

IBM WebSphere Commerce



Payments Cassette for VisaNet Supplement

Version 5.5

IBM WebSphere Commerce



Payments Cassette for VisaNet Supplement

Version 5.5

Note

Before using this information and the product it supports, be sure to read the general information under Appendix C, "Notices", on page 81.

Fifth Edition (June 2003)

This edition applies to version 5.5 of IBM WebSphere Commerce Payments and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2000, 2003. All rights reserved.

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

Contents

About this book	v
Conventions used in this book	v
Terminology used in this book	vi
Additional information	vii
Using the online help	vii
Locating the printable documentation	vii
Viewing the WebSphere Commerce Web site for product information	vii
Other WebSphere Commerce Payments documents and Web sites	viii
VisaNet-related documents	viii

Chapter 1. Overview of VisaNet.	1
Vital Processing Services	2
FHMS	3
Communication methods	3
VisaNet merchant registration	4
Registering when using Vital Processing Services	4
Registering when using FHMS	4
American Express CID support	5
Merchant identification	5
Cassette features	5
Address Verification Service	5
Purchasing cards	6
Sensitive data protection	7
WebSphere Commerce Payments roles	8

Chapter 2. VisaNet and WebSphere Commerce Payments concepts	11
A VisaNet purchase example	11
WebSphere Commerce Payments object model implementation	13
Administration objects	13
Financial objects	14
Cassette-specific characteristics and behaviors	14
Retry parameters	14
Cassette for VisaNet payment command summary	15
Summary of state changes	16

Chapter 3. Before you start	19
--	-----------

Chapter 4. Tutorial	21
Step 1: Accessing the WebSphere Commerce Payments user interface	22
Step 2: Configuring the cassette to communicate with external sites	22
Step 3: Creating a WebSphere Commerce Payments merchant and authorizing a cassette	24
iSeries support of SSL for the Cassette for VisaNet	24
Step 4: Defining WebSphere Commerce Payments users	25
Step 5: Assigning user roles	25

Logging in as the Merchant Administrator	26
Step 6: Configuring the merchant cassette settings	26
Step 7: Creating an account	27
Step 8: Creating orders using the Sample Checkout	28
Step 9: Approving orders with the Sale function	30
Approving orders from the Order page	31
Approving multiple orders at one time	32
Separate approvals and deposits	33
Step 10: Settling batches	34
Step 11: Issuing a credit	35
Step 12: Viewing batch totals	36

Chapter 5. Cassette for VisaNet Cashier profiles	39
---	-----------

Chapter 6. Command reference	41
Cassette for VisaNet commands	41
AcceptPayment	41
Approve	43
ApproveReversal	44
BatchOpen	44
BatchPurge	44
BatchClose	44
CassetteControl	44
CloseOrder	45
CreateAccount	45
CreatePaySystem	45
DeleteBatch	46
Deposit	46
DepositReversal	47
ModifyAccount	47
ModifyCassette	48
ModifyPaySystem	49
ReceivePayment	50
Refund	50
RefundReversal	50

Chapter 7. Object reference	51
Financial objects used by Cassette for VisaNet	51
VisaNet Order	51
VisaNet Payment	53
VisaNet Credit	55
VisaNet Batch	56
Administrative objects used by Cassette for VisaNet	57
CassetteAdmin	57
AccountAdmin	58
PaySystemAdmin	59
Address Verification Service (AVS) result codes	61

Appendix A. Cassette for VisaNet return codes	63
--	-----------

Appendix B. Cassette for VisaNet messages	71
--	-----------

Appendix C. Notices 81
Trademarks 82

Glossary 83

Index 91

About this book

This book is for users and administrators of the Cassette for VisaNet who are responsible for implementing the cassette in an IBM® WebSphere® Commerce Payments environment. This information will help you to understand the concepts behind the cassette and what you need to use the cassette. Programmers who are responsible for developing applications to manage WebSphere Commerce Payments may find the cassette-specific information provided here useful. Reference information about API commands, parameter requirements, and objects is provided, and XML examples showing how objects are used are included.


This book serves as a supplement to the *WebSphere Commerce Administration Guide* and *WebSphere Commerce Installation Guide*. For information about how to install and configure payment cassettes, refer to those documents.

Note: IBM WebSphere Commerce Payments (hereafter called WebSphere Commerce Payments) was previously known as Payment Manager. 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 used in this book


This book uses the following highlighting conventions:

- **Boldface** type indicates commands or graphical user interface (GUI) controls such as names of fields, icons, or menu choices.
- Monospace type indicates examples of text you enter exactly as shown, file names, and directory paths and names.
- *Italic* type is used to emphasize words. Italics also indicate names for which you must substitute the appropriate values for your system. When you see the following names, substitute your system value as described.

 indicates information specific to the Windows® operating environment.

 indicates information specific to AIX®.

 indicates information specific to the Solaris Operating Environment.

 indicates information specific to the IBM iSeries™ 400 (formerly called AS/400®).

 indicates information specific to Linux.


References to *Linux* apply to both Linux on Intel® workstations and also to Linux on IBM eServer iSeries, pSeries™, zSeries™ and S/390® unless otherwise specified.

WC_installdir represents the following default installation paths for WebSphere Commerce:

 /usr/lpp/WebSphere/CommerceServernn

  /opt/WebSphere/CommerceServernn

 *drive:*\WebSphere\CommerceServernn

 /QIBM/ProdData/CommerceServernn

Payments_installdir represents the following default installation paths for WebSphere Commerce Payments:

 /usr/lpp/WebSphere/CommerceServernn/payments

  /opt/WebSphere/CommerceServernn/payments

 *drive:*\WebSphere\CommerceServernn\payments

 /QIBM/ProdData/CommercePayments/Vnn

Terminology used in this book

This book may use some terms that are unfamiliar to you, such as *payment cassette*, *merchant server*, and *payment gateway*. Refer to the glossary provided in this document for a definition of terms used in this book and in other WebSphere Commerce Payments documentation. Terms are also described in the WebSphere Commerce online help.

The following terms used in WebSphere Commerce Payments documents have similarities to other terms used in WebSphere Commerce online help and publications:

Store and merchant

In WebSphere Commerce, the term *store* is used to refer to an *online store*. An online store uses Internet technologies to sell or exchange goods or services. In WebSphere Commerce Payments, a store is equivalent to a *merchant*. For example, when you see a reference in this document to merchant settings or adding merchants, think of it as store settings or adding stores.

Site Administrator and Payments Administrator

A *Site Administrator* is a defined role in WebSphere Commerce that installs, configures, and maintains WebSphere Commerce and the associated software and hardware. This role typically controls access and authorization and has the most authority when performing administrative tasks.

Similarly, in the Payments component of WebSphere Commerce, the *Payments Administrator* has the most authority when performing Payment functions. Although the Site Administrator can perform Payments Administrator tasks, the Payments Administrator cannot perform all Site Administrator tasks.

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

More information about WebSphere Commerce and the Payments component is available from a variety of sources in different formats. The following are sources of WebSphere Commerce information:

- Online help
- Portable document format (PDF) files
- Web sites

Using the online help

The WebSphere Commerce online information provides information about customizing, administering, and reconfiguring WebSphere Commerce.

The WebSphere Commerce Payments online help provides information about how to use the graphical user interfaces associated with the Payments component. The Payments online help is available by clicking the question mark icon in the upper right corner of the user interface panel.

Locating the printable documentation

Some of the WebSphere Commerce online information is also available on your system in PDF files, which you can view and print using Adobe Acrobat Reader. In addition, WebSphere Commerce Payments documents are provided as PDF files. You can download the Acrobat Reader for free from the Adobe Web site at the following Web address:

<http://www.adobe.com>

PDF files can be accessed through the WebSphere Commerce online help and through the WebSphere Commerce Web site for product information.

Viewing the WebSphere Commerce Web site for product information

WebSphere Commerce product information is available at the WebSphere Commerce technical library Web site:

<http://www.ibm.com/software/commerce/wscm/library/lit-tech.html>.

A copy of this book, and any updated versions of this book, are available as PDF files from the Web site.

Other WebSphere Commerce Payments documents and Web sites

The following documents provide information related to the Payments component of WebSphere Commerce:

- The *WebSphere Commerce Installation Guide* provides instructions on how to install and configure WebSphere Commerce Payments for your platform.
- The *WebSphere Commerce Administration Guide* contains conceptual information and shows how to configure WebSphere Commerce Payments using the Configuration Manager user interface.

This document supplements these books. Additional cassette supplements may be available for other types of payment cassettes. All documents are provided in Portable Document Format (PDF).

Visit the following Web sites for more information about WebSphere Commerce Payments:

- <http://www.ibm.com/software/webservers/commerce/payment/> provides more information on the WebSphere Commerce payment-processing software, including information about the payment cassettes that are available for use with IBM WebSphere Commerce Payments.
- <http://www.ibm.com/software/webservers/commerce/payments/support.html> provides current WebSphere Commerce Payments technical information and links to the latest WebSphere Commerce Payments documentation.
- <http://www.ibm.com/software/webservers/commerce/payment/paymentcards.html> provides information about WebSphere Commerce Payments cassette development.

WebSphere Commerce support and download information is available at the following Web sites:

- <http://www.ibm.com/software/commerce/wscom/support/index.html>
- <http://www.ibm.com/software/commerce/wscom/downloads/index.html>

VisaNet-related documents

This supplement describes how the Cassette for VisaNet works with two different processing networks: Vital Processing Services and First Horizon (also known as First Horizon Merchant Services or FHMS).

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

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

FHMS follows the same data formats and protocols as those listed in the Vital documents. For information about FHMS link-level protocols, contact FHMS. For more information about FHMS, see <http://www.fhmsi.com>.

Chapter 1. Overview of 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 cannot 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 high-capacity transaction networks for payment processing.

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.

IBM WebSphere Commerce supports Internet commerce with VisaNet through a payment cassette called the 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 interprets generic payment transaction messages into specific payment-protocol messages that are sent to the appropriate payment gateway for further processing through a financial network. The Cassette for VisaNet supports the processing of payment transactions through these financial networks:

- Vital Processing Services (also called Vital)
- First Horizon Merchant Services (FHMS)

Figure 1 on page 2 shows the major components involved with the IBM Cassette for VisaNet in an e-commerce environment.

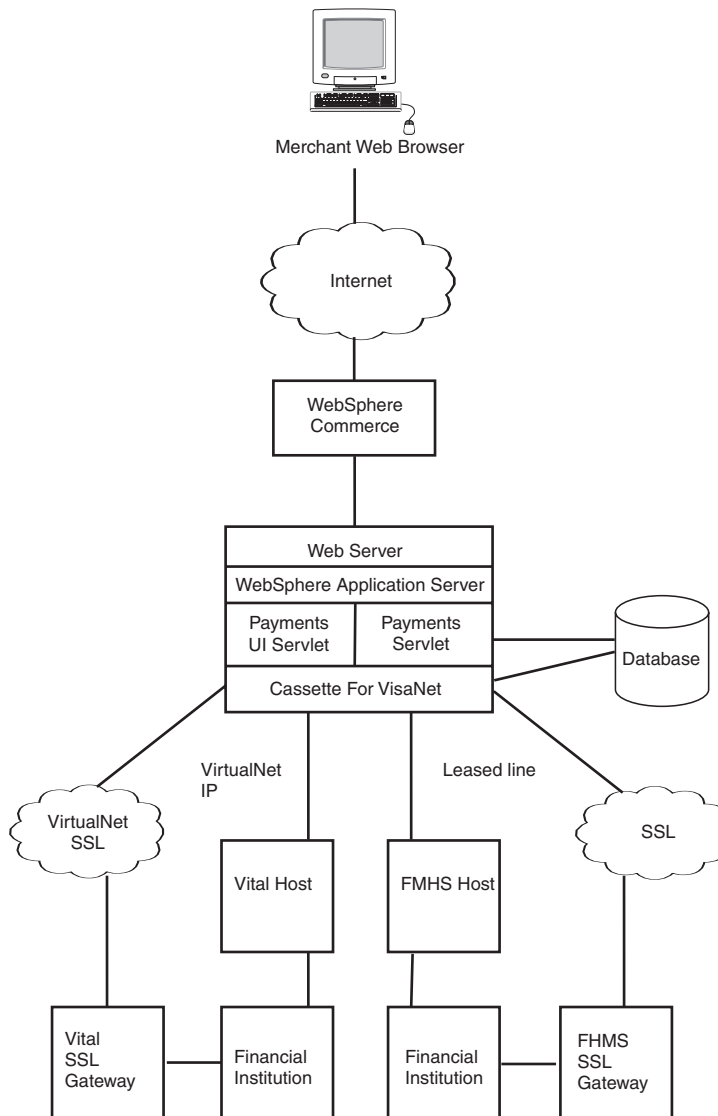


Figure 1. High-level overview of VisaNet processing

Vital Processing Services

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 (using a leased-line connection)
- Secure Sockets Layer (SSL) Gateway

The two gateways that offer connectivity into the VisaNet transaction network are VirtualNet IP and VirtualNet SSL. 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.

The Vital VirtualNet SSL Gateway provides acquirers the ability to receive real-time credit card transactions from the Internet using the encryption capability of SSL technology. The service is provided on behalf of Vital by UUNET, using an SSL server that interfaces to the UUNET Transaction Network. The gateway uses HTTP and operates over SSL.

When communicating with the Vital host, 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.

FHMS

FHMS is a leading provider of transaction processing and bankcard acquiring services. FHMS provides processing services for major credit cards and debit cards. FHMS is a subsidiary of First Tennessee National Corporation (FTNC), which provides processing services for major card types, as well as most regional and national debit and EBT cards. FHMS also offers merchants similar connectivity options:

- IP Gateway (using a leased-line connection)
- SSL Gateway

Communication methods

The Cassette for VisaNet supports the use of either a leased-line or SSL gateway to communicate with backend host systems.

The communication method is determined by the account that is set up with the provider of processing services. Consideration of the type of communication method includes these factors:

- Cost. The cost of using SSL is typically less than that of a leased line.
- Transaction volume. If you expect to process a high volume of payment transactions, performance is normally better through a leased line.
- Availability. Leased-line communication is usually more reliable than communication through the open Internet.

The type of backend host (Vital or FHMS) with which to communicate is specified for an account through the Account settings as described in "Step 7: Creating an account" on page 27.

The cassette can be configured, on an account basis, to communicate with either Vital or FHMS. However, communications with the backend host can occur only one way—either through leased line, or SSL. In other words, you cannot use the cassette, for example, to communicate with FHMS through *both* leased line and SSL.

VisaNet merchant registration

Before you can use the Cassette for VisaNet, you must ensure that the merchant has been registered to process VisaNet transactions. Merchant registration information is required by the cassette to process payment transactions. (“Step 6: Configuring the merchant cassette settings” on page 26 shows an example of how the data is used.) The following sections describe more about the registration process.

Registering when using Vital Processing Services

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.

Registering when using FHMS

The registration process and outputs of the registration process are the same as that for Vital.

American Express CID support

American Express CID is a card verification tool designed to reduce fraud losses, usually on transactions when the card is not available. The CID is a 3 or 4 digit value (following the credit card account number) on the signature line of the credit card.

A merchant does not automatically get American Express CID verification by setting the \$CARDVERIFYCODE keyword. In addition to setting this parameter (described in “AcceptPayment” on page 41), the acquiring bank must enable CID verification, for the merchant, by contacting American Express merchant services.

Merchant identification

The reporting structure in FHMS is different than that of Vital Processing Services. Vital assigns a single merchant number to a merchant, whereas FHMS can assign more than one merchant ID to a merchant. FHMS provides transaction reports to the merchant based on these merchant IDs.

In WebSphere Commerce, a store maps a single WebSphere Commerce Payments merchant to multiple FHMS merchant IDs (to associate multiple departments, for example, to a single store). If you are using the FHMS backend host and have more than one FHMS merchant ID and need to see transaction reports based on these IDs, you can specify a unique FHMS merchant ID for every account you create for the merchant. A unique FHMS merchant ID can be specified on the Accounts window as described in “Step 7: Creating an account” on page 27.

Cassette features

The following sections describe special features of the Cassette for VisaNet:

- Address Verification Service support
- Purchasing card support
- Sensitive data protection

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 61 for more information on possible AVS codes.

Purchasing cards

Purchase transactions can also involve the processing of *purchasing card* data. 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 cardholders 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 under the following circumstances:

- 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.

Sensitive data protection

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 JVM system 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. The parameter is defined through the WebSphere Commerce Configuration Manager by setting the Minimum Access Role field for the Payments instance to a value of clerk, supervisor, madmin (Merchant Administrator), psadmin (Payments Administrator), or none (no one is allowed to view sensitive data).

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 1. 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.

Table 1. Sensitive data processed by Cassette for VisaNet (continued)

Data	How data is protected
\$CARDVERIFYCODE	Verification code for the payment card. The entire value is masked with asterisks.

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 *WebSphere Commerce Payments Programming Guide and Reference*.

WebSphere Commerce Payments roles

WebSphere Commerce Payments enforces roles such that each user is presented with a different view based on the user's role, for example, from the perspective of a Payments Administrator versus a Merchant Administrator. Within the merchant organization, WebSphere Commerce Payments enables the notion of different roles so that the merchant can monitor their own users. For example, a Clerk is restricted to operations such as approving an order, while a Merchant or Payments Administrator can modify a relationship with a financial institution.

When you create users within the WebSphere Commerce Organization Administration Console, you must first assign those users a WebSphere Commerce role. Then the users will display in the Payments user interface where you can assign them a Payments roles. It is recommended that these roles be assigned to WebSphere Commerce users having the roles shown in Table 2.

Table 2. Suggested role assignment

Payments role	WebSphere Commerce role
Payments Administrator	Site Administrator
Merchant Administrator	Site Administrator
Supervisor	Operations Manager, Sales Manager
Clerk	Customer Service Supervisor

For more information about WebSphere Commerce roles, refer to the Roles topic in the WebSphere Commerce Production online help.

Both Payments Administrators and Merchant Administrators can manage WebSphere Commerce Payments. Supervisors and Clerks are financial roles. While they do not administer WebSphere Commerce Payments, they do manage the payment-processing functions. The following table describes the responsibilities for each Payments role:

Table 3. Role responsibilities

Role	Responsibilities
Payments Administrator	<ul style="list-style-type: none"> • Define Merchant Administrators, Supervisors, and Clerks • Configure merchants and their cassettes • Identify the Payments host name and status • Configure any installed cassettes • Add, delete, and update event listeners • Settle payments • Approve or sale orders • Issue credits and reverse credits • Deposit orders • Search for orders and batches • View daily batch totals
Merchant Administrator	<ul style="list-style-type: none"> • Define Merchant Administrators, Supervisors, and Clerks • Configure merchants and their cassettes • Add, delete, and update event listeners
Supervisor	<ul style="list-style-type: none"> • Settle payments • Approve or sale orders • Issue credits and reverse credits • Deposit orders • Search for orders and batches • View daily batch totals
Clerk	<ul style="list-style-type: none"> • Settle payments • Approve or sale orders • Deposit orders • Search for orders and batches • View daily batch totals

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.

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 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.

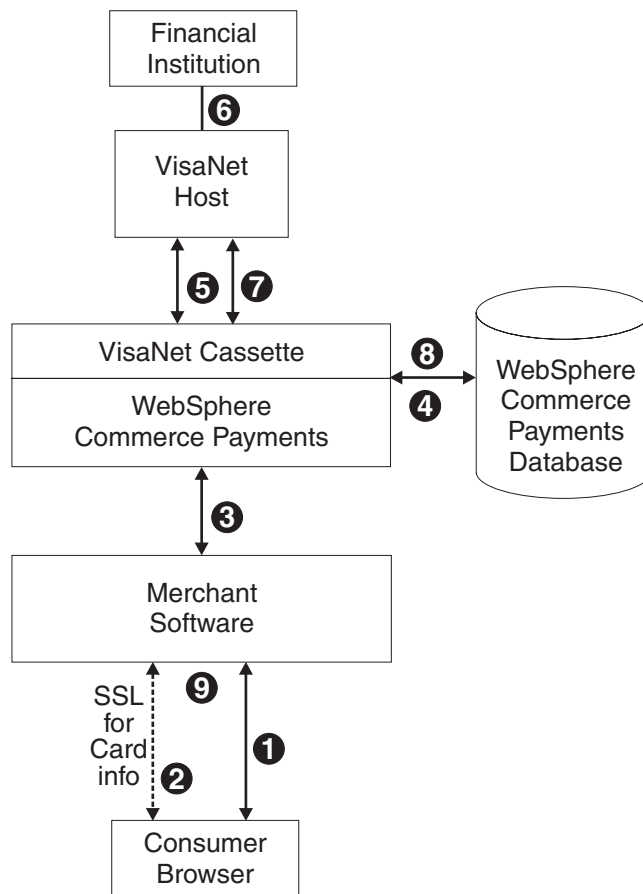


Figure 2. Example of a typical purchase flow using the Cassette for VisaNet

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.

- 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 provided by the WebSphere Commerce Payments framework.

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 *WebSphere Commerce Payments Programming Guide and Reference* for a 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, “Object reference”, on page 51.

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.

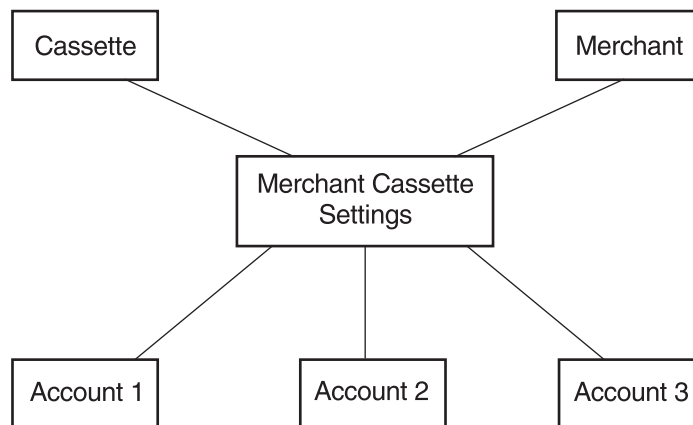


Figure 3. VisaNet PaySystem

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, “Object reference”, on page 51. For descriptions of the payment objects and for programming information, see the *WebSphere Commerce Payments Programming 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.

Retry parameters

The Cassette for VisaNet extends the WebSphere Commerce Payments Cassette object with several parameters related to communicating with the VisaNet host. These parameters appear on the VisaNet Cassette Settings screen as follows:

- SSL Read Timeout
- SSL Max Retries
- Attempt Interval
- Max Attempts

There are four connection methods: Vital and leased line, FHMS and leased line, Vital and SSL, and FHMS and SSL. The SSL Read Timeout field and SSL Max Retries field are applicable only to Vital and SSL, and FHMS and SSL.

The Attempt Interval field and Max Attempts field are applicable to all connection methods and control the attempts of the cassette to recover after failed communications with the VisaNet host. These parameters are used in pending operations and are not actually related to connection methods and communication protocol.

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, “Command reference”, on page 41. The cassette will attempt to recover communications failures under the direction of the above parameters.

For each command that requires communication with the VisaNet host, a connection must be established between WebSphere Commerce Payments and the VisaNet host. Once this connection is established, the Cassette for VisaNet attempts to send an appropriate request message and then waits for a period of time for a response. The amount of time the cassette will wait for a response is based on the

ReadTimeout parameter (this parameter is called \$READTIMEOUT on the ModifyCassette API command). If the cassette receives a response message indicating that the request is complete before a timeout occurs, the message exchange is considered complete. Otherwise, this is considered one communication attempt, and the cassette will immediately retry the operation based on the **Max Retries** parameter (this parameter is called \$MAXRETRIES on the ModifyCassette API command). If the communication is unsuccessful after all immediate retries have been attempted, the cassette enters "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.

Guidelines for setting retry parameters

Care should be taken when setting up the cassette's communication retry parameters. If timeouts and or retries are excessive, the performance of the cassette could be adversely affected. It is strongly recommended that the following guidelines are followed:

- The combination of the ReadTimeout and the MaxRetries should never equal or exceed 3 minutes. For example, if the ReadTimeout is specified as **60 seconds**, and the MaxRetries is specified as **3**, then the combined timeout value would be **3 minutes**.
- To ensure good cassette performance and throughput, keep the ReadTimeout as low as possible. It is recommended that the ReadTimeout be specified as **30 seconds** or less.

Cassette for VisaNet payment command summary

Table 4 on page 16 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 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 4. 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
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.

Table 5. State changes

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

Table 5. State changes (continued)

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

Chapter 3. Before you start

The cassette software is installed when the WebSphere Commerce Payments component is installed as part of your WebSphere Commerce installation. Unlike previous versions, you do *not* need to install the Cassette for VisaNet software in addition to the WebSphere Commerce Payments framework software. The WebSphere Commerce installation program will ensure that all prerequisite products necessary for the WebSphere Commerce Payments framework and cassette to function are available. For more information about how to install the WebSphere Commerce Payments component, refer to the *WebSphere Commerce Installation Guide*.

The minimum Payments framework level supported by the cassette is 5.5. You cannot use the Cassette for VisaNet Version 5.5 with earlier versions of the WebSphere Commerce Payments framework.

Before you can configure the Cassette for VisaNet, you must do the following:

- Ensure that the WebSphere Commerce Payments component was installed as part of your WebSphere Commerce installation.
- Create a WebSphere Commerce Payments instance, or use an existing Payments instance to which you can add this cassette.
- Use the WebSphere Commerce Configuration Manager to add the cassette to the Payments instance.
- Start the Payments instance.
- Define a WebSphere Commerce Payments user with administrative authority.
- Register as a VisaNet merchant.
- Create a merchant and Merchant administrator for that merchant.

To configure the cassette, you must log on to WebSphere Commerce Payments as a Merchant or Payments Administrator.

Chapter 4. Tutorial

This tutorial guides you through an initial setup and configuration of the Cassette for VisaNet. You must configure the cassette before you can process customer transactions. As part of this initial setup, WebSphere Commerce Payments provides tutorial support using the Cassette for VisaNet and a Sample Checkout application. For detailed information on administration, configuration, and payment functions, see the online help for the WebSphere Commerce Payments user interface.

Note: The steps you perform in this tutorial using the WebSphere Commerce Payments user interface are very similar to how you would perform them in production using the WebSphere Commerce user interface (Administration Console or Accelerator). The windows or navigation may be slightly different, however, in the WebSphere Commerce Administration Console or Accelerator. For example, the tutorial mentions a "Navigation frame" in the WebSphere Commerce Payments user interface. In the WebSphere Commerce user interface, this frame is not displayed. Use the equivalent functions in the WebSphere Commerce user interface to perform the tasks in a real situation.

Following are the tasks described in this tutorial to set up an operational Cassette for VisaNet:

1. Access the WebSphere Commerce Payments user interface.
2. Configure the cassette.
3. Create a WebSphere Commerce Payments merchant and authorize the merchant to use the cassette.
4. Define WebSphere Commerce Payments users.
5. Assign user roles.
6. Configure the merchant cassette settings.
7. Create an account.
8. Create orders.

After the orders are created, you are ready to begin the following payment-processing tasks that merchants typically perform on a daily basis:

9. Approve orders and deposit payments.
10. Settle batches.
11. Issue credits.
12. View daily batch totals.

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.

Step 1: Accessing 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 log onto the WebSphere Commerce Payments user interface, do the following:

1. In a Web browser point to `http:// host_name:port/webapp/PaymentManager/`, where *host_name* is the host name of the machine running the Web Server for Payments, and *port* refers to the port number Payments is running on as shown in the Configuration Manager WebServer information for your Payments instance.

If you are using SSL with the Payments instance, use `https://` instead.


2. Type your WebSphere Commerce User ID.
3. Type your corresponding WebSphere Commerce Password.
4. Click **Logon**.


Important: If the HTTP server that the WebSphere Commerce Payments instance is using is configured for a port number other than that specified in the WebSphere Commerce Configuration Manager as the default, include the port number following the host name in the WebSphere Commerce Payments Web address links throughout this tutorial.

The icons in the upper right page of the user interface have the following uses:

- Click the multidirectional arrow to refresh the page.
- Click the left-pointing arrow to return to the last page visited, instead of your browser's back button.
- Click the question mark to access context sensitive online help for the page.

Step 2: Configuring the cassette to communicate with external sites

 During this tutorial, your WebSphere Commerce Payments will attempt to send messages to the Vital Processing Services system and FHMS system. You will need to configure settings to enable this communication. Also, 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.

 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 Cassette for VisaNet. For more information about configuring the iSeries TCP/IP SOCKS support, see the iSeries Information Center and online library at: <http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm>. 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.

To configure the cassette for external communications, do the following:

1. Click on **Cassettes** in the navigation frame.
2. Click on the **VisaNet** cassette icon. At the next page, you will see several entry fields that allow you to tailor the way the Cassette for VisaNet communicates with the external hosts. Enter the following information as appropriate.

The fields that are required will vary depending on what host and communication method you are using. Because the cassette supports only one method of communication (leased line or SSL) per target communication host (Vital or FHMS), you cannot supply data in all the fields shown. A message is displayed to inform you if you exceed the number of entries permitted. In a production environment, if you need to use both the Vital and FHMS target host systems, you must set up two different accounts. Account creation is described later in “Step 7: Creating an account” on page 27.

Table 6. Cassette for VisaNet page settings

Field	Description
VirtualNet IP Host Name	The TCP/IP host name to access the Vital host. If you enter a value in this field you must also enter a value in the Port Number field. If this field is blank, the Port Number field must also be blank.
Port Number	The TCP/IP port number to access the Vital host. If you enter a value in this field, you must also enter a value in the VirtualNet IP Host Name field. If this field is blank, the VirtualNet IP Host Name field must be blank.
FHMS IP Host Name	The TCP/IP host name to access the FHMS host. If you enter a value in this field, you must also enter a value in the FHMS Auth Port Number and FHMS Batch Port Number fields. If one of these three fields is blank, all three fields must be blank.
FHMS Auth Port Number	The TCP/IP port number to access the FHMS host for credit card approval. If you enter a value in this field, you must also enter a value in the FHMS IP Host Name and FHMS Batch Port Number fields. If one of these three fields is blank, all three fields must be blank.
FHMS Batch Port Number	The TCP/IP port number to access the FHMS host for batch settlement. If you enter a value in this field, you must also enter a value in the FHMS IP Host Name and FHMS Auth Port Number fields. If one of these three fields is blank, all three fields must be blank.
VirtualNet SSL Gateway URL	The URL to access Vital’s VirtualNet SSL Gateway.
FHMS SSL Gateway URL	The URL to access the FHMS SSL Gateway.
SOCKS Host Name	The fully qualified TCP host address for the SOCKS server. Note: Enter this name carefully (for example, <i>mycomputer.city.company.com</i> is fully qualified).*
SOCKS Port Number	The TCP port number for the SOCKS server.
*If you enter the wrong SOCKS host name, the Payments application server may fail when a Payments transaction is attempted and you will not be able to use the Payments GUI. If this happens, stop and restart the Payments instance, return to this page, and enter the correct SOCKS host name.	

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

Step 3: Creating a WebSphere Commerce Payments merchant and authorizing a cassette

If you have not already done so, use your WebSphere Commerce user ID to log on to WebSphere Commerce Payments as the Payments Administrator. You now have global views and global authority. The first step in configuring WebSphere Commerce Payments is to create a merchant and authorize that merchant to use a payment cassette. Do the following to create a merchant and authorize a cassette:

1. From the navigation frame click **Merchant Settings**.
2. From the Merchant Settings page click **Add a Merchant**.
3. On the Merchant Settings page, type the following information.

Table 7. Merchant settings: Create a merchant

Field	Description
Merchant name	Type Test Store. This is the name that you assign to the merchant. Its only function is to provide display information in the user interface.
Merchant number	Type 123456789. This is a number that you assign which uniquely identifies the merchant in all transaction data.
Authorized cassettes	Check the box next to <i>VisaNet</i> . Checking this box authorizes the merchant to use the Cassette for VisaNet.

4. Click **Create Merchant** to save the merchant configuration.

If you have already created a merchant whom you want to authorize to use this cassette, perform these steps:

1. Click **Merchant Settings**.

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.

2. Click the Merchant Name.
3. Select the box for **VisaNet**.
4. Click **Update**.

The merchant is now authorized to use the cassette.

iSeries support of SSL for the Cassette for VisaNet

400 On the iSeries, when the Cassete for VisaNet is configured to use SSL, it will use .jks (java keyStore) certificates in the system default truststore, /QIBM/ProdData/Java400/jdk13/lib/security/cacert. This truststore is a collection of Certificate Authority certificates that the iSeries SSL support will trust. By using the system default truststore, the iSeries SSL support will accept the certificates signed by most of the major Certificate Authorities. If your Cassete for VisaNet gateway sends a certificate which the iSeries cannot accept, you will need to do the following:

- Create a new truststore.
- Add the Gateway certificate to that truststore.
- Configure your WebSphere Application Server to use the new truststore.

Refer to the product README for more information about how to perform these tasks.

Step 4: Defining WebSphere Commerce Payments users

For this tutorial, you will work with the following users:

- A WebSphere Commerce Site Administrator user ID created during installation (for more information refer to the *WebSphere Commerce Installation Guide*).
- *Pat*, a user you will define.

You will use the WebSphere Commerce Organization Administration Console to accomplish tasks such as defining and managing users. Defining users in WebSphere Commerce Payments is a two-part process. For example, to define the user *Pat* you must use the WebSphere Commerce Organization Administration Console and create and assign *Pat* a WebSphere Commerce role. Then, you can assign *Pat*'s user role to Merchant Administrator within the Payments user interface directly, or through the WebSphere Commerce Administration Console. Note that before you can assign access to a user, you must create a merchant.

To configure Payments users, do the following:

1. In a Web browser point to `https://host_name:8004/orgadminconsole`.
2. Click **Access Management>Users**.
3. Click **New**.
4. Create the new user, *Pat*, using the New User wizard.
5. From the Roles page, assign *Pat* a WebSphere Commerce role.

Step 5: Assigning user roles

Users must be assigned to one of the WebSphere Commerce Payments roles listed in the following table. It is recommended that these Payments roles be assigned to WebSphere Commerce users having the roles shown in the table.

Table 8. Role mapping

Payments role	WebSphere Commerce role
Payments Administrator	Site Administrator
Merchant Administrator	Site Administrator
Supervisor	Operations Manager, Sales Manager
Clerk	Customer Service Supervisor

After creating the following users, you are ready to assign *Pat*'s role in the WebSphere Commerce Payments configuration:

- A user, *Pat*
- A merchant, *Test Store*

Exception: You can also assign the role No WebSphere Commerce Payments access to deny users access to WebSphere Commerce Payments. For more information on

WebSphere Commerce Payments role permissions, see the Role Permissions Table in the *WebSphere Commerce Payments Programming Guide and Reference*.

To assign Pat the role of Merchant Administrator for the Test Store, do the following:

1. In a Web browser point to `http://host_name:port/webapp/PaymentManager` to log on to Payments.
If you are using SSL with the Payments instance, use `https://` instead.
2. From the navigation frame click **Users**.
3. On the Users Search page, type the user name Pat and click **Search**.
4. From the Users page, click the user name **Pat**.
5. From the **Merchant** scroll box, select the merchant name. For example, **Test Store**.
6. Select the radio button for **Merchant Administrator**.
7. Click **Update** to save the user configuration.

At this point, you should log off the WebSphere Commerce Payments user interface and log on again, this time as the Merchant Administrator, Pat.

Logging in as the Merchant Administrator

To log off and log in again, do the following:

1. From the navigation frame, click **Logoff user** on the navigation frame of the WebSphere Commerce Payments user interface, and you will return to the main WebSphere Commerce Payments Login window.
2. Type the user ID (for example, **Pat**).
3. Type the **Password** defined for the user, as created during the new WebSphere Commerce user process.
4. Click **OK**.

For the remainder of the tutorial, your role will be the Merchant Administrator for the Test Store. 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 functions.

Step 6: Configuring the merchant cassette settings

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, do the following:

1. From the navigation frame click **Merchant Settings**.
2. From the Merchant Settings page, click the **Cassette for VisaNet** icon in the Test Store.
3. From the Cassette for VisaNet page, click **Merchant Cassette Settings**.
4. At the next page, you will be prompted to enter the following information:

Table 9. Merchant Cassette Settings

Field	Description
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. Note: This value can be overridden by a Merchant ID value specified on the Accounts window (as described in "Step 7: Creating an account").
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.

Step 7: 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, do the following:

1. From the navigation frame click **Merchant Settings**.
2. From the Merchant Settings page, click the **Cassette for VisaNet** icon in the Test Store.
3. From the Cassette for VisaNet page, click **Accounts**.
4. On the Accounts page, click **Add an Account**.
5. Complete the following fields (note that the italicized text *must* be entered in these fields for the tutorial):

Table 10. Add an account settings

Field	Description
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.
Communication Host	This is the target communication host with which the account is communicating. Select either VITAL or FHMS. If the host is not specified, VITAL is assumed. Note: If your actual environment requires communication with <i>both</i> Vital and FHMS, you must set up two different accounts.
Merchant ID	(Optional) The Merchant ID provided by FHMS (1–12 digits). Enter this value only if the Communication Host is FHMS. This value overrides the VisaNet Merchant Number provided in the Merchant Cassette Settings window described in "Step 6: Configuring the merchant cassette settings" on page 26. This value can be used to specify a unique department for the merchant, so that payment transactions can be identified as being associated with a particular department or unit within the store.

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

Step 8: Creating orders using the Sample Checkout

As the Merchant Administrator, you have global merchant authority, which means that you can do the following:

- 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 Sample Checkout to create three orders for use in payment processing. The Sample Checkout tool provides a user interface you can use to create sample orders to test your cassette implementation. Note that to access Sample Checkout, you must first edit a file as described in the following section.

To access the WebSphere Commerce Payments Sample Checkout and create orders, do the following:

Note: Remember that this is a sample application. In a real production environment, the actual windows you use to create orders may be slightly different.

1. Open the `SampleCheckout.xml` file in the following directory:

```
WAS_installdir/installedApps/host_name/payments_instance_Commerce_Payments_App.ear/  
SampleCheckout.war
```

▶ 400 For iSeries, the directory path is

```
/QIBM/UserData/WebAS5/Base/WAS_instance/installedApps/node_name/  
Payments_instance_Commerce_Payments_App.ear/SampleCheckout.war
```

2. At the `SampleCheckout` element, change the following attribute values:

```
pmHostname="fully_qualified_host_name"  
pmPort="port"  
userid="wc_userid"  
password="wc_password"
```

For `pmHostname`, enter the fully qualified host name for the WebSphere Commerce Payments Web server. For `pmPort`, enter the port number WebSphere Commerce Payments is running on as shown in the Configuration Manager WebServer information for your Payments instance. For the `userid` and `password`, enter the user ID and password associated with the WebSphere Commerce user.

If you are using SSL with the Payments instance, be sure to also specify the value of "1" for the `useSSL` attribute (`useSSL="1"`).

3. Save the file.
4. Point your browser to `http://host_name:port/webapp/SampleCheckout/`, where `host_name` is the host name of the machine running the Web Server for Payments, and `port` refers to the port number Payments is running on as shown in the Configuration Manager WebServer information for your Payments instance.

If you are using SSL with the Payments instance, use `https://` instead.

5. At the Sample Checkout page enter the following (note that all fields except the Card verification value *must* be entered in these fields for the tutorial):

Table 11. Sample Checkout fields for Cassette for VisaNet

Field	Description
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.

6. Click **Buy**.

Repeat these steps two more times (each time with a different order number) so that you have three orders for which to process payments.

Step 9: 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, you can approve these orders. Follow these steps to approve and deposit an order:

1. Point your browser again to `http://host_name:port/webapp/PaymentManager/` and log in as the Merchant Administrator for the Test Store merchant (for example, Pat).
If you are using SSL with the Payments instance, use `https://` instead.
2. From the navigation frame, click **Approve**.
3. From the Approve page, 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**.
4. The Approve Results page displays the status of your sale request. When processing is complete, success or failure status will appear next to each order submitted for sale.
5. When your sale is complete, click **Return to the Approve Screen**.

Two orders are still awaiting your approved sale. You could have approved them all at once (for their full amounts), by clicking **Sale All** from the Approve page. However, to better demonstrate the approve functions, this tutorial guides you to work with each order individually.

Approving orders from the Order page

In this section, you will approve and deposit an order from the Order page (rather than from the Approve page), but you will approve only *part* of the total order amount. You may find it useful to approve only part of an order when some of the goods associated with the order are not available for delivery at order processing.

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

Table 12. Order Approve fields

Field	Description
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 page refreshes and displays the sale approval status. You will notice that approval and deposit amounts in the Order page 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):

Table 13. Payment details

Field	Description
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).

Table 13. Payment details (continued)

Field	Description
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 cassette or protocol 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 61.
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 page by clicking **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 page, 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. Do the following:

1. Click **Sale All** in the Approve page.
2. In the Approve Results page, a progress bar indicates the status of your sale request. When processing is complete, the status of the approval is displayed 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.
3. When this step is complete, click **Return to the Approve Screen**.

Separate approvals and deposits

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

Approve

Approval without deposit is performed through the same pages as the Sale function (that is, the Approve or Order pages). Instead of clicking 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 through 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 32, 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**.
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 page, a progress bar indicates the status of your deposit request. When processing is complete, the status of the deposit is displayed 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 page, click the payment number for the payment that you want to partially deposit.
2. The Payment page is displayed, as described in “Approving orders from the Order page” on page 31. Click **Deposit** 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 page, which will be updated with the new deposit amount.

Step 10: Settling batches

A batch is a collection of payments and credits that are processed as a unit by a financial institution. A batch is associated with a merchant and an account. The payments that you deposited in the previous exercise will now appear in a batch. You must *settle* this batch to initiate processing by the financial institution. The financial institution is responsible for the transfer of funds once settlement is complete.

To settle the batch that contains the payments you have created so far, do the following:

1. From the navigation frame click **Batch Search**.
2. At the Batch Search page you can enter the following information to narrow your search. For the purposes of this tutorial, you can fill in either the payment type: *VisaNet* or the account.

Table 14. Batch search fields

Field	Description
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 cassette 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). If more than 500 accounts have been defined, type the account number in the entry field.

3. Click **Search**.

Tip: You can also use the before and after fields to narrow search results by excluding certain batches from the search. For example, you could search on all batches opened before 08/01/2003 and after 08/15/2003 thus excluding batches opened between 08/02/2003 and 08/14/2003.

4. Click the batch number to view information about the batch.

5. 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, the settle status is displayed.

Note: If you use SSL as your communication method to the backend host, it is strongly recommended that you keep the batch sizes to under 1MB. Leased-line batches do not have this limitation.

Step 11: Issuing a credit

Credits are issued against orders and can be given for any amount. To issue a credit, do the following:

1. To find the order for which you want to issue the credit, from the navigation frame click **Order Search**.
2. In the Order Search page, you can type the following (note that for the purposes of this tutorial, you do not need to complete these fields to narrow your search):

Table 15. Order search fields

Field	Description
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">• Refundable• Canceled• Closed
Payment Type	Identifies the payment cassette 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). If more than 500 accounts have been defined, type the account number in the entry field.

3. Click **Search**.
4. From the Order Search Results page, click an order number for an order in Refundable state to view the details of that order.
5. From the Order page, click **Credit** to create a credit against this order.
6. The Create Credit page is displayed with the following information:

Table 16. Create credit fields

Field	Description
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.

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

When credit processing has completed, the Order page refreshes and displays the credit status. The newly created credit is displayed under **Credits**. If you view details, you will see the Transaction Date and Transaction Time.

Step 12: Viewing batch totals

The last step in this tutorial is viewing daily batch totals. The WebSphere Commerce Payments reports function allows you to view *daily totals* for batches in a closed state.

To generate a daily batch totals report, do the following:

1. From the navigation frame click **Reports**.
2. From the Reports page, click **Daily Batch Totals**.
3. At the Batch Totals Report page, type the **Date** for which you would like a batch totals report. Leave this field blank to generate a report for the current date.
4. Click **Search**.

The Daily Batch Totals report computes the totals for all batches that were closed on the date specified on the Search page. These totals are computed on a per-currency basis, so there is one line per currency. These totals cover all payments and credits made for all payments (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.

Chapter 5. Cassette for VisaNet Cashier profiles

The WebSphere Commerce Payments Cashier 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 developer 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 API commands (for example, `AcceptPayment`). 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 Programming Guide and Reference*.

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 following Cashier profiles are provided with WebSphere Commerce for the Cassette for VisaNet:

- WC51_VisaNet.profile and WC51_VisaNet_PCard.profile

These profiles can be used with the WebSphere Commerce sample stores, or in a production environment. If you do not intend to process purchasing card data, use the WC51_VisaNet.profile. The WC51_VisaNet_PCard.profile contains additional fields needed to process purchasing card data. If you used the default instance name of **demo**, the profiles are stored in the following directory: *WC_installdir/instances/demo/xml/payment*.

For more information about using payment cassettes with WebSphere Commerce sample stores, see the *WebSphere Commerce Store Development Guide*.

- SampleCheckoutVisaNet.profile

This profile can be used for test or simulation purposes. The Sample Checkout application can be used to simulate the creation of orders that require payment processing. If you use the Sample Checkout application for test purposes, be aware that it requires that the Cassette for VisaNet profile be named SampleCheckoutVisaNet.profile.

Initially, the SampleCheckoutVisaNet.profile is installed in directory path *Payments_installdir/cassettes/VisaNet/SampleCheckout/profiles*. When the cassette is added to an instance, the file is copied to *Payments_installdir/wc.mpf.ear/SampleCheckout.war/profiles*. If you want to change the profile for an instance, you must change the profile located in the following directory: *WAS_installdir/installedApps/node_name/payments_instance_Commerce_Payments_App.ear/SampleCheckout.war/profiles*.

Note: Do not change the profile in the *Payments_installdir/wc.mpf.ear/SampleCheckout.war/profiles* directory for a given instance. Changes made to the profile in this location will affect *all* Payments instances.

All of these Cashier profiles use full AVS information.

If necessary, you can edit the profile to set certain parameters, such as APPROVEFLAG and DEPOSITFLAG. (These flags are described in “AcceptPayment” on page 41.) For more details on designing and tailoring profiles, see Chapter 3 of the *WebSphere Commerce Payments Programming Guide and Reference* for the framework version you are using.

For information about customizing the cassette for use with the WebSphere Commerce sample stores, see the *WebSphere Commerce Store Development Guide*.

Chapter 6. Command reference

For each WebSphere Commerce Payments application programming interface (API) command, the following sections describe:

- 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 *WebSphere Commerce Payments Programming Guide and Reference* for a complete list 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 *WebSphere Commerce Payments Programming 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 *WebSphere Commerce Payments Programming Guide and Reference* for more information on Asynchronous Auto Approve.

Table 17. 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".

Table 18. Optional keywords for AcceptPayment command

Optional keywords	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.
\$PURCHASEORDERNUMBER	Optional Purchase Order Number assigned by the merchant. A 1–25 character value.
\$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.
\$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.

Table 18. Optional keywords for AcceptPayment command (continued)

Optional keywords	Value
\$TRANSACTIONTYPE	<p>Specifies the type of transaction. Valid values are:</p> <p>5 – 3-D Secure transaction was successful (applies only if 3-D Secure support is enabled). The values for \$VISA_CAVV and \$VISA_XID are expected.</p> <p>6 – 3-D Secure transaction was attempted but the cardholder does not participate in 3-D Secure (applies only if 3-D Secure support is enabled for the merchant). The values for \$VISA_CAVV and \$VISA_XID are not expected to be provided.</p> <p>If 3-D Secure support is enabled, the \$SECURECONNECTION keyword value is ignored.</p>
\$VISA_CAVV	<p>The cardholder authentication verification value used for credit card authorization when 3-D Secure support is enabled. Consists of a 28-byte value (a 20-character alphanumeric value that is Base64 encoded for a 28-byte result). (The cassette performs the necessary decoding.)</p> <p>If this keyword is specified, \$VISA_XID should also be specified.</p>
\$VISA_XID	<p>A unique transaction identifier determined by the merchant and used when 3-D Secure support is enabled. Consists of a 28-byte value (a 20-character alphanumeric value that is Base64 encoded for a 28-byte result). (The cassette performs the necessary decoding.)</p> <p>If this keyword is specified, \$VISA_CAVV should also be specified.</p>

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.

Table 19. Cassette-specific processing of parameters for the Approve command

Keyword	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 reopened, 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 Cassette for VisaNet, 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 *WebSphere Commerce Payments Programming Guide and Reference*.

If you use SSL, it is strongly recommended that you keep batch sizes to under 1MB. Leased-line batches do not have this limitation.

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

Table 20. 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.
\$COMMUNICATIONHOST	The 1-5 character field that identifies the type of host with which this account is communicating. Valid values are either VITAL or FHMS. If the host is not specified, VITAL is assumed.

Table 21. Optional keywords for CreateAccount command

Optional keyword	Value
\$FHMSMERCHANTID	The 1-12 digit Merchant ID assigned by FHMS. This value is used only when the \$COMMUNICATIONHOST keyword contains FHMS. If a value is assigned to this field, it will override the value for \$VISAMERCHANTNUMBER defined in the Merchant Cassette Settings (PaySystemAdmin object).

CreatePaySystem

Table 22. Required keywords for CreatePaySystem command

Required keywords	Value
\$VISANETMERCHANTNUMBER	The 1-12 digit Merchant number assigned by the merchant's bank or processor.

Table 22. Required keywords for CreatePaySystem command (continued)

Required keywords	Value
\$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	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-xxxxxxx format. The dash is required.
\$MERCHANTSTATE	A 2-character field containing the merchant state or province code provided by the signing member or processor.

Table 23. 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.

Table 24. Optional keywords for Deposits command

Optional keywords	Value
\$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

Table 25. 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	This is a 1–6 digit value 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.

Table 25. Optional keywords for ModifyAccount command (continued)

Optional keywords	Value
\$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.
\$FHMSMERCHANTID	The 1–12 digit Merchant ID assigned by FHMS. This value is used only when the \$COMMUNICATIONHOST keyword contains FHMS. If a value is assigned to this field, it will override the value for \$VISAMERCHANTNUMBER defined in the Merchant Cassette Settings (PaySystemAdmin object).

ModifyCassette

Table 26. 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 (0–254 character string). Specify the fully qualified host name carefully. For example, <i>mycomputer.city.company.com</i> is fully qualified. If you enter the wrong SOCKS host name, the Payments application server may fail when a Payments transaction is attempted and you will not be able to use the Payments GUI. (To recover, stop and restart the Payments instance.)
\$SOCKSPORTNUMBER	TCP Port Number for socks server. An integer between 0 and 2147483647.
\$HOSTNAME	The TCP/IP host name to access the VisaNet host (1–254 character string). If specified, the \$PORTNUMBER keyword must also be specified. If this keyword contains a null value, \$PORTNUMBER must contain a null value.
\$PORTNUMBER	The TCP/IP port number to access the VisaNet host. An integer between 0 and 2147483647. If specified, the \$HOSTNAME keyword must also be specified. If this keyword contains a null value, \$HOSTNAME must contain a null value.

Table 26. Optional keywords for ModifyCassette command (continued)

Optional keywords	Value
\$FHMSHOSTNAME	The TCP/IP host name to access the FHMS host (1–254 character string). If specified, you must also specify a value for \$FHMSAUTHPORTNUMBER and FHMSBATCHPORTNUMBER. If one of these three keywords contains a null value, all three must contain null values.
\$FHMSAUTHPORTNUMBER	The TCP/IP port number to access the FHMS host for authorization. An integer between 0 and 2147483647. If specified, you must also specify a value for \$FHMSHOSTNAME and FHMSBATCHPORTNUMBER. If one of these three keywords contains a null value, all three must contain null values.
\$FHMSBATCHPORTNUMBER	The TCP/IP port number to access the FHMS host for batch settlement. An integer between 0 and 2147483647. If specified, you must also specify a value for \$FHMSHOSTNAME and FHMSAUTHPORTNUMBER. If one of these three keywords contains a null value, all three must contain null values.
\$FHMSURL	The SSL gateway to access the FHMS host through HTTPS (1–254 character string).
\$VITALURL	The SSL gateway to access the Vital host through HTTPS (1–254 character string).
\$MAXRETRIES	Maximum number of retry sets. An integer between 0 and 2147483647. This keyword applies to SSL only.
\$READTIMEOUT	Number of seconds to wait until trying the next read attempt. An integer between 0 and 2147483647. This keyword applies to SSL only.

ModifyPaySystem

Table 27. 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.

Table 27. Optional keywords for ModifyPaySystem command (continued)

Optional keywords	Value
\$TIMEZONEDIFFERENTIAL	The 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	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.

Table 28. Cassette-specific processing of parameters for the Refund command.

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. Object reference

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

Table 29. 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>
</PSApiResponse>

```

VisaNet Payment

Table 30. 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 61.
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.

Table 30. Cassette properties that belong to a PSPayment Object (continued)

Field name	Description
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="968461230000" 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

Table 31. 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

Table 32. 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>
```

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

CassetteAdmin

Table 33. Cassette properties that belong to a `PSCassette` Object

Field name	Description
<code>hostName</code>	The TCP/IP host name to access the Vital host.
<code>portNumber</code>	The TCP/IP port number to access the Vital host.
<code>fhmsHostName</code>	The TCP/IP host name used to access the FHMS host.
<code>fhmsAuthPortNumber</code>	The TCP/IP port number to access the FHMS host for authorizations.
<code>fhmsBatchPortNumber</code>	The TCP/IP port number to access the FHMS host for settlement.
<code>vitalURL</code>	The URL used to access the Vital SSL Gateway.
<code>fhmsURL</code>	The URL used to access the FHMS SSL Gateway.
<code>socksHostName</code>	TCP host address for SOCKS server.
<code>socksPortNumber</code>	TCP port number for SOCKS server.
<code>maxAttempts</code>	The maximum number of retry sets. The default is 3.
<code>attemptInterval</code>	The number of seconds to wait before trying the next set of retries. The default is 60 seconds (1 minute).
<code>readTimeout</code>	The number of seconds to wait before trying to read from the host. Applies to SSL only.
<code>maxRetries</code>	The maximum number of retries reading from the host. Applies to SSL only.

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" 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>
<CassetteProperty>
<CassetteProperty propertyId="maxRetries" value="3">
</CassetteProperty>
<CassetteProperty>
<CassetteProperty propertyId="readTimeout" value="30">
</CassetteProperty>
<CassetteProperty>
<CassetteProperty propertyId="fhmsHostName" value="">
</CassetteProperty>
<CassetteProperty>
<CassetteProperty propertyId="fhmsAuthPortNumber" value="">
</CassetteProperty>
<CassetteProperty>
<CassetteProperty propertyId="fhmsBatchPortNumber" value="">
</CassetteProperty>
<CassetteProperty>
<CassetteProperty propertyId="fhmsURL"
value="https://gateway-bmd.nxt.com:4443/FTB/process-transaction.cgi">
</CassetteProperty>
<CassetteProperty>
<CassetteProperty propertyId="vitalURL" value="">
</CassetteProperty>

</CassetteExtensionObject>
</PSCassette>
</CassetteCollection>
</PSApiResult>

```

AccountAdmin

Table 34. 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.

Table 34. Cassette properties that belong to a PSMerchantAccount (continued)

Field name	Description
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.
communicationHost	A 1–5 character field identifying the host with which this account is communicating. Valid values are VITAL or FHMS.
fhmsMerchantID	The 1–12 digit Merchant ID assigned by FHMS. If a value is assigned to this field, the value will override the \$VISANETMERCHANTNUMBER defined in the Merchant Cassette Settings (PaySystemAdmin object).

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>
        <CassetteProperty propertyId="communicationHost" value="FHMS">
        </CassetteProperty>
        <CassetteProperty propertyId="fhmsMerchantID" value="192000950">
        </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:

Table 35. 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 or region.
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 or region.
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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 36. WebSphere Commerce Payments common AVS result codes mapped to Cassette for VisaNet AVS result codes

VisaNet AVS result code (returned from acquirer)	WebSphere Commerce Payments 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.
N	3	Neither the street address nor the postal code matches.

Table 36. WebSphere Commerce Payments common AVS result codes mapped to Cassette for VisaNet AVS result codes (continued)

VisaNet AVS result code (returned from acquirer)	WebSphere Commerce Payments common AVS result code	Explanations
R	4	This constant maps: <ul style="list-style-type: none"> • Address information unavailable • System unavailable (maybe ude 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 ude 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 ude 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 ude 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. Cassette for 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 *WebSphere Commerce Payments Programming Guide and Reference* for this list.
- **Secondary Return Codes:** The majority of the secondary return codes generated by the Cassette for VisaNet are defined by the framework (See the *WebSphere Commerce Payments Programming Guide and Reference* for a list.) The following table lists VisaNet-specific errors and their definitions.

Secondary Return Code	Value	Description
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.

Secondary Return Code	Value	Description
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_CASSETTE_VITALURL	10029	Response refers to the \$VITALURL parameter
SRC_CASSETTE_FHMSURL	10030	Response refers to the \$FHMSURL parameter
SRC_CASSETTE_FHMSHOSTNAME	10031	Response refers to the \$FHMSHOSTNAME parameter
SRC_CASSETTE_FHMSAUTHPORTNUMBER	10032	Response refers to the \$FHMSAUTHPORTNUMBER parameter
SRC_CASSETTE_FHMSBATCHPORTNUMBER	10033	Response refers to the \$FHMSBATCHPORTNUMBER parameter
SRC_CASSETTE_COMMHOSTNAME	10034	Response refers to the \$COMMUNICATIONHOST parameter
SRC_CASSETTE_FHMSMERCHANTID	10035	Response refers to the \$FHMSMERCHANTID parameter
SRC_CASSETTE_READTIMEOUT	10036	Response refers to the \$READTIMEOUT parameter

Secondary Return Code	Value	Description
SRC_CASSETTE_MAXRETRIES	10037	Response refers to the \$MAXRETRIES parameter
SRC_METHOD_NOT_IMPLEMENTED	20000	The VisaNet 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 VisaNet accounts.
SRC_ACCOUNT_CREATE_SQL_FAILURE	21001	Database error occurred while attempting to create VisaNet account.
SRC_ACCOUNT_UPDATE_SQL_FAILURE	21002	Database error occurred while attempting to update a VisaNet account.
SRC_ACCOUNT_DELETE_SQL_FAILURE	21003	Database error occurred while attempting to delete a VisaNet 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 VisaNet batch
SRC_BATCH_CREATE_SQL_FAILURE	22003	Database error occurred while attempting to create a VisaNet batch
SRC_BATCH_UPDATE_SQL_FAILURE	22004	Database error occurred while attempting to update a VisaNet batch
SRC_BATCH_DELETE_SQL_FAILURE	22005	Database error occurred while attempting to delete a VisaNet batch
SRC_BATCH_NULL_ORDER_FOR_PAYMENT	22006	Framework associated with the VisaNet Batch's payment cannot be retrieved
SRC_BATCH_NULL_PAYMENT	22007	VisaNet Batch payment cannot be retrieved.
SRC_BATCH_BAD_BATCH_IN_PAYMENT	22008	VisaNet 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 VisaNet Batch's credit cannot be retrieved
SRC_BATCH_NULL_CREDIT	22010	VisaNet Batch credit cannot be retrieved.

Secondary Return Code	Value	Description
SRC_BATCH_BAD_BATCH_IN_CREDIT	22011	VisaNet 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 VisaNet order
SRC_ORDER_CREATE_SQL_FAILURE	23002	Database error occurred while attempting to create a VisaNet order
SRC_ORDER_UPDATE_SQL_FAILURE	23003	Database error occurred while attempting to update a VisaNet order
SRC_ORDER_DELETE_SQL_FAILURE	23004	Database error occurred while attempting to delete a VisaNet 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 VisaNet payment
SRC_PAYMENT_CREATE_SQL_FAILURE	24002	Database error occurred while attempting to create a VisaNet payment
SRC_PAYMENT_UPDATE_SQL_FAILURE	24003	Database error occurred while attempting to update a VisaNet payment
SRC_PAYMENT_DELETE_SQL_FAILURE	24004	Database error occurred while attempting to delete a VisaNet 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

Secondary Return Code	Value	Description
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 VisaNet credit
SRC_CREDIT_CREATE_SQL_FAILURE	25002	Database error occurred while attempting to create a VisaNet credit
SRC_CREDIT_UPDATE_SQL_FAILURE	25003	Database error occurred while attempting to update a VisaNet credit
SRC_CREDIT_DELETE_SQL_FAILURE	25004	Database error occurred while attempting to delete a VisaNet 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.

Secondary Return Code	Value	Description
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.
SRC_MALFORMED_URL	30016	Incorrect URL.
SRC_CONNECT_FAILED_TO_HOST	30017	URL connection to host failed.
SRC_VIRTUALNET_NOT_INITIALIZED	30018	Virtualnet not initialized.
SRC_BAD_BIN_OR_HOST_DISCONNECT	30019	Host Disconnected.
SRC_UNEXPECTED_GATEWAY_ERROR	30020	Unexpected Gateway Error.
SRC_UNEXPECTED_MIME_TYPE	30021	Incorrect MIME type.
SRC_INVALID_COMMUNICATIONS_PARAMETER	30022	Invalid Communications parameter.

Secondary Return Code	Value	Description
SRC_UNSUPPORTED_CONTENT_TYPE	30023	An unsupported content-type was received from the host.

Appendix B. Cassette for VisaNet messages

This chapter contains the Cassette for VisaNet-specific messages.

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 support 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 shut down.

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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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: Try to create the account again, but

specify 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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.

User Response: Contact the 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 support 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 support 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 support 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 support 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 support 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 support 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 support 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 support 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 support 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 support 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 WebSphere Commerce Payments database.

User Response: Ensure that the order exists for the 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce

Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 support 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 support 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 support 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 support 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 WebSphere Commerce Payments database.

User Response: Ensure that the credit exists for the merchant and the 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments machine and the database server.

CEPVisaNet8001: An SQL exception occurred while updating cassette configuration in the database.

Severity: Error

Explanation: An SQL exception occurred while updating a record in the WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce Payments 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 WebSphere Commerce

Payments 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 WebSphere Commerce Payments 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 IBM support representative.

CEPVisaNet0606: An internal error occurred: *exception text*.

Severity: Error

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

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

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
North Castle Drive
Armonk, NY 10504-1785
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 503
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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

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

- AIX
- AS/400
- IBM
- iSeries
- pSeries
- S/390
- WebSphere
- zSeries

Vital Processing Services is a registered trademark of Vital Corporation.

VisaNet is a registered trademark of Visa International Services Organization.

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.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

Glossary

This glossary defines technical terms used in the documentation of WebSphere Commerce Payments. The most current IBM Dictionary of Computing is available on the World Wide Web at <http://www.ibm.com/ibm/terminology/goc/gocmain.htm>.

A

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.

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 authorizes the transaction

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.

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.

authentication. (1) In computer security, verification that a message has not been altered or damaged. (2) In computer security, verification of the identity of a user or the user's eligibility to access an object. (3) The process of identifying an individual, usually based on a user ID and password. In security systems, authentication is distinct from authorization. Authentication merely ensures that the individual is who she claims to be; it does not define the access rights of the individual.

authorization. (1) 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. 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.

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.

batch. (1) A collection of payment transactions, such as captures, credits, capture reversals, and credit reversals, processed as a group. A batch is submitted as a single unit to the Acquirer's financial system. Business guidelines regarding the use of batch processing are developed by credit acquiring institutions. Merchants also establish policies that align

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

brand. Within IBM e-commerce, the Cassette object for all of the WebSphere Commerce Payments cassettes (for example, Cassette for VisaNet and Cassette for Paymentech). 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.

C

capture. The process by which the Acquirer receives payment from the customer's financial institution and remits the payment. A capture is the guarantee that the funds are available and that the transfer will take place.

card processor. An agent for an Acquirer to whom merchants send their transaction requests. The card processor provides much of the administrative and organizational infrastructure by which merchants process their transactions.

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

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, VisaNet and BankServACH) within the WebSphere Commerce Payments framework. Thus, WebSphere Commerce Payments is an extensible product that can support additional protocols.

certificate. (1) In computer security, 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 authority (CA) issues a certificate. (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

CGI program. A program that runs on a Web server and uses the common gateway interface (CGI) to perform tasks that are not usually done by the server, such as database access and form processing. The OS/400® operating system supports compiled CGI programs that are written in ILE C, ILE RPG, and ILE COBOL languages.

Clerk. In IBM e-commerce, this is a WebSphere Commerce Payments concept. WebSphere Commerce Payments has four different access rights. A clerk is defined on a per-merchant basis and has the lowest level of access.

client. (1) A functional unit that receives shared services from a server. For example, a personal computer requesting HTML documents from a Web server is a client of that server. (2) 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.

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

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.

credit. 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.

D

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

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.

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-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.

encryption. (1) 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. (2) The conversion of data into a form that cannot be easily understood so as to prevent unauthorized access, especially during transmission.

event. (1) A representation of a change that occurs to a part. The change enables other interested parts to receive notification when something about the part changes. For example, a push button generates an event by signalling that it has been clicked, which may cause another part to display a window. (2) 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.

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. (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.

F

financial institution. (1) 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. 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. If the reconciliation step fails, the batch is still not closed on WebSphere Commerce Payments (although it may be out of balance or not closed at the financial institution).

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.

G

gateway. A functional unit that connects a local data network to another network

H

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

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 on the Web.

I

installment payments. A type of payment transaction negotiated between the merchant and the cardholder which permits the merchant to process multiple authorizations.

integrity. In computer security, assurance that the information that arrives at a destination is the same as the information that was sent.

internet. (1) In TCP/IP, a collection of interconnected networks that functions as a single, large 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.

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.

issuer. (1) 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 application. Any deployable unit of J2EE functionality. This can be a single module or a group of modules packaged into an .ear file with a J2EE application deployment descriptor.

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

K

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

L

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

LUHN formula. An industry standard used by many credit card companies as a rudimentary prevention of credit card fraud.

M

merchant. 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. The merchant supervises the overall store objectives and management, in addition to tracking the store sales.

merchant bank. An Acquiring Financial institution. A merchant bank 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 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) 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.

N

number of credits. 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. 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

online catalog. General term for a collection of catalog groups or catalog entries available for display and purchase at an online store.

order. In WebSphere Commerce Payments, 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 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

payment. 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) 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) 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 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.

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). 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.

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.

private key. (1) In secure communication, an algorithmic pattern used to encrypt messages that only the corresponding public key can decrypt. The private key is also used to decrypt messages that were encrypted by the corresponding public key. The private key is kept on the user's system and is protected by a password (2) In computer security, a key that is known only to its owner.

public key. (1) In secure communication, an algorithmic pattern used to decrypt messages that were encrypted by the corresponding private key. A public key is also used to encrypt messages that can be decrypted only by the corresponding private key. Users broadcast their public keys to everyone with whom they must exchange encrypted messages. (2) In computer security, a key that is made available to everyone.

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

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. 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.

S

sale. 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.

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.

Secure Electronic Transaction. See SET Secure Electronic Transaction.

Secure Sockets Layer (SSL). A security protocol that allows the client to authenticate the server and all data and requests to be encrypted. The URL of a secure server protected by SSL begins with HTTPS (rather than HTTP).

server. (1) A functional unit that provides services to one or more clients over a network. (2) A computer that provides shared services to other computers over a network; for example, a file server, a print server, or 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 Secure Electronic Transaction™. An industry standard developed for secure credit card and debit card payments over open networks such as the Internet.

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 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.

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

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 proxy server that provides a secure one-way connection through a firewall to server applications in a nonsecure network. The server applications in the secure network must be compatible with the socket interface.

SSL. See Secure Sockets Layer.

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

T

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.

U

uniform resource locator (URL). The address of a file on the Internet. The URL contains the name of the protocol, the fully qualified domain name, and the path and file location.

URL. See uniform resource locator.

V

void payment. Within IBM e-commerce, a verb meaning to nullify or cancel a payment operation.

W

wallet. 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 browser. (1) Within IBM e-commerce, software running on the cardholder processing system that provides an 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 page. Any document that can be accessed by a uniform resource locator (URL) on the World Wide Web.

Web server. A server on the Web that serves requests for HTTP documents. The Web server controls the flow of transactions to and from WebSphere Commerce. It protects the confidentiality of customer transactions and ensures that the user's identity is securely transmitted to the WebSphere Commerce Server. The Web server implements the Secure Sockets Layer (SSL) protocol to achieve this level of security.

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.as400.ibm.com/icswg.html`, the Web site for IBM AS/400 is indicated by `www.as400.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.

X

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 write applications to handle document types, to author and manage structured information, and to transmit and share structured information across diverse computing systems. XML is defined by the World Wide Web Consortium (W3C).

Index

Numerics

3-D Secure support 43

A

AcceptPayment command 41
AccountAdmin object 13, 58, 59
accounts, creating 27
Address Verification Service 5, 61
administration objects 13
 AccountAdmin 13
 CassetteAdmin 13
 PaySystemAdmin 13
American Express CID 5
API command
 AcceptPayment 41
 Approve 43
 ApproveReversal 44
 BatchClose 44
 BatchOpen 44
 BatchPurge 44
 CassetteControl 44
 CloseOrder 45
 CreateAccount 45
 CreatePaySystem 45, 46
 DeleteBatch 46
 Deposit 46
 DepositReversal 47
 ModifyAccount 47
 ModifyCassette 48
 ModifyPaySystem 49, 50
 ReceivePayment 50
 Refund 50
 RefundReversal 50
approvals 31, 33
Approve command 43
ApproveReversal command 44
authorize, definition vii
authorizing cassettes 24

B

batch
 definition vii
 deleting 46
 settling 34
Batch object 56
Batch objects 14
batch totals
 viewing 36
Batch Totals report, Daily 36
BatchClose command 44
BatchOpen command 44
BatchPurge command 44

C

capture, definition vii
Cashier profiles 39

cassette features 5
Cassette for VisaNet
 description 1
 tutorial 21
 using SOCKS server 22
cassette properties
 AccountAdmin 58, 59
 Batch 56
 CassetteAdmin 57
 Credit 55
 Order 51
 Payment 53, 54
 PaySystemAdmin 60
cassette-specific parameters
 Approve 43
 Refund 50
CassetteAdmin object 13, 57
CassetteControl command 44
CEPVisaNet messages 71
certificate acceptance 25
CID verification 5
CloseOrder command
 delete option 45
codes, secondary return 63
command summary 16
communication methods 3
communication parameters 14
communications, external 22
concepts, payment 11
configuration steps 21
connectivity options 3
CreateAccount command 45
CreatePaySystem command 45
creating accounts 27
creating merchants 24
Credit object 55
Credit objects 14
credits
 definition vii
 issuing 35

D

Daily Batch Totals report 36
defining users 25
delete option, closeorder 45
DeleteBatch command 46
deposit 46
Deposit command 46
Deposit function 33
DepositReversal command 47

E

external communications,
 configuring 22

F

financial objects
 Batch 56
 Credit 55
 Order 51
 Payment 53
First Horizon Merchant Services
(FHMS) 1
flow diagram 1, 11
framework object
 extensions 57

I

icons 22
installation, cassette 19
IP gateways 3

K

keywords, command
 AcceptPayment 41
 Approve 43
 CreateAccount 45
 CreatePaySystem 45, 46
 Deposit 46
 ModifyAccount 47, 48
 ModifyCassette 48, 49
 ModifyPaySystem 49, 50
 Refund 50

L

leased-line connection 3
logging in 26

M

managing payment processing 28
maxRetries 57
merchant
 identification 5
 registration 4
merchant cassette settings 26
merchant settings 24
messages 71
model, object 11, 51
 implementation 13
ModifyAccount command 47
ModifyCassette command 48
ModifyPaySystem command 49, 50

N

notices 81

O

- object model 11
 - cassette 51
- object models
 - implementation 13
- object, framework
 - extensions 57
- objects
 - administration 13
 - AccountAdmin 13
 - CassetteAdmin 13
 - PaySystemAdmin 13
 - administrative
 - AccountAdmin 58, 59
 - CassetteAdmin 57
 - PaySystemAdmin 60
 - financial 51
 - Batch 14, 56
 - Credit 14, 55
 - Order 14, 51
 - Payment 14, 53, 54
 - payment 14
- Order objects 14
 - XML example 52
- orders
 - approving with Sale function 30
 - creating 28
 - details about 31

P

- parameters
 - communication 14
 - protocol 41
- payment objects
 - Batch 14
 - Credit 14
 - Order 14
 - Payment 14
- Payment objects 14
 - XML example 54
- payment processing
 - implementation 11
 - managing 28
 - tutorial tasks 21
- PaySystemAdmin object 60
- profiles, Cashier 39
- properties, cassette
 - AccountAdmin 58, 59
 - Batch 56
 - Credit 55
 - Order 51
 - Payment 53, 54
 - PaySystemAdmin 60
- protocol parameters 41
- PSApiResult document 51
- publications, related viii
- purchase example 11
- purchasing cards 6

Q

- query call 51

R

- read timeout 57
- ReceivePayment command 50
- Refund command 50
- RefundReversal command 50
- report, Daily Batch Totals 36
- retry parameters 57
- return codes, Cassette for
 - Visanet-specific 63
- reversal, deposit 47
- role assignment 8

S

- Sale function 33
 - approving orders using 30
- Sample Checkout application 29, 40
- secondary return codes 63
- sensitive data protection 7
- settling batches 34
- SSL gateways 3
- SSL support, iSeries 25
- starting the Payments interface 22
- state changes, object 16

T

- tasks, payment processing 21
- terminology vi
- Totals report, Daily Batch 36
- trademarks 82
- truststore, default 25
- tutorial 21

U

- user interface, Payments 22
- user roles 25
- users, defining 25

V

- VisaNet Batch object 56
- VisaNet Credit object 55
- VisaNet Order object 51
- VisaNet Payment object 53
- VisaNet, Cassette for
 - SOCKS server 22
 - tutorial 21
- Vital Processing Services 1

W

- wcsadmin user 25
- Web sites viii



Printed in U.S.A.