IBM WebSphere Commerce

Payments Cassette for Paymentech Supplement

Version 5.5

IBM WebSphere Commerce

Payments Cassette for Paymentech 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 75.

Third 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. Make sure you are using the correct edition for the level of the product.

Contents

About this book

This book is for users and administrators of the Cassette for Paymentech 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.

Windows indicates information specific to the Windows® operating environment.

AIX indicates information specific to AIX®.

Solaris indicates information specific to the Solaris Operating Environment.

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

Linux 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® systems unless otherwise specified.

`WC_installdir` represents the following default installation paths for WebSphere Commerce:

AIX /usr/lpp/WebSphere/CommerceServer*nn*

Linux Solaris /opt/WebSphere/CommerceServer*nn*

Windows *drive*:\WebSphere\CommerceServer*nn*

400 /QIBM/ProdData/CommerceServer*nn*

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

AIX /usr/lpp/WebSphere/CommerceServer*nn*/payments

Linux Solaris /opt/WebSphere/CommerceServer*nn*/payments

Windows *drive*:\WebSphere\CommerceServer*nn*\payments

400 /QIBM/ProdData/CommercePayments/V*nn*

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 the following:

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/wscom/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).

Paymentech processing services are described in the following documents:

- *Paymentech Payment Processing: 96-Byte Technical Specification (revision 1.7.1)*
- *Paymentech On-Line Processing: Technical Specification (revision 6.0.1)*

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

Chapter 1. Overview of Paymentech

Paymentech, Inc., formerly known as First USA Merchant Services, is a merchant acquirer based out of Salem, NH. In 1999, Paymentech became the nation's second largest merchant processor based on transactions when it combined with Bank One Payment Services. Paymentech manages its own direct links to Visa and MasterCard, and it operates under its own specific Bank Identification Number and Interbank Card Association Number.

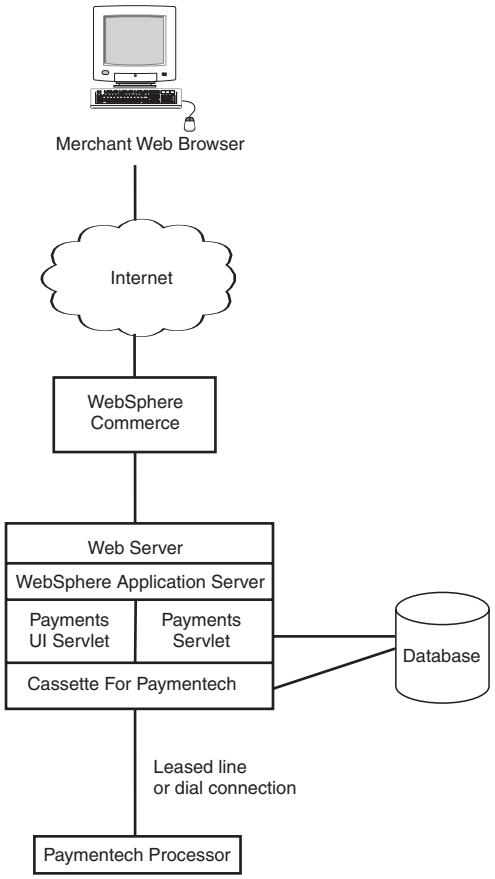

Figure 1. High-level overview of Cassette for Paymentech processing

Paymentech adopts a total system approach that offers speed, security, flexibility and reliability, which are all based upon the specific merchant needs. For example, Paymentech's comprehensive array of products and services include the following:

- Credit and debit authorization and settlement
- T&E card authorization
- International currency processing for direct response
- Canadian and US electronic check processing (direct debit)
- Real-time authorization service available twenty four hours a day
- Reporting and information management

Figure 1 on page 1 shows the major components involved with the Cassette for Paymentech in a WebSphere Commerce Payments environment.

How Paymentech works

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 settlement in a secure environment.

Through the use of relational databases, Paymentech determines the need for authorization reversals, either full or partial, to ensure the most affordable interchange rates. In cases where a new authorization is required, Paymentech attempts these transactions prior to the reversal of the original authorization assuring that the deposited transaction will always bear a valid authorization.

Paymentech processes transactions through Visa, MasterCard, American Express, and Novus (Discover). This allows Paymentech to provide authorization services for Visa, MasterCard, American Express, Discover, Novus, Diners, Carte Blanche, JCB and Private Label.

Paymentech offers a range of fraud detection mechanisms, and thereby, ensures that all of their merchants fully adhere to Visa and MasterCard CPS/DM (Custom Payment Services/Direct Marketing) Compliance. Merchants, therefore, are eligible for a lower interchange rate.

Paymentech merchant registration

Paymentech assigns merchants specific settings based on account data (such as connectivity, requested Paymentech functionality and system configuration) that the merchant provides. For example, Paymentech issues IDs and passwords which are used by the merchant when submitting batch files to Paymentech. These settings include the Presenter Identification (PID) and password and the Submitter Identification (SID) and password. These settings are specified in the Cassette for Paymentech via the cassette's Merchant Cassette settings.

Paymentech issues division numbers for each type of currency that the merchant desires to support. Division numbers are used to delineate business units for purposes of financial reporting. An unlimited number of merchant division numbers can be assigned to organize your various divisions in any combination. Division numbers are unique to each merchant and are mapped to their unique (PID). In addition, Paymentech uses the division number as one criteria to identify duplicate transactions.

For more information on merchant registration, consult a Paymentech Technical Support representative by contacting operations at (800) 228-7782.

Cassette features

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 Paymentech enables users of WebSphere Commerce Payments to access Paymentech's Salem gateway through a leased-line connection.

It provides online authorization and settlement of credit card and non-PIN based debit card payments. The Cassette for Paymentech provides merchants with the ability to send real-time Internet credit card transactions to the Paymentech system for processing. In addition, the Cassette for Paymentech supports prior authorization detection, Address Verification Service (AVS) and Card Verification Codes (CVC) to ensure data integrity and validation.

Prior Authorization Detection

To prevent double authorization, the Cassette for Paymentech supports Prior Authorization Detection. If a telecommunications line fails, the Paymentech system has a record of what occurred prior to the disconnection and reapplies the authorizations where appropriate. Hence, inadvertent authorization inflation against the customer's available credit line can be prevented.

Address Verification Service

Paymentech supports AVS as a major tool to screen and combat fraud for Visa, MasterCard, American Express and Discover. AVS matches the cardholder's billing address to the address given to you, thereby providing a method to identify potentially fraudulent transactions. This service is available as part of the authorization process.

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 Paymentech 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. Paymentech supports Visa, MasterCard, and Discover AVS, and American Express's Automated Address Verification (AAV).

If you are interested in additional information regarding AVS and merchant chargeback liabilities, contact your acquiring financial institution. See the "Address Verification Service (AVS) result codes" on page 56 for more information on possible AVS codes.

Card Verification Codes

In addition to AVS, the cassette for Paymentech supports CVC, which provide additional fraud detection. Both Paymentech and the Cassette for Paymentech support Visa's Card Verification Value Card 2 (CVV2), MasterCard's Card Validation Code 2 (CVC2), American Express Cardholder Identification Code (CID), and Discover CID.

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 Paymentech:

Table 1. Sensitive data processed by Cassette for Paymentech

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

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 role. 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	• 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	• Define Merchant Administrators, Supervisors, and Clerks • Configure merchants and their cassettes • Add, delete, and update event listeners
Supervisor	• Settle payments • Approve or sale orders • Issue credits and reverse credits • Deposit orders • Search for orders and batches • View daily batch totals
Clerk	• Settle payments • Approve or sale orders • Deposit orders • Search for orders and batches • View daily batch totals

Chapter 2. Paymentech 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 Paymentech presents the Paymentech services through the WebSphere Commerce Payments' object model and API set. In addition, cassette-specific behaviors and requirements are discussed.

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

- AcceptPayment creation of orders only. Wallet-driven purchases are not supported.
- Traditional payment-oriented commands.

A Paymentech purchase example

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

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

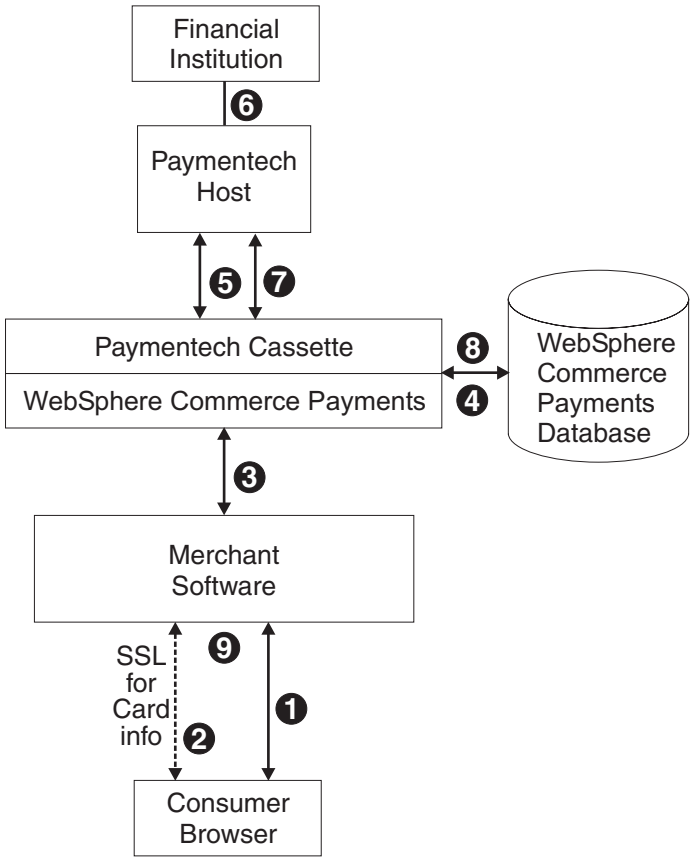

Figure 2. A typical purchase using the Cassette for Paymentech

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. WebSphere Commerce Payments and the Cassette for Paymentech record the information they need to execute payment transactions.
5. The Cassette for Paymentech sends an approve request to the Paymentech host.
6. The Paymentech host forwards this request to the financial institution which processes the request and responds to the Paymentech host.
7. The Paymentech host records the result, and sends a success response to the cassette.
8. The Cassette for Paymentech, along with WebSphere Commerce Payments, updates status in the database and returns the success response to the merchant.

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

A Paymentech batch settlement process example

The following example illustrates the online batch settlement process that is utilized by the Cassette for Paymentech:

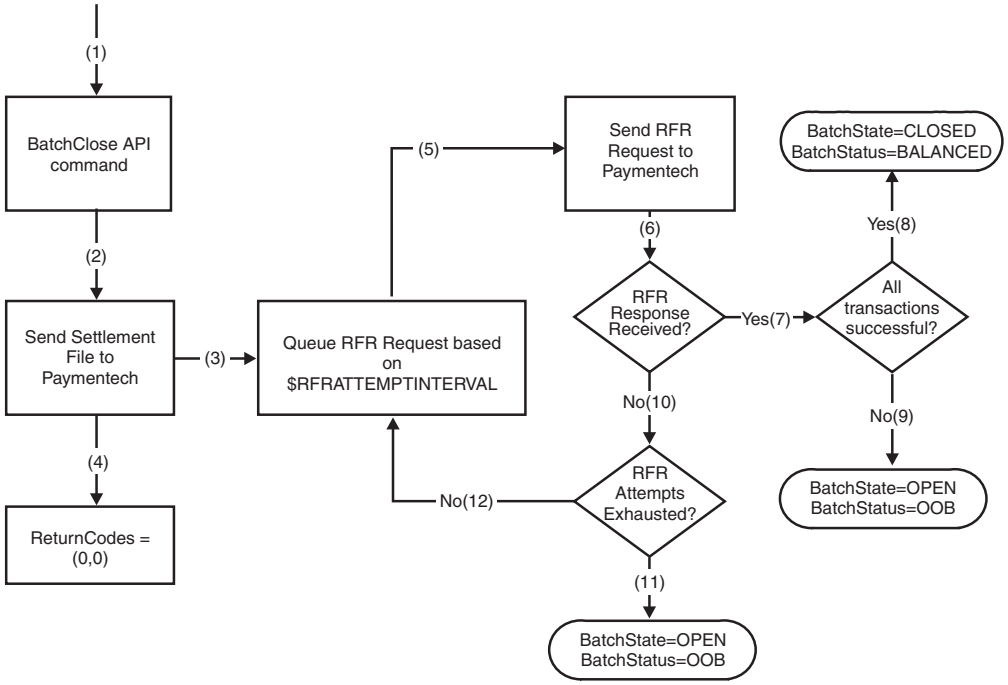

Figure 3. Batch settlement process for the Cassette for Paymentech

1. Merchant server software issues the BatchClose API command when it wishes to initiate the settlement process.
2. The Cassette for Paymentech sends the settlement file to Paymentech.
3. Based on the $RFRDELAYTIME value that is configured in the Cassette's configuration, a Request For Response (RFR) is queued and waits until the RFR is scheduled to be sent to Paymentech. For example, if $RFRDELAYTIME is 10, then the Cassette for Paymentech will not issue the first RFR request until 10 minutes after the BatchClose API command was issued.
4. Provided there were no communications failures while sending the settlement file, the Cassette for Paymentech returns successful return codes (0,0) to the Merchant Server Software.
5. Once the RFR Delay Time has passed, the Cassette for Paymentech sends an RFR request to Paymentech.
6. Provided there were no communications failures, some type of response is received from Paymentech (either the "No Data to send back at this time" response, or the actual RFR Response).
7. If an actual RFR Response is received, the Cassette for Paymentech looks at the results of each individual transaction in the batch.
8. If all transactions in the batch were successfully settled, then the WebSphere Commerce Payments batch is marked as **CLOSED** and **BALANCED**.

9. If all transactions in the batch were not successfully deposited, then the WebSphere Commerce Payments batch is marked as **OPEN** and **OUT_OF_BALANCE**. At this point, the merchant must contact Paymentech to resolve the problems with the batch.
10. If the ″No Data to send back at this time″ response is received (i.e., an actual RFR Response was not received), then the Cassette for Paymentech determines if all RFR Requests have been attempted. This is based on the $RFRATTEMPTS value that is configured in the Cassette's configuration. For example, if $RFRATTEMPTS is 3, then the Cassette for Paymentech issues a maximum of 3 RFR Requests to Paymentech.
11. If the RFR attempts have been exhausted, then the WebSphere Commerce Payments batch is marked as **OPEN** and **OUT_OF_BALANCE**. At this point, the merchant must contact Paymentech to resolve the problems with the batch.
12. If the RFR attempts have not been exhausted, then a Request For Response (RFR) is queued and waits until the RFR is scheduled to be sent to Paymentech. At this point, the process repeats itself starting back at step 5.

WebSphere Commerce Payments object model implementation

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

Administration objects

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 Paymentech augments four of the framework administration objects with its own attributes. Paymentech Administration objects are described in detail in Chapter 7, “Object reference”, on page 51.

CassetteAdmin object

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

AccountAdmin 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 Paymentech merchant account represents. The cassette extends the WebSphere Commerce Payments AccountAdmin object with attributes that identify and describe the corresponding AccountAdmin merchant account. Only one account per merchant can be defined.

PaySystemAdmin 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.

MerchantCassette object

The MerchantCassette object is an administrative object that contains properties that apply only to one merchant. MerchantCassette objects are identified through a combination of the merchant number, object name and cassette name. In the

Cassette for Paymentech, these objects represent the merchant's divisions and corresponding currency values, as assigned by Paymentech.

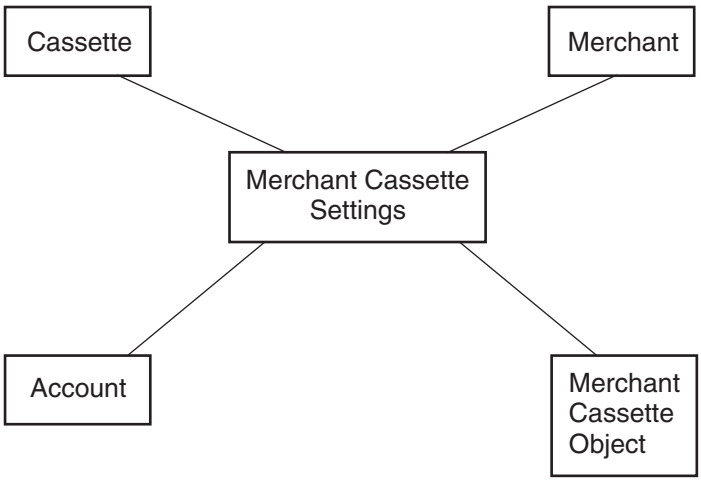

Figure 4. Paymentech PaySystem

Financial objects

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

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

For details on how the Cassette for Paymentech extends these financial objects, see Chapter 7, "Object reference", on page 51. For descriptions of the financial 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 Paymentech.

Retry parameters

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

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

You can modify any of the Cassette Settings values through the user interface (select Cassettes under the navigation frame, then select the Paymentech cassette icon, then select Advanced Settings) or through the ModifyCassette API command. For information on the ModifyCassette command, see "ModifyCassette" on page 48. The Cassette for Paymentech will attempt to recover communications failures under the direction of the above parameters.

For each command that requires communication with the Paymentech gateway, a connection must be established between WebSphere Commerce Payments and the Paymentech gateway. Once this connection is established, the Cassette for Paymentech 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:

1. The cassette will return a return code that indicates the operation is pending (PRC_OPERATION_PENDING).
2. The request message is queued and waits a predetermined amount of time as specified by the cassette setting called **Attempt Interval**.
3. Once the attempt interval expires, the request is removed from the internal queue and is retried.
4. 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 **15 seconds** or less.
- The RFRDelayTime should be specified as a minimum of **10 minutes**.

Cassette for Paymentech payment command summary

Table 4 on page 13 summarizes the way the Cassette for Paymentech 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 RC_COMMAND_NOT_SUPPORTED, RC_NONE.
  - "Handled by WebSphere Commerce Payments; no message sent" means that the command is processed completely within WebSphere Commerce Payments without communicating with a Paymentech host.
  - In any other case, the primary Paymentech command (or commands) used to accomplish the function will be shown.

Table 4. Cassette for Paymentech. Summary of Payment API Commands

API command	Payment card functions	Paymentech 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	Handled by WebSphere Commerce Payments; no message sent
BatchClose	Close an existing batch	Settlement file sent to Paymentech
BatchOpen	Open a new batch	Not supported by cassette (cassette opens batches internally as needed)
BatchPurge	Purge an existing batch	Not supported by cassette
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. Summary of API command 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
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 Paymentech 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 Paymentech Version 5.5 with earlier versions of the WebSphere Commerce Payments framework.

Before you can configure the Cassette for Paymentech, 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 Paymentech 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 Paymentech. 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 Paymentech 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 Paymentech:

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 a Paymentech division.
9. 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:
10. Approve orders and deposit payments.
11. Settle batches.
12. Issue credits.
13. View daily batch totals.

Before starting this tutorial

There are a number of configuration steps that require information from your Paymentech 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 Paymentech. 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

During this tutorial, your WebSphere Commerce Payments will attempt to send messages to the Paymentech System. To tailor the way the cassette communicates with the Paymentech System, do the following:

1. Click on **Cassettes** in the navigation frame.
2. Click on the **Paymentech** cassette icon. At the next page, you will see several entry fields. Enter the following:

Table 6. Cassette for Paymentech page settings

Field	Description
Online IP Address	The Paymentech IP Address used for online authorizations.
Online Port Number	The port number used for online authorizations.
Batch IP Address	The Paymentech IP Address used for batch settlement.
Batch Port Number	The port number used for batch settlement.

3. Click **Update** to update your cassette configuration.
4. Click **Stop Cassette** to stop the cassette.
5. Click **Start 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:

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 *Paymentech*. Checking this box authorizes the merchant to use the Cassette for Paymentech.

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 **Paymentech**.
4. Click **Update**.

The merchant is now authorized to use the cassette.

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 7. 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

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

Note: The following merchant settings are assigned by Paymentech. For more information on data assigned by Paymentech, see "Paymentech merchant registration" on page 2.

Table 8. Merchant cassette settings

Field	Description
Presenter's ID (PID)	This is the assigned Paymentech Presenter's Identification Number (PID).
PID Password	This is the assigned Paymentech Presenter's Password (PID Password).
Submitter's ID (SID)	This is the assigned Paymentech Submitter's Identification Number (SID).
SID Password	This is the assigned Paymentech Submitter's Password (SID Password).

Step 7: Creating an account

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

An account is a relationship between the merchant and the financial institution which processes transactions for that merchant. In the Cassette for Paymentech, there can be only one account defined for each merchant.

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 Paymentech** icon in the Test Store.
3. From the Cassette for Paymentech page, click **Accounts**.
4. From 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 9. Add an account settings

Field	Description
Account name	Enter *Paymentech Account*. 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 *1*. 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 *Paymentech Bank*. 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.
Batch Close Time	Enter *1830*. This is an optional parameter that enables the Merchant to specify a time when any currently open and online batches will be automatically closed on a daily basis. The format is HHMM, where HH represents the hour and MM represents the minutes. This value should be specified in 24-hour time. For example, when 1830 is specified, then at 6:30 p.m. all open batches for that account will be automatically closed. If the BatchCloseTime is not specified, all batches must be manually closed via the BatchClose API command.

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

Step 8: Creating a Paymentech division

Paymentech assigns a unique division number to process each currency. A single division number can support different transaction types and methods of payment in the same currency. All transaction types and currencies may be submitted in the same file for authorization or settlement, but the correct division number and currency code must accompany the corresponding transaction.

To create a Paymentech division:

1. From the navigation frame click **Merchant Settings**.
2. From the Merchant Settings page, click the **Cassette for Paymentech** icon in the Test Store.
3. From the Cassette for Paymentech page, click **Paymentech Division Settings**.
4. Click **Add a Division**.
5. Complete the following fields:

Field	Description
Division Number	Enter the Paymentech assigned division number.

Field	Description
Currency Code	Enter *840*, which represents the US Dollars currency. This value also represents the ISO currency code associated with this division.

6. Click **Create** to create the new division.

Step 9: 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 Paymentech Cassette 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 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 the italicized text or fields listed as required *must* be entered for the tutorial):

Table 10. Sample Checkout fields for Cassette for Paymentech

Field	Description
Merchant number	Enter *123456789*, the number used when creating the merchant, to represent a Merchant number. (Required)
Order number	Enter any unique number to represent an Order number. (Required)
Amount	Enter *25* to represent the total numeric amount of the order. (Required)
Currency	Select *US dollar*. The currency used to place this order. (Required)
Payment method	Choose *Paymentech* as the payment method. (Required)
Card number	Enter the credit or debit card number used to place the order. This is a 10–19 digit string. (Required)
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 security presence	Enter the indicator used to validate the presence of a card security value. The following values are valid: **1**–Value is present **2**–Value on card, but illegible **9**–Cardholder states the card has no card security value. **Note:** Currently, these values only apply to Visa and Discover.
Card verification value	Enter a 3–4 digit number to represent the verification code printed on the signature panel of the card.

Table 10. Sample Checkout fields for Cassette for Paymentech (continued)

Field	Description
Transaction type	Enter the transaction type that specifies the type of transaction. The following values are valid: **2**–Recurring Transaction: Designates a transaction that represents an arrangement between a cardholder and the merchant where transactions are going to occur on a periodic basis. **3**–Installment Payment: Designates a group of transactions that originated from a single purchase where the merchant agrees to bill the cardholder in installments. **7**–Non-SET Transaction Channel Encrypted: Designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction does not include the management of any SET™ certificates, but does include the use of transaction encryption such as SSL. **8**–Non-Secure Electronic Commerce Transaction: Designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction does not include the management of any SET certificates and does not include the use of any transaction encryption such as SSL.
Settlement Mode	Enter the indicator used to specify how the batch will be settled: **0**–Batch should be settled online **1**–Batch should be settled offline
Cardholder name	Enter the name of the credit or debit card holder.
Street address	Enter the cardholder's street address.
Street address 2	Enter any additional street address information.
City	Enter the current city of the cardholder.
State/Province	Enter the current state of the cardholder.
Zip code	Enter the current zip code of the cardholder. (Required)
Country code	Enter the 2–character string country code.
Phone number	Enter the current phone number of the cardholder. This is a 10– or 14– character string value in the format AAAEEENNNNXXXX or AAAEEENNNN, where AAA = Area Code EEE = Exchange NNNN = Number XXXX = Extension
Phone type	Enter the cardholder's phone type. This is a 1–character string value, which can be one of the following values: D-Day H-Home N-Night W-Work

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 10: Approving orders with the Sale function

The Sale function allows you to approve and deposit payments with one command. (The Cassette for Paymentech 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 function, 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 page, you can view order details. Click **Sale** to approve and deposit this order.
3. The Order Sale window displays the following information:

Table 11. Order sale 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 12. 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 (Paymentech Account).
Order URL	A link to a fictitious order description. In a real merchant's online shopping system, this field might be filled in to point to a corresponding entry in the order entry database.
Time Created	The time that this payment was created.
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 (Paymentech).
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 Paymentech.
Auth Response Code	Paymentech response reason code which indicates the status of the authorization request. An authResponseReason of *100* indicates approval. See the Paymentech Online 6.0.1 specification for a complete list of all Response Reason Codes.
Auth Response Date	The format for this value is YYMMDD.
Auth Code	This represents the Authorization Code that the issuer uses to show an authorization request was approved. This is a 1-6 character string.
AVS Response Code	Specifies the response to address verification request. For more information, see "Address Verification Service (AVS) result codes" on page 56.

Table 12. Payment details (continued)

Field	Description
CVV2 Response Code	CVV2 Result code returned by the card issuer in response to a CVV2/CVC2 request. Valid values are: M-Value matched N-Value not matched P-Not processed S-Should be on the card (Discover, Visa only) U-Unsupported by the issuer I-Invalid (Master, Visa Only) " "-Blank if not Discover, MasterCard, Visa
Merchant Order Number	A 1-8 digit number generated by the Cassette for Paymentech that uniquely identifies the transaction.
Payment Settled?	**Y** or **N** indicator that indicates if payment was successfully settled. For Offline settlement transactions, this will always be **N** since the transaction is never settled within WebSphere Commerce Payments.

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 Paymentech 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", 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 26. 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 11: 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 purposes of this tutorial, you can fill in either the payment type: *Paymentech* or the account.

Table 13. 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.

Table 13. Batch search fields (continued)

Field	Description
State	The state of the batch: • Open • Closed
Status	The balance status of this batch: • **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 **Paymentech**.
Batch Open Date	Use the *after* and *before* fields below to search for batches opened during the specified range in time: • **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 *before* and *after* fields below to search for batches closed during the specified range in time: • **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 (Paymentech Account). If more than 500 accounts have been defined, type the account number in the entry field.

3. Click **Search** to initiate a batch 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. From the Batch window, you can view useful batch information, including the total number and amount of both payments and credits in the batch.
6. Click **Settle** to settle the batch. When processing is complete, the settle status is displayed. This causes the settlement file to be sent to Paymentech, and thereby initiating the Request for Response (RFR) process. For an example of how the complete batch settlement process works, see "A Paymentech batch settlement process example" on page 9.
7. To view the status of the RFR request, view the Batch Details from the Batch window, as described in step 5. Check the **RFR Status** field, which displays the status of the RFR request.

Step 12: 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):

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: • Requested • Ordered • Refundable • Canceled • Closed
Payment Type	Identifies the payment cassette or protocol used to place the order. Select **Paymentech**.
Order Date	Use the *after* and *before* fields below to search for orders opened during the specified range in time: • **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	• **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 Paymentech account under which this order is being processed. If more than 500 accounts have been defined, type the account number in the entry field.

3. Click **Search**.
4. From the next 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 14. 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.

7. 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**.

Step 13: 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** and specific merchant 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 screen. 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 Paymentech).

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 WebSphere Commerce 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 system and demonstrates fundamental payment processing implemented through the Cassette for Paymentech. For more information on specific fields in the WebSphere Commerce Payments user interface, see the online help.

Chapter 5. Cassette for Paymentech 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 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 Paymentech:

- `WC51_Paymentech.profile`

  This profile can be used with the WebSphere Commerce sample stores, or in a production environment. If you used the default instance name of **demo**, the profile is stored in the following directory: *WC_installdir*/instances/demo/xml/payment.

- `SampleCheckoutPaymentech.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 Paymentech profile be named `SampleCheckoutPaymentech.profile`.

  Initially, the `SampleCheckoutPaymentech.profile` is installed in directory path *Payments_installdir*/cassettes/Paymentech/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 *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.

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 35.) 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 Paymentech-specific protocol parameters
- Any special notes related to the Cassette for Paymentech handling of framework parameters

Note: For any framework commands that are not listed here, there are no specific Paymentech parameters or unique behaviors. See the *WebSphere Commerce Payments Programming Guide and Reference* for a complete list of generic framework commands.

Cassette for Paymentech commands

The following section outlines information specific to the Paymentech 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 Paymentech 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 a generic payment and a Paymentech cassette payment are created, and an Authorization message is sent to Paymentech. If the DepositFlag is set to **1**, the Payment is added to the currently open batch for the specified merchant (if there is no open batch for the merchant, one will be created implicitly). An ApproveFlag of **2** indicates that the transaction should be approved asynchronously. See the *WebSphere Commerce Payments Programming Guide and Reference* for more information on Asynchronous Auto Approve.

After executing the AcceptPayment command, the Order object moves to the REFUNDABLE state. For more information on the Payment object's states for cases where the auto-approve and or auto-deposit flags are set, see the Approve and Deposit commands.

Table 15. Required and optional keywords for AcceptPayment command

Keywords	Type	Value
AMOUNT	Required	Transaction amount. Specify without decimal point. For example, input 1234 for $12.34. For Paymentech-supported Visa and MasterCard transactions, the amount must be greater than $.01 and less than or equal to $25,000 (or the international currency equivalent). For all other brands, the amount must be greater than $1.00 and less than or equal to $25,000 (or the international currency equivalent). (Min/Max: 0 - 2147483647)
CURRENCY	Required	Indicates the currency code for the transaction. Paymentech supports the following currencies: 124–Canadian Dollars 036–Australian Dollars 826–British Pounds Sterling 208–Danish Krone (Krona) 300–Greece Drachma 344–Hong Kong (Dollars) 554–New Zealand Dollars 578–Norwegian Krone (Krona) 702–Singapore Dollar 710–South African Rand 752–Sweden (Swedish) Krona 756–Swiss (Switzerland) Franc 392–Japanese Yen 840–U.S. Dollars 978–Euro (Min/Max: 3 digits)
PAYMENTTYPE	Required	The name specified must be **Paymentech**.
BATCHNUMBER	N/A	Not allowed (must not be specified) since all batches are opened implicitly.
$PAN	Required	Specifies card account number (credit card number). Value must pass the Luhn check to be considered valid. (Min/Max: 10-19 digits)
$EXPIRY	Optional	Specifies card expiration date, using the format YYYYMM. (Min/Max: 200001-999912)
$CARDVERIFYCODE	Optional	Amex CID, Visa CVV or MasterCard CVC number. (Min/Max: 3-4 digits)

Table 15. Required and optional keywords for AcceptPayment command (continued)

Keywords	Type	Value
$CARDSECURITYPRESENCE	Optional	This indicator is used to validate the presence of a card security value. The following values are valid: **1**–Value is present **2**–Value on card, but illegible **9**–Cardholder states the card has no card security value. **Note:** Currently, these values only apply to Visa and Discover.
$CARDHOLDERNAME	Optional	Specifies the Bill To or the Order initiator's name (Min/Max: 0-30 characters)
$AVS.STREETADDRESS	Optional	Street address. (Min/Max: 0-30 characters)
$AVS.STREETADDRESS2	Optional	Specifies additional street information. (Min/Max: 0-28 characters)
$AVS.CITY	Optional	Specifies the city name of the cardholder's address. (Min/Max: 0-20 characters)
$AVS.STATEPROVINCE	Optional	U.S. state abbreviation. The valid values are: AL, AK, AZ, AR, CA, CO, CT, DE, DC, FL, GA, HI, ID, IL, IN, IA, KS, KY, LA, ME, MD, MA, MI, MN, MS, MO, MT, NE, NV, NH, NJ, NM, NY, NC, ND, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VT, VA, WA, WV, WI, WY (Min/Max: 2 characters)
$AVS.POSTALCODE	Required	Specifies the cardholder's postal code. (Min/Max: 5–9 characters)
$AVS.COUNTRYCODE	Optional	This is the ISO country code. Paymentech currently supports the following values: US–United States CA–Canada GB–Great Britain (Min/Max: 2 characters)
$AVS.PHONENUMBER	Optional	AVS telephone number in the following format: AAAEEENNNNXXXX or AAAEEENNNN, where AAA = Area Code EEE = Exchange NNNN = Number XXXX = Extension If the specified telephone number is 10 characters in length, then AAAEEENNNN format is assumed. If the specified telephone number is 14 characters in length, then AAAEEENNNNXXXX format is assumed. (Min/Max: 10 or 14 characters)

Table 15. Required and optional keywords for AcceptPayment command (continued)

Keywords	Type	Value
$AVS.PHONETYPE	Optional	Telephone type. Valid values are: D - Day H - Home N - Night W - Work
$TRANSACTIONTYPE	Optional	Specifies the type of transaction. Valid values are: **2**—Recurring Transaction: designates a transaction that represents an arrangement between a cardholder and the merchant where transactions are going to occur on a periodic basis. **3**— Installment Payment: designates a group of transactions that originated from a single purchase where the merchant agrees to bill the cardholder in installments. **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. **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. **7**—Non-SET Transaction Channel Encrypted: designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction does not include the management of any SET certificates, but does include the use of transaction encryption such as SSL. **8**—Non-Secure Electronic Commerce Transaction: designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction does not include the management of any SET certificates and does not include the use of any transaction encryption such as SSL. (Min/Max: 1-8) See also the notes that follow.

Table 15. Required and optional keywords for AcceptPayment command (continued)

Keywords	Type	Value
$TRANSACTIONTYPE	Optional	**Notes:** 1. The setting of this value can affect the interchange rate of the transaction. 2. For recurring and installment transactions, Paymentech recommends that the merchant specify a transaction type of either 7 or 8 for the first transaction, as appropriate, and then specify a transaction type of either 2 or 3, respectively, for subsequent recurring/installment transactions. 3. Paymentech allows the Merchant to set up a default Transaction Type to be specified at the division level. The value specified for $TRANSACTIONTYPE will override any default set by the Merchant at the division level.
$SETTLEMENTMODE	Optional	Specifies the settlement mode of the transaction. Valid values include the following: **0**—Online settlement: designates a transaction that is settled through the Paymentech processor when BatchClose command is received. **1**— Offline settlement: designates a transaction in which settlement occurs outside the scope of Paymentech. The merchant is responsible for settling these transactions. If not specified, the settlement mode will default to **0**, online settlement. (Min/Max: 0-1)
$VISA_CAVV		The cardholder authentication verification value used for credit card authorization when 3-D Secure support is enabled. Consists of 28-byte value (a 20-character alphanumeric value that is Base64 encoded for a 28-byte result.) (The cassette performs the necessary decoding.) If this keyword is specified, $VISA_XID should also be specified.

Table 15. Required and optional keywords for AcceptPayment command (continued)

Keywords	Type	Value
$VISA_XID		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.) If this keyword is specified, $VISA_CAVV should also be specified.

Approve

The Approve command causes a generic payment and Paymentech 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.

If the authorization command succeeds, then the Payment object will be in the APPROVED state. If the command fails because an error ResponseCode is returned from Paymentech, then the Payment state will be DECLINED. If the authorization is never successfully delivered to Paymentech or a response never arrives from Paymentech, then the Payment state will be VOID.

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

Table 16. Cassette-specific parameters processing of parameters for the Approve command

Field	Type	Value
PAYMENTTYPE	Required	Specified name must be **Paymentech**.
AMOUNT	Required	Specified payment amount. Input this value without the decimal point. For example, input 1234 for $12.34. For Paymentech supported Visa and MasterCard transactions, the amount must be > $.01 and <= $25,000 (or the international currency equivalent). For all other brands, the amount must be > $1.00 and <= $25,000 (or the international currency equivalent). (Min/Max: 0 - 2147483647)
BATCHNUMBER	Not Allowed	Not allowed. Must not be specified because all batches are opened implicitly.

ApproveReversal

Authorization Reversals do not result in a message going to the Paymentech processor. This is due to the fact that Paymentech handles partial approve reversals automatically by reversing any unused approved amount at settlement time. For example, if Paymentech receives (and approves) an Authorization message for $100 and, at batch settlement time, only $75 are being deposited for that transaction, then Paymentech will automatically reverse the difference ($25).

Full authorization reversals are accomplished by never attempting to deposit the previously approved transaction; thereby, leaving the prior authorization to expire.

Note: Only the VISA brand supports partial authorization reversals. No issuers support full authorization reversals.

The Paymentech cassette implements Approve Reversals as follows:

- Partial reversal (AMOUNT is less than or equal to the original Approval amount) — The cassette changes the approval amount associated with the payment to ensure that it contains the new amount when the payment is deposited.
- Full reversal (AMOUNT is 0) — Cassette places the payment into VOID state. Thus, the payment does not appear in the settlement file, and the reservation (open-to-buy) against the card is left to expire.

Table 17. Required keywords for ApproveReversal command

Field	Type	Value
PAYMENTTYPE	Required	Specified name must be **Paymentech**.

BatchOpen

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

- PRC_COMMAND_NOT_SUPPORTED
- RC_NONE

BatchPurge

In general, since Paymentech does not support resubmitting a Batch file when a settlement error occurs, the BatchPurge command is not supported. Thus, if this command is issued with the PAYMENTTYPE parameter value specified as Paymentech, then the command will fail with the following return code: PRC_COMMAND_NOT_SUPPORTED and RC_NONE.

BatchClose

The Cassette for Paymentech offers two different behaviors for the BatchClose command. They include:

- Online settlement—The BatchClose command will result in the payments and credits being retrieved from the database, and a settlement file created and sent to Paymentech for processing.
- Offline settlement—The BatchClose command automatically marks the batch (and it's associated payments/credits) as CLOSED without sending a settlement file to Paymentech. The merchant can query the contents of the batch using the QueryBatches command, and do settlement via a process outside of WebSphere Commerce Payments.

As a result of the two different types of settlement, each merchant can have a maximum of two open and active batches at any given time—one batch that contains transactions to be settled via Paymentech (Online batch), and one batch that contains transactions to be settled outside the scope of WebSphere Commerce Payments (Offline batch). Transactions are placed in the appropriate batch based on the $SETTLEMENTMODE protocol data that is passed on the AcceptPayment command.

BatchClose command when the batch is marked as Online

The Cassette for Paymentech utilizes Paymentech's Request For Response (RFR) process for settling Online batches. The RFR process works as follows:

1. The settlement file is sent to Paymentech (when the BatchClose API command is issued). There is no response returned.
2. At a predetermined time (recommended to be 10 minutes), an RFR request is sent to Paymentech, asking for the results of the settlement.
3. Paymentech will either return a *No data to send back at this time* message or an actual RFR response.
4. If an actual RFR response is returned, it consists of *S* Record output records, each one indicating if the transactions in the batch were successfully deposited (or refunded, in the case of refund transactions).
5. If an actual RFR response is never able to be retrieved, then there is a problem with the batch and the merchant must call Paymentech to resolve the problems (i.e., reconciliation is done manually and offline).
6. The cassette allows the merchant to manually issue an RFR request if an RFR response file was never successfully retrieved from Paymentech and all automatic retries have been exhausted. This is accomplished by issuing a BatchClose command, specifying the $REISSUERFR parameter. See the following table for details.

Table 18. Required and Optional keywords for BatchClose command

Keyword	Type	Value
PAYMENTTYPE	Required	Specified name must be **Paymentech**.
FORCE	Optional	Valid values are **0** and **1**. A value of **1** indicates a local close (i.e., will not cause a message to be sent to Paymentech) should be performed. A value of **1** will only be allowed when the batch is in the OPEN state with a status of OUT_OF_BALANCE. This can occur in the following situations: • The RFR Attempts have been exhausted without receiving a good response from Paymentech. • An RFR Response was received from Paymentech; however, there are problems with the batch.
$REISSUERFR	Optional	This Boolean indicator is used to indicate if an RFR request should be manually issued to Paymentech. Valid values are: **0**–Do not issue RFR request **1**–Issue RFR request A value of **1** will only be allowed when an RFR response file was never successfully retrieved from Paymentech and all automatic retries have been exhausted. The RFR request will only be tried once; that is, no automatic retries will occur. The merchant must query the batch's RFR status to determine if the RFR request was successful.

The following table describes the circumstances in which the above keywords are valid. Both the FORCE and $REISSUERFR keywords are valid only after the cassette has completed its normal RFR processing.

Table 19. RFR results and allowed keywords

RFR results	Batch state	Batch status	RFR status	$REISSUERFR allowed?	Force allowed?
RFR response received; batch is balanced.	CLOSED	BALANCED	**5**–Complete	No	No
Paymentech returns **No data to send back at this time** in response to an RFR Request. The cassette has not yet exhausted its automatic retries.	CLOSING	NOT_YET_ BALANCED	**1**–File not available for pickup	No	No
Paymentech returns **No data to send back at this time** in response to an RFR Request. The cassette has exhausted its automatic retries.	OPEN	BATCH_OUT _OF_BALANCE	**2**–File not available for pickup; attempts exhausted	Yes	Yes
RFR Response received, but not all transactions were successfully deposited and/or refunded.	OPEN	BATCH_OUT _OF_BALANCE	**5**–Complete	No	Yes
A communi-cation error occurred while attempting to send and/or receive RFR. The cassette has not yet exhausted its automatic retries.	CLOSING	NOT_YET_ BALANCED	**6**–Communi-cations Error	No	No

Table 19. RFR results and allowed keywords (continued)

RFR results	Batch state	Batch status	RFR status	$REISSUERFR allowed?	Force allowed?
A communication error occurred while attempting to send and/or receive RFR. The cassette has exhausted its automatic retries.	OPEN	NOT_YET_BALANCED	7–Communications Error; attempts exhausted	Yes	No

For more information on how the batch settlement process works, see "A Paymentech batch settlement process example" on page 9.

BatchClose command when the batch is marked as Offline

If the batch is considered Offline, then the Cassette for Paymentech will do the following:

1. Mark the batch as CLOSED.
2. Close all Payments/Credits in the batch.
3. Update the Batch Status to BATCH_BALANCED.

Merchants who wish to settle offline transactions might operate in the following way:

1. AcceptPayment with AutoApprove and AutoDeposit, specifying $SETTLEMENTMODE=1 (offline).
2. When the merchant is ready to obtain a list of authorized transactions for offline settlement:
   a. Issue a QueryAccounts, specifying the MERCHANTNUMBER.
   b. Based on results returned in QueryAccounts, find all batches associated with the account, by issuing a QueryBatches command, specifying the appropriate MERCHANTNUMBER and ACCOUNTNUMBER, with a filter of STATE= "batch_open". This will return the batchNumber of the currently open batch(es) for that Merchant/Account.
   c. Based on results, determine which of these batches are destined for Offline settlement (based on the settlementMode attribute returned in the QueryBatches command). For those Offline batches:
      1) Issue a BatchClose (causes batch and associated payments/credits to be CLOSED).
      2) Issue a QueryBatches command, specifying the appropriate MERCHANTNUMBER, ACCOUNTNUMBER, and BATCHNUMBER, with the WITHCREDITS and WITHPAYMENTS flags set to True. This will return all transactions (and the transaction details such as Authorization Codes) associated with the batch. The merchant can then settle these transactions outside of WebSphere Commerce Payments.
      3) The Batch can be deleted via the DeleteBatch command when the merchant no longer needs to keep the transaction data in WebSphere Commerce Payments.

In addition, the BatchClose command is fully supported as documented in the *WebSphere Commerce Payments Programming Guide and Reference*.

CassetteControl

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

- PRC_COMMAND_NOT_SUPPORTED
- RC_NONE

CloseOrder

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

CancelOrder

Cassettes are responsible for deleting ancillary objects. For the Cassette for Paymentech, all related records and tables associated with the canceled Order are deleted when the CancelOrder command is issued.

CreateAccount

Table 20. Required and optional keywords for CreateAccount command

Required keywords	Type	Value
CASSETTENAME	Required	The name specified here must be **Paymentech**.
$BATCHCLOSETIME	Optional	Time of day to close the account's online batches (HHMM in 24–hour format). If not specified, no automatic close will occur. (Min/Max: 0000-2359)

DeleteAccount

Table 21. Required keywords for the DeleteAccount command

Keywords	Type	Value
CASSETTENAME	Required	The name specified must be **Paymentech**.

CreateMerchantCassetteObject

The CreateMerchantCassetteObject command creates a Paymentech Division for the specified Merchant. One Division object exists for each division supported by the Merchant.

Table 22. Required keywords for the CreateMerchantCassetteObject command

Keywords	Type	Value
CASSETTENAME	Required	The specified name must be **Paymentech**.
OBJECTNAME	Required	The identifier for the MerchantCassette object. Must be **DIVISION**.
$DIVISIONNUMBER	Required	Paymentech Division Number. Must be 6 digits.

Table 22. Required keywords for the CreateMerchantCassetteObject command (continued)

Keywords	Type	Value
$CURRENCY	Required	This is the currency code associated with the Division Number. (Min/Max: 0-999)

DeleteMerchantCassetteObject

Table 23. Required keywords for the DeleteMerchantCassetteObject command

Keywords	Type	Value
CASSETTENAME	Required	The name specified must be **Paymentech**.
OBJECTNAME	Required	The identifier for the MerchantCassette object. Must be **DIVISION**.
$DIVISIONNUMBER	Required	Paymentech Division Number. Must be 6 digits.

CreatePaySystem

Table 24. Required keywords for CreatePaySystem command

Keywords	Type	Value
CASSETTENAME	Required	Specified name must be **Paymentech**.
$PRESENTERID	Required	The Paymentech Presenter's ID. A 6-digit number assigned by Paymentech. (Min/Max: 100000-999999)
$PRESENTERPASSWORD	Required	The Paymentech Presenter's password. (Min/Max: 1-8 characters)
$SUBMITTERID	Required	The Paymentech submitter's ID. A 6-digit number assigned by Paymentech. (Min/Max: 100000-999999)
$SUBMITTERPASSWORD	Required	The Paymentech submitter's password. (Min/Max: 1-8 characters)

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 there is no open batch, then one will be created). If the transaction being deposited has a SettlementMode of Online, then it is deposited in a batch that is designated as Online. If the transaction that is being deposited has a SettlementMode of Offline, then it is deposited in a batch that is designated as Offline. Only Online batches will be settled through Paymentech. Executing this

command does not send a message to the financial institution. It is a local operation only. If the operation is successful, then the payment moves from the APPROVED state to the DEPOSITED state.

Table 25. Required keywords for Deposit command

Keyword	Type	Value
PAYMENTTYPE	Required	Specified name must be **Paymentech**.
AMOUNT	Required	Specifies the deposited amount. Specify without the decimal point. For example, input 1234 for $12.34. For Paymentech supported Visa and MasterCard transactions, the amount must be > $.01 and <= $25,000 (or the international currency equivalent). For all other brands, the amount must be > $1.00 and <= $25,000 (or the international currency equivalent). (Min/Max: 0 - 2147483647)
BATCHNUMBER	Not Allowed	Not allowed since all batches are opened implicitly (must be specified).

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.

Table 26. Required keyword for DepositReversal command

Keyword	Type	Value
PAYMENTTYPE	Required	Specified name must be **Paymentech**.

ModifyAccount

Table 27. Required and optional keywords for ModifyAccount command

Optional keywords	Type	Value
CASSETTENAME	Required	The name specified here must be **Paymentech**.
$BATCHCLOSETIME	Optional	Time of day to close the account's online batches (HHMM in 24–hour format). If not specified, no automatic close will occur. (Min/Max: 0000-2359)

ModifyCassette

Table 28. Required and optional keywords for ModifyCassette command

Optional keywords	Type	Value
CASSETTENAME	Required	The name specified here must be **Paymentech.**
$ONLINEIPADDR	Required	IP address for online authorizations. (Min/Max: 0-254 characters)
$ONLINEPORTNUMBER	Optional	Port number for online authorizations. (Min/Max: 0-65535)
$BATCHIPADDR	Optional	IP address for batch settlement. (Min/Max: 0-254 characters)
$BATCHPORTNUMBER	Optional	Port number for batch settlement. (Min/Max: 0-65535)
$READTIMEOUT	Optional	Number of seconds to wait while communicating with the Paymentech payment gateway. (Min/Max: 0-65535)
$MAXRETRIES	Optional	When a communications error occurs (not a connection failure), the maximum number of immediate retires to attempt before either returning a communication error, or before entering the delayed retry cycle. (Min/Max: 0-65535)
$ATTEMPTINTERVAL	Optional	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). (Min/Max: 0-2147483647)
$MAXATTEMPTS	Optional	Maximum number of retry sets. An integer between 0 and 2147483647. If not specified, the default is **3**. (Min/Max: 0-65535)
$RFRDELAYTIME	Optional	The number of minutes to wait before issuing RFR to Paymentech. The default and recommended value is 10 minutes. (Min/Max: 0-2147483647)
$RFRATTEMPTS	Optional	When the RFR is sent to Paymentech, and a *no data to send back* response is received, then this is the maximum number of times the cassette will reattempt to retrieve the Batch Response file from Paymentech. The default value is 3. (Min/Max: 1-65535)

ModifyPaySystem

Table 29. Required and optional keywords for ModifyPaySystem command

Optional keywords	Type	Value
CASSETTENAME	Required	The specified name must be **Paymentech**.
$PRESENTERID	Optional	The specified Presenter ID for Paymentech. (Min/Max: 100000-999999)
$PRESENTERPASSWORD	Optional	The specified Presenter password for Paymentech. (Min/Max: 1-8 characters)
$SUBMITTERID	Optional	The specified Submitter ID for Paymentech. (Min/Max: 100000-999999)
$SUBMITTERPASSWORD	Optional	The specified Submitter password for Paymentech. (Min/Max: 1-8 characters)

DeletePaySystem

Table 30. Required and optional keywords for the DeletePaySystem command

Keywords	Type	Value
CASSETTENAME	Required	The name specified must be **Paymentech**.

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 Paymentech 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 Paymentech 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. It is a local operation only.

Table 31. Cassette-specific processing of parameters for the Refund command

Required keyword	Type	Value
PAYMENTTYPE	Required	Specified name must be **Paymentech**.

Table 31. Cassette-specific processing of parameters for the Refund command (continued)

Required keyword	Type	Value
AMOUNT	Required	Specifies refund amount. Specify without the decimal point. For example, input 1234 for $12.34. For Paymentech supported Visa and MasterCard transactions, the amount must be > $.01 and <= $25,000 (or the international currency equivalent). For all other brands, the amount must be > $1.00 and <= $25,000 (or the international currency equivalent). (Min/Max: 0 - 2147483647)

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.

Table 32. Required keyword for RefundReversal command

Keyword	Type	Value
PAYMENTTYPE	Required	Specified name must be **Paymentech**.

Chapter 7. Object reference

The object model of the Cassette for Paymentech closely reflects the generic model of 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 Paymentech object extensions appear in these documents as extensions to the generic objects of the framework.

Financial objects used by Cassette for Paymentech

Each of the framework's generic financial objects is extended by the Cassette for Paymentech.

Paymentech Order

Table 33. Cassette properties that belong to a PSOrder Object

Field Name	Description
pan	The Credit or Debit card number as specified on the AcceptPayment command. This value is always present. 10-19 digit string.
expiry	The Credit or Debit card's expiration date in the form YYYYMM as specified on the AcceptPayment command.
cardVerifyCode	Card Verification Value that is used to assist in authenticating the physical presence of a card. This is a 3-4 digit string.
cardSecurityPresence	Indicator that is used to validate the presence of a card security value. Valid values are: **1**–Value is present **2**–Value on card, but illegible **9**–Cardholder states the card has no card security value **Note:** These values currently only apply to Visa and Discover.
cardHolderName	Name of the cardholder. This is a 1-30 character string.
avsStreetAddress	Specifies the street address of the location of the cardholder. This is a 0-30 character string.
avsStreetAddress2	Additional street address information. This is a 0-28 character string.
avsCity	Indicates the city name of the location of the cardholder. This is a 0-20 character string.
avsStateProvince	Indicates the name or abbreviation of the state of the location of the cardholder. This is a 2–character string (must be a valid US state abbreviation).
avsPostalCode	The postal code of the location of the cardholder. This is a 5–9 character string.

Table 33. Cassette properties that belong to a PSOrder Object (continued)

Field Name	Description
avsCountryCode	Country code of the location (country or region) of the cardholder. This is a 2–character string.
avsPhoneNumber	Phone number of the cardholder in the format AAAEEENNNNXXXX or AAAEEENNNN, where AAA = Area Code EEE = Exchange NNNN = Number XXXX = Extension
avsPhoneType	Type of phone number specified in avsPhoneNumber. Valid values are: D - Day H - Home N - Night W - Work
transactionType	Describes the circumstances under which the transaction takes place. Valid values are: **2**–Recurring Transaction: Designates a transaction that represents an arrangement between a cardholder and the merchant where transactions are going to occur on a periodic basis. **3**–Installment Payment: Designates a group of transactions that originated from a single purchase where the merchant agrees to bill the cardholder in installments. **7**–Non-SET Transaction Channel Encrypted: Designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction does not include the management of any SET certificates, but does include the use of transaction encryption such as SSL. **8**–Non-Secure Electronic Commerce Transaction: Designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction does not include the management of any SET certificates and does not include the use of any transaction encryption such as SSL.
divisionNumber	The Paymentech-assigned division number. This is a 6–digit string.
settlementMode	Indicates the settlement mode of the Order. Valid values are: **0**–Online settlement: Designates a transaction that is settled through the Paymentech processor when the BatchClose command is issued. **1**–Offline settlement: Designates a transaction in which settlement occurs outside the scope of Paymentech. The merchant is responsible for settling these transactions.

Paymentech Payment

Table 34. Cassette properties that belong to a PSPayment Object

Field Name	Description
authResponseReason	Paymentech response reason code which indicates the status of the authorization request. An authResponseReason of *100* indicates approval. See the Paymentech Online 6.0.1 specification for a complete list of all Response Reason Codes.
authResponseDate	The date the authorization response was obtained, which is specified in the YYMMDD format.
authCode	Authorization Code that the issuer uses to show an authorization request was approved. This is a 1–6 character string.
avsResultCode	Response to address verification request. See the Paymentech Online 6.0.1 specification for a complete list of all AVS Response Codes.
commonAVSCode	The 2-character string Common Address Verification Service result code. For more information, see "Address Verification Service (AVS) result codes" on page 56.
cvv2ResultCode	CVV2 Result code returned by the card issuer in response to a CVV2/CVC2 request. Valid values are: M—Value matched N—Value not matched P—Not processed S—Should be on the card (Discover, Visa only) U—Unsupported by the issuer I—Invalid (Master, Visa Only) ""—Blank if not Discover, MasterCard, Visa
merchantOrderNumber	A 1–8 digit number generated by the Cassette for Paymentech that uniquely identifies the transaction.
paymentDeposited	**Y** or **N** indicator that indicates if the payment was successfully deposited. Note that for Offline settlement transactions, this will always be **N** since the transaction is never settled within WebSphere Commerce Payments.

Paymentech Credit

Table 35. Cassette properties that belong to a PSCredit Object

Field Name	Description
responseReasonCode	Paymentech response reason code that indicates the status of the refund request. A responseReasonCode of *100* indicates approval. See the Paymentech Online 6.0.1 specification for a complete list of all Response Reason Codes.
responseDate	The date the refund response was obtained, which is specified in YYMMDD format.
merchantOrderNumber	A 1–8 digit number generated by the Cassette for Paymentech that uniquely identifies the transaction.

Paymentech Batch

Table 36. Cassette properties that belong to a PSBatch Object

Field Name	Description
creationDate	Date in YYMMDD format that indicates when the batch was sent to Paymentech for settlement.
settlementMode	Indicates the settlement mode of the batch. Valid values are: **0**–Online settlement: Designates a batch that is settled through the Paymentech processor when the BatchClose command is issued. **1**–Offline settlement: Designates a batch in which settlement occurs outside the scope of Paymentech.
rfrStatus	Indicates the status of the RFR Request. Valid values are: 0—RFR has not been sent 1—File not available for pickup 2—File not available for pickup; attempts exhausted 3—Format error; cannot create RFR Request 4—Format error; cannot process RFR Response 5—Complete 6—Communications error 7—Communications error; attempts exhausted 8—Not Applicable (batch settlement is offline)

Administration objects used by Cassette for Paymentech

The Cassette for Paymentech uses and extends these framework objects for WebSphere Commerce Payments administration:

- CassetteAdmin
- AccountAdmin
- PaySystemAdmin
- MerchantCassetteObject (Division)

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

CassetteAdmin

Table 37. Cassette properties that belong to a PSCassette Object

Field	Description
onlineIPAddress	IP Address issued by Paymentech for connectivity into the Paymentech Online Authorization system.
onlinePortNumber	Port number issued by Paymentech for connectivity into the Paymentech Online Authorization system.
batchIPAddress	IP Address issued by Paymentech for connectivity into the Paymentech Batch Settlement system.
batchPortNumber	Port number issued by Paymentech for connectivity into the Paymentech Batch Settlement system.
readTimeout	Number of seconds to wait while communicating with Paymentech.

Table 37. Cassette properties that belong to a PSCassette Object (continued)

Field	Description
maxRetries	When a communications error occurs (that is, not a connection failure), the maximum number of immediate retries to attempt before either returning a communication error, or before entering the delayed retry cycle.
maxAttempts	Maximum number of delayed retry sets.
attemptInterval	When a communications error occurs, the number of seconds to wait before trying the next set of (delayed) retries.
rfrDelayTime	The number of minutes to wait between RFR attempts.
rfrAttempts	The maximum number of RFR attempts.

AccountAdmin

Table 38. Cassette properties that belong to a PSMerchantAccount

Field Name	Description
batchCloseTime	Time in 24-hour HHMM format that indicates when Online batches associated with this account should be scheduled for automatic closure.

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 Paymentech PaySystem data:

Table 39. Cassette properties that belong to PSMerchantCassetteSettings

Field Name	Description
presenterID	The 6-digit Presenter ID assigned by Paymentech that is used during the settlement process.
presenterPassword	The 1–8 character Presenter password assigned by Paymentech that is used during the settlement process.
submitterID	The 6-digit Submitter ID assigned by Paymentech that is used during the settlement process.
submitterPassword	The 1–8 character Submitter password assigned by Paymentech that is used during the settlement process.

MerchantCassetteObject (Division)

Table 40. Cassette properties that belong to a PSMerchantCassetteObject

Field Name	Description
divisionNumber	The 6-digit division number assigned by Paymentech.

Table 40. Cassette properties that belong to a PSMerchantCassetteObject (continued)

Field Name	Description
currencyCode	ISO Currency code associated with this division.

Address Verification Service (AVS) result codes

When Address Verification Services (AVS) are requested on an AcceptPayment 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 "Address Verification Service" on page 3.

The following table illustrates the way the Cassette for Paymentech maps the Paymentech-specific AVS result codes to the WebSphere Commerce Payments framework's common AVS codes. Note that the Paymentech 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 41. WebSphere Commerce Payments common AVS result codes mapped to Cassette for Paymentech AVS result codes

Paymentech AVS result code (returned from issuer)	WebSphere Commerce Payments common AVS result code	Explanations
I1	0 (Complete match)	Both the postal code (that is, the AVS 5-digit and 9-digit) and the street address match.
I3	0 (Complete match)	Both the 5-digit postal code and the street address match.
IA*	0 (Complete match)	Both the postal code (that is, the AVS 5-digit and 9-digit) and the street address match.
I5	1 (Street match)	Only the 4-digit portion of the 9-digit postal code and the street address match.
I7	1 (Street match)	The street address matches, but the postal code (that is, the AVS 5-digit and 9-digit) does not match.
IB*	1 (Street match)	International street address matches. Postal code not verified due to incompatible formats (Both were sent.)
I2	2 (Postal code match)	The 5-digit and the 9-digit postal code match, but the street address does not match.
I4	2 (Postal code match)	The 5-digit postal code matches, but the 9-digit postal code and the street address do not match.

Table 41. WebSphere Commerce Payments common AVS result codes mapped to Cassette for Paymentech AVS result codes (continued)

Paymentech AVS result code (returned from issuer)	WebSphere Commerce Payments common AVS result code	Explanations
IP*	2 (Postal code match)	International postal code matches. Street address not verified due to incompatible formats (Both were sent.)
I6	3 (No match)	Only the 4-digit portion of the 9-digit postal code matches (i.e., the 5-digit postal code and the street address do not match).
I8	3 (No match)	The following do not match: • The 5-digit postal code • The 9-digit postal code • The street address
N1	4 (Other)	No address given with order
N2	4 (Other)	Bill-to address did not pass Paymentech's edit checks
" "	4 (Other)	AVS not performed (Blanks returned)
IG	4 (Other)	Issuer does not participate in Global AVS (International issuer). International address information not provided, or provided and issuer does not participate.
IU	4 (Other)	Issuer does not participate in Global AVS (Domestic issuer)
ID	4 (Other)	Issuer does not participate in AVS
IE	4 (Other)	Edit error—AVS data is invalid
IS	4 (Other)	System unavailable or time-out
IC*	4 (Other)	International street address and postal code not verified due to incompatible format (Both were sent.)

Note: International AVS codes denoted with an asterisk are for VISA only.

Appendix A. Cassette for Paymentech return codes

Almost all of the error conditions raised by the Cassette for Paymentech 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 Paymentech are defined by the framework (see the *WebSphere Commerce Payments Programming Guide and Reference* for a list). The following table lists secondary return codes for Paymentech errors and their definitions.

Secondary Return Code	Value	Description
SRC_CASSETTE_AVS_STREETADDRESS2	10001	Refers to $AVS.STREETADDRESS2 protocol data.
SRC_CASSETTE_AVS_PHONENUMBER	10002	Refers to $AVS.PHONENUMBER protocol data.
SRC_CASSETTE_TRANSACTIONTYPE	10003	Refers to $TRANSACTIONTYPE protocol data.
SRC_CASSETTE_ATTEMPTINTERVAL	10005	Refers to $ATTEMPTINTERVAL protocol data.
SRC_CASSETTE_MAXATTEMPTS	10006	Refers to $MAXATTEMPTS protocol data.
SRC_CASSETTE_PRESENTERID	10011	Refers to $PRESENTERID protocol data.
SRC_CASSETTE_PRESENTERPASSWORD	10012	Refers to $PRESENTERPASSWORD protocol data.
SRC_CASSETTE_SUBMITTERID	10013	Refers to $SUBMITTERID protocol data.
SRC_CASSETTE_SUBMITTERPASSWORD	10014	Refers to $SUBMITTERPASSWORD protocol data.
SRC_CASSETTE_DIVISIONNUMBER	10015	Refers to $DIVISIONNUMBER protocol data.
SRC_CASSETTE_DEPOSIT_AMOUNT	10016	Refers to the deposit amount.
SRC_CASSETTE_READTIMEOUT	10021	Refers to $READTIMEOUT protocol data.
SRC_CASSETTE_MAXRETRIES	10022	Refers to $MAXRETRIES protocol data.
SRC_CASSETTE_RFRDELAYTIME	10023	Refers to $RFRDELAYTIME protocol data.
SRC_CASSETTE_RFRATTEMPTS	10024	Refers to $RFRATTEMPTS protocol data.
SRC_CASSETTE_AVS_PHONETYPE	10025	Refers to $AVS.PHONETYPE protocol data.
SRC_CASSETTE_ONLINEIPADDR	10026	Refers to $ONLINEIPADDR protocol data.
SRC_CASSETTE_ONLINEPORTNUMBER	10027	Refers to $ONLINEPORTNUMBER protocol data.

Secondary Return Code	Value	Description
SRC_CASSETTE_BATCHIPADDR	10028	Refers to $BATCHIPADDR protocol data.
SRC_CASSETTE_BATCHPORTNUMBER	10029	Refers to $BATCHPORTNUMBER protocol data.
SRC_CASSETTE_DIVISION	10030	Refers to the merchant division.
SRC_CASSETTE_SETTLEMENTMODE	10031	Refers to $SETTLLEMENTMODE protocol data.
SRC_CASSETTE_REISSUERFR	10032	Refers to $REISSUERFR protocol data.
SRC_CASSETTE_CARDSECURITYPRESENCE	10033	Refers to $CARDSECURITYPRESENCE protocol data.
SRC_METHOD_NOT_IMPLEMENTED	20000	The Paymentech Cassette does not support the framework method.
SRC_CASSETTE_BUNDLE_ID_MISMATCH	20001	Resource bundle loaded by framework is different from the constant used to identify the resource bundle in the cassette.
SRC_ACCOUNT_SELECT_SQL_FAILURE	21000	Database error occurred while attempting to retrieve a Paymentech account.
SRC_ACCOUNT_CREATE_SQL_FAILURE	21001	Database error occurred while attempting to create a Paymentech account.
SRC_ACCOUNT_UPDATE_SQL_FAILURE	21002	Database error occurred while attempting to update a Paymentech account.
SRC_ACCOUNT_DELETE_SQL_FAILURE	21003	Database error occurred while attempting to delete a Paymentech 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 Paymentech batch.
SRC_BATCH_CREATE_SQL_FAILURE	22003	Database error occurred while attempting to create a Paymentech batch.
SRC_BATCH_UPDATE_SQL_FAILURE	22004	Database error occurred while attempting to update a Paymentech batch.
SRC_BATCH_DELETE_SQL_FAILURE	22005	Database error occurred while attempting to delete a Paymentech batch.
SRC_BATCH_NULL_ORDER_FOR_PAYMENT	22006	The batch contains a null order for a payment.
SRC_BATCH_NULL_PAYMENT	22007	The batch contains a null payment.
SRC_BATCH_BAD_BATCH_IN_PAYMENT	22008	The batch contains a payment which has a different batch number.

Secondary Return Code	Value	Description
SRC_BATCH_NULL_ORDER_FOR_CREDIT	22009	The batch contains a null order for a credit.
SRC_BATCH_NULL_CREDIT	22010	The batch contains a null credit.
SRC_BATCH_BAD_BATCH_IN_CREDIT	22011	The batch contains a credit which a has a different batch number.
SRC_BATCH_PURGE_INCOMPLETE	22012	Unable to complete BatchPurge operation.
SRC_BATCH_DUPLICATE	22013	The batch already exists.
SRC_BATCH_MUST_BE_OUT_OF_BALANCE	22014	A batch cannot be purged unless the batch is out of balance.
SRC_BATCH_RFR_NO_FILE_AVAILABLE	22015	No file (″S″ record) was available for the RFR to retrieve.
SRC_BATCH_RFR_REQUEST_FORMAT_ERROR	22016	A format error occurred while building the RFR.
SRC_BATCH_RFR_RESPONSE_FORMAT_ERROR	22017	A format error occurred while processing the RFR response.
SRC_BATCH_TIMER_WORK_ITEM_FAILURE	22018	A failure occurred while attempting to automatically close the batch.
SRC_BATCH_CLOSE_WORK_ITEM_FAILURE	22019	A failure occurred while attempting to schedule an automatic batch close.
SRC_BATCH_NOT_READY_FOR_SETTLEMENT	22020	The batch is not ready for settlement. All authorizations in the batch must be at least 30 minutes old before settlement can take place.
SRC_BATCH_CANNOT_BE_FORCE_CLOSED	22021	The batch cannot be forced closed. The batch is not in the valid state to allow the Force option on BatchClose.
SRC_BATCH_ANOTHER_BATCH_IN_CLOSING_STATE	22022	The batch cannot be closed since there is another batch in the process of being closed for the account.
SRC_BATCH_CANNOT_SPECIFY_BOTH_FORCE _AND_REISSUERFR	22023	Both the FORCE flag and the ReissueRFR flag were specified on the BatchClose API.
SRC_BATCH_CANNOT_REISSUE_RFR	22024	The batch is not in a state in which the RFR can be re-issued.
SRC_BATCH_SETTLEMENT_FILE_ALREADY_SENT	22025	A BatchClose was requested; however, a settlement file has already been sent for the batch. Specify either the Force Flag or the ReissueRFR flag and try again.
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 Paymentech order.
SRC_ORDER_CREATE_SQL_FAILURE	23002	Database error occurred while attempting to create a Paymentech order.

Secondary Return Code	Value	Description
SRC_ORDER_UPDATE_SQL_FAILURE	23003	Database error occurred while attempting to update a Paymentech order.
SRC_ORDER_DELETE_SQL_FAILURE	23004	Database error occurred while attempting to delete a Paymentech 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 error; unable to obtain a crypto key.
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 Paymentech payment.
SRC_PAYMENT_CREATE_SQL_FAILURE	24002	Database error occurred while attempting to create a Paymentech payment.
SRC_PAYMENT_UPDATE_SQL_FAILURE	24003	Database error occurred while attempting to update a Paymentech payment.
SRC_PAYMENT_DELETE_SQL_FAILURE	24004	Database error occurred while attempting to delete a Paymentech payment.
SRC_PAYMENT_SELECT_CLOSE_FAILURE	24005	Database error occurred while attempting to end a query.
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 Paymentech credit.
SRC_CREDIT_CREATE_SQL_FAILURE	25002	Database error occurred while attempting to create a Paymentech credit.
SRC_CREDIT_UPDATE_SQL_FAILURE	25003	Database error occurred while attempting to update a Paymentech credit.
SRC_CREDIT_DELETE_SQL_FAILURE	25004	Database error occurred while attempting to delete a Paymentech credit.
SRC_QUERY_ORD_SELECT_SQL_FAILURE	26000	A database error occurred while processing the cassette orders.
SRC_QUERY_PAY_SELECT_SQL_FAILURE	26001	A database error occurred while processing the cassette payments.
SRC_QUERY_CRE_SELECT_SQL_FAILURE	26002	A database error occurred while processing the cassette credits.
SRC_QUERY_BAT_SELECT_SQL_FAILURE	26003	A database error occurred while processing the cassette batches.
SRC_QUERY_ACC_SELECT_SQL_FAILURE	26004	A database error occurred while processing the cassette accounts.

Secondary Return Code	Value	Description
SRC_QUERY_CASSETTE_SELECT_SQL_FAILURE	26005	A database error occurred while processing the cassette configuration.
SRC_QUERY_PAYSYS_SELECT_SQL_FAILURE	26006	A database error occurred while processing the cassette pay systems.
SRC_CASSETTE_ADMIN_SELECT_SQL_FAILURE	27000	Database error occurred while attempting to retrieve a Paymentech cassette admin.
SRC_CASSETTE_ADMIN_UPDATE_SQL_FAILURE	27001	Database error occurred while attempting to update a Paymentech cassette admin.
SRC_NO_RESPONSE_TO_REQUEST_RETRIES_EXHAUSTED	30000	A communication error occurred; no response has been received.
SRC_CONNECT_FAILED_TO_PAYMENTECH_HOST	30001	Unable to open a socket with the configured Paymentech host.
SRC_WRITE_FAILED_TO_PAYMENTECH_HOST	30002	A communication error occurred; unable to send request to Paymentech host.
SRC_SOCKET_READ_FAILED	30003	A communication error occurred; socket read failure.
SRC_TOO_MANY_OUTSTANDING_AUTHS	30004	A communication error occurred; too many outstanding authorization requests.
SRC_OPERATION_INTERRUPTED_BY_CASSETTE_ SHUTDOWN	30005	A communication error occurred; operation interrupted by cassette shutdown.
SRC_PAYMENTECHLLP_NETWORKING_IS_STOPPED	30006	A communication error occurred; Paymentech LLP Networking is stopped.
SRC_PAYMENTECHLLP_PROTOCOL_UNEXPECTED_ BYTE_VALUE_RECEIVED	30007	A communication error occurred; unexpected byte value received.
SRC_PAYMENTECHLLP_PROTOCOL_READ_TIMEOUT_ EXPIRED	30008	A communication error occurred; read timeout expired.
SRC_PAYMENTECHLLP_READ_PACKET_TOO_LARGE	30009	A communication error occurred; read packet too large.
SRC_PAYMENTECH_PACKET_FORMAT_ERROR	30010	Format error in the response received from the Paymentech host.

Appendix B. Cassette for Paymentech messages

This appendix contains the Cassette for Paymentech-specific messages.

CEPPaymentech1001: The Paymentech 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.

CEPPaymentech1002: The Paymentech Cassette has started.

Severity: Information

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

User Response: None

CEPPaymentech1003: The Paymentech Cassette has shut down.

Severity: Information

Explanation: The Paymentech Cassette is no longer active.

User Response: None

CEPPaymentech2000: 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.

CEPPaymentech2001: Unable to create the account. An account for merchant number *merchNum* **already exists.**

Severity: Error

Explanation: The specified merchant already has an account defined. Each merchant can have only one account associated with it.

User Response: Try to create the account again but specify a different merchant number, or delete the existing merchant's account and re-create it.

CEPPaymentech2002: 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.

CEPPaymentech2003: 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.

CEPPaymentech2004: 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.

CEPPaymentech2005: 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.

CEPPaymentech2006: 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.

CEPPaymentech2007: 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.

CEPPaymentech2008: 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.

CEPPaymentech2009: 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.

CEPPