

servlet. An application program, written in the Java programming language, that is executed on a Web server. A reference to a servlet appears in the markup for a Web page, in the same way that a reference to a graphics file appears. The Web server executes the

servlet and sends the results of the execution (if there are any) to the Web browser. Contrast with applet.

SET. See SET Secure Electronic Transaction.

SET logo. In SETCo., the SET logo or SET Mark is your assurance that the merchant is using software that has successfully completed the SET Software Certification test.

SET cassette. A payment cassette that provides support for the SET protocol.

SET Secure Electronic Transaction. (1) In SETCo., the SET Secure Electronic Transaction™ protocol is an open industry standard developed for the secure transmission of payment information over the Internet and other electronic networks. (2) A specification for securing payment card transactions over open networks such as the Internet. SET was developed by Visa, MasterCard, IBM, and other technology companies.

settle. Within IBM e-commerce, a WebSphere Commerce Payments verb. An attempt to close a Batch object and transfer funds. As part of the settling procedure, there may be some reconciliation or balancing steps (depending on the cassette and financial institution policy) to ensure that the merchant and financial institution agree on the funds being transferred. If the reconciliation step fails, the batch may remain in an open state.

settle all. Settles all batches displayed. The batches can then submit payments and refunds for processing by a payment processor.

settle batches. Settle batches is used to submit batches (payments and refunds) for processing by a payment processor. You can choose to settle one Batch, or multiple Batches.

settle selected. Settles the batches you selected. The selected batches can then submit payments and refunds for processing by a payment processor.

sibling. In SETCo., sibling products are components which, by virtue of being within the same operating system family, are closely related to baseline products. Siblings must be of the same operating system family as the baseline product from which they were created, with identical functionality. Refer to the *SET Testing Policies and Procedures* for a complete explanation.

SMIT. See System Management Interface Tool.

socket. An endpoint provided by the transport service of a network for communication between processes or application programs.

socks-enabled. Pertaining to TCP/IP software, or to a specific TCP/IP application, that understands the socks protocol. "Socksified" is a slang term for socks-enabled.

socksified. See socks-enabled.

socks protocol. A protocol that enables an application in a secure network to communicate through a firewall via a socks server.

socks port. The port on which the Socks server is listening.

socks server. A circuit-level gateway that provides a secure one-way connection through a firewall to server applications in a nonsecure network.

SSL. See Secure Sockets Layer.

stack. A slang term for the set of protocols that comprise TCP/IP. The preferred term is TCP/IP.

supervisor. Can perform all payment processing functions for the merchant.

supervisor mode. In the IBM Payment Gateway, the processing scheme in which a batched SET message is presented as a series of individual requests to the customized exits of the Payment Gateway Application. Contrast with normal mode.

System Management Interface Tool (SMIT). An interface tool of the AIX operating system for installing, maintaining, configuring, and diagnosing tasks.

T

TEC. See Tivoli Enterprise Console.

terminal capture. Within IBM e-commerce, a CyberCash concept. One of the three processing models supported by the CyberCash CashRegister service. In particular, the AcquirerProfile field of an account may be set to Terminal Capture = 3, which indicates that the merchant controls batch processing.

thread. A stream of computer instructions that is in control of a process. A multi-threaded process begins with one stream of instructions (one thread) and may later create other instruction streams to perform tasks.

thread pool. The threads that are being used by or are available to a computer program.

time approved. The date and time that this Payment entry was created.

time opened. The time that the batch was created.

time ordered. The time that the order entry was created.

Tivoli Enterprise Console (TEC). A Tivoli product that collects, processes, and automatically initiates corrective actions for system, application, network, and database events; it is the central control point for events from all sources. The Tivoli Enterprise Console provides a

centralized, global view of the network computing environment; it uses distributed event monitors to collect information, a central event server to process information, and distributed event consoles to present information to system administrators.

Tivoli GEM. See Tivoli Global Enterprise Manager.

Tivoli Global Enterprise Manager (Tivoli GEM). A Tivoli product that allows system administrators to graphically monitor, control, and configure applications residing in distributed and host (S/390) environments and to use the concept of business systems management to organize related components, thereby providing a business perspective for management decisions. Tivoli Global Enterprise Manager gives information technology staff a logical view of the computing environment; this view shows, at a glance, the status of the multiple applications that comprise the enterprise's business system, including application components, the relationships among and between components, and the flow of data between the applications. By providing this view from a business perspective, Tivoli Global Enterprise Manager enables system administrators to quickly make determinations about the business impact of any component failure. Addressing technology problems from the business perspective greatly improves the effectiveness of system administrators and provides a higher level of service to users.

Tivoli Inventory. A Tivoli product that enables system administrators to gather hardware and software information for a network computing environment. It scans the managed resources and stores inventory information in the configuration repository.

Tivoli management software. The overall descriptor for software from Tivoli Systems Inc., which includes Tivoli Enterprise software (for systems management in a large organization), Tivoli IT Director (for systems management in a small or medium organization), and Tivoli Cross-Site (for the management of e-commerce systems). Tivoli management software enables organizations to centrally manage their computing resources (including the critical applications that drive business performance and profits) in a simple and straightforward manner.

Tivoli Ready. Pertaining to a product that has passed rigorous product certification testing by Tivoli Systems Inc. to ensure that the product delivers turnkey (or "out-of-the-box") integration with Tivoli management software. A product that has passed this certification testing carries the Tivoli Ready logo.

transaction. In SETCo., a sequence of one or more related messages.

trust chain. In SETCo., a synonym for certificate chain. See 76.

trusted root. In the Secure Sockets Layer (SSL), the public key and associated distinguished name of a certificate authority (CA).

U

uniform resource locator (URL). (1) A sequence of characters that represent information resources on a computer or in a network such as the Internet. This sequence of characters includes (a) the abbreviated name of the protocol used to access the information resource and (b) the information used by the protocol to locate the information resource. For example, in the context of the Internet, these are abbreviated names of some protocols used to access various information resources: http, ftp, gopher, telnet, and news; and this is the URL for the IBM home page:

http://www.ibm.com . (2) The address of an item on the World Wide Web. It includes the protocol followed by the fully qualified domain name (sometimes called the host name) and the request. The Web server typically maps the request portion of the URL to a path and file name. For example, if the URL is http://www.networking.ibm.com/nsg/nsgmain.htm, the protocol is http; the fully qualified domain name is www.networking.ibm.com; and the request is /nsg/nsgmain.htm.

URL. See uniform resource locator.

user exit routine. A user-written routine that receives control at predefined user exit points. User exit routines can be written in assembler or a high-level language.

V

virtual sales slip. In SETCo., detailed information on a financial transaction which is generated by the merchant's online store and downloaded to your digital wallet. Typical items contained in the virtual sales slip are confirmation of your order, shipping details, tax (if applicable), and total amount of sale.

virtual store. An interactive simulation of a store on the World Wide Web.

void payment. Within IBM e-commerce, a verb meaning to nullify or cancel a payment operation (that is, to make it as if it never happened).

W

wallet. In the IBM Payment Suite, software that enables a user to make approved payments to authenticated merchants over public networks and to manage payment card accounts and purchases.

WAR file. A Web Archive (WAR) file is a Java archive file used to store one or more of the following: servlets; JavaServer Pages (JSP) files; utility classes; static

documents (such as HTML files, images and sound); client-side applets, beans and classes; descriptive meta-information. Its standard file extension is .war. WAR files are used to package Web modules.

Web. See World Wide Web.

Web browser. (1) Within IBM e-commerce, software running on the cardholder processing system that provides interface to public data networks. (2) A client program that initiates requests to a Web server and displays the information that the server returns.

Web browser plug-in. In SETCo., software installed on the cardholder's computer used to add functions to the Web browser.

webmaster. The person who is ultimately responsible for managing and maintaining a particular Web site.

Web page. Any document that can be accessed by a uniform resource locator (URL) on the World Wide Web. Contrast with home page.

Web server. A server that is connected to the Internet and is dedicated to serving Web pages.

Web site. A Web server that is managed by a single entity (an organization or an individual) and contains information in hypertext for its users, often including hypertext links to other Web sites. Each Web site has a home page. In a uniform resource locator (URL), the Web site is indicated by the fully qualified domain name. For example, in the URL `http://www.networking.ibm.com/nsg/nsgmain.htm`, the Web site is indicated by `www.networking.ibm.com`, which is the fully qualified domain name.

WebSphere. Pertaining to a family of IBM software products that provide a development and deployment environment for basic Web publishing and for transaction-intensive, enterprise-scale e-business applications.

well-known port. In the Internet suite of protocols, one of a set of preassigned protocol port numbers that address specific functions used by transport-level protocols such as the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). The File Transfer Protocol (FTP) and the Simple Mail Transfer Protocol (SMTP), for example, use well-known port numbers.

World Wide Web (WWW). A network of servers that contain programs and files. Many of the files contain hypertext links to other documents available through the network.

WWW. See World Wide Web.

X

XML. See Extensible Markup Language.

Numerics

2KP transaction. A SET transaction in which the cardholder messages are unsigned and two key pairs (one for the merchant and one for the payment gateway) are used for encryption.

3KP transaction. A SET transaction in which the cardholder messages are unsigned and three key pairs (one for the merchant, one for the payment gateway, and one for the cardholder) are used for encryption.

Index

A

- AcceptPayment 35
- account
 - creating 22
- Account 50
- AccountAdmin object 8
- acquirer
 - definition v
- administration objects 8
 - account 8
 - cassette 8
 - PaySystemAdmin 8
- Approve 37
- ApproveReversal 37
- Authorize
 - definition vi

B

- Batch 48
 - definition vii
- Batch Object
 - XML example 48
- Batch objects 9
- Batch totals
 - Viewing 29
- Batch Totals report, Daily 30
- batch, delete 39
- Batches
 - Settling 27
- BatchOpen 37
- BatchPurge 37

C

- Capture
 - definition vii
- card processor
 - definition v
- cardholder
 - definition v
- Cassette for VisaNet
 - installation 13
 - SOCKS server 20
 - tutorial 19
 - uninstalling on Windows NT 15
- Cassette properties
 - accountadmin 50
 - batch 48
 - Credit 47
 - order 43
 - paysystemadmin 51
 - VisaNet payment 45
- Cassette-specific parameters
 - Approve 37
 - Refund 42
- CassetteAdmin 49

- CassetteAdmin object 8
- CassetteControl 38
- closeorder
 - delete option 38
- communication parameters 9
- configuration steps 20
- create a merchant
 - tutorial 21
- CreateAccount 38
- Credit 47
 - definition vii
- Credit Object
 - XML example 47
- Credit objects 9
- Credits
 - Issuing 28

D

- Daily Batch Totals report 30
- delete batch 39
- delete option, closeorder 38
- deposit 39
- Deposit function 30
- depositreversal 40
- document, PSApiResult 43

F

- financial network
 - definition vi
- Framework object
 - extensions 49
- function, Deposit 30
- function, Sale 30

G

- gateway
 - definition vi

I

- IBM Universal Database
 - web site viii
- IBM WebSphere Application Server
 - web site viii
- issuer vi
- Issuing credits 28

K

- keywords, Optional
 - acceptpayment 35
 - createaccount 39
 - deposit 39
 - modifyaccount 40

keywords, Optional *(continued)*
 modifycassette 41
 modifypaysystem 41
keywords, Required
 acceptpayment 35
 createaccount 38

L

leased line
 definition vi

M

Managing payment processing 23
MaxRetries 49
merchant
 definition vi
merchant server
 definition vi
model, object 43
 implementation 8
models, object 5
ModifyAccount 40
ModifyCassette 41

N

Notices 71

O

object model 43
 implementation 8
object models 5
object, account 8
Object, Batch
 XML example 48
object, cassette 8
Object, Credit
 XML example 47
object, Framework
 extensions 49
object, Order
 XML example 43
Object, Payment
 XML example 46
objects, administration 8
 account 8
 cassette 8
 PaySystemAdmin 8
objects, Batch 9
objects, Credit 9
objects, Order 9
objects, payment
 batch 9
 credit 9
 order 9
 payment 9
objects, Payment 9

Optional keywords
 acceptpayment 35
 createaccount 39
 deposit 39
 modifyaccount 40
 modifycassette 41
 modifypaysystem 41
Order object
 XML example 43
Order objects 9
Orders
 Approving
 Sale function 25
 Order details 25

P

parameters, communication 9
parameters, protocol 35
Payment Object
 XML example 46
payment objects
 batch 9
 credit 9
 order 9
 payment 9
Payment objects 9
payment processing
 implementation 5
Payment processing
 managing 23
payment processing tasks
 tutorial 19
payment server
 definition vi
PaySystemAdmin 52
properties, Cassette
 accountadmin 50
 batch 48
 Credit 47
 order 43
 paysystemadmin 51
 VisaNet payment 45
protocol parameters 35
PSApiResult document 43
Purchase example 6
Purchasing cards 5

Q

query call 43

R

ReadTimeout 49
ReceivePayment 42
Refund 42
RefundReversal 42
report, Daily Batch Totals 30
Required keywords
 acceptpayment 35

Required keywords *(continued)*

 createaccount 38
Retry 49
reversal, deposit 40

S

Sale function 30
 Approving orders using 25
Settle 28
Settling Batches 27
starting the WebSphere Commerce Payments 20
steps, configuration 20

T

tasks, payment processing
 tutorial 19
technical information, WebSphere Commerce Payments
 web site viii
Totals report, Daily Batch 30
trademarks 72

U

Universal Database, IBM
 web site viii

V

Viewing batch totals 29
VisaNet Order 43
VisaNet Payment 45
VisaNet, Cassette for
 SOCKS server 20
 tutorial 19

W

Web sites
 IBM WebSphere Commerce payment
 processing viii
WebSphere Application Server, IBM
 web site viii
WebSphere Commerce Payments technical information
 web site viii
WebSphere Commerce Payments Web site 13

Readers' Comments — We'd Like to Hear from You

IBM WebSphere Commerce Payments for Multiplatforms
Cassette for VisaNet Supplement
Version 3.1

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Software Reengineering
Department G71A / Bldg 503
P.O. Box 12195
Research Triangle Park, NC
27709-9990



Fold and Tape

Please do not staple

Fold and Tape



Printed in U.S.A.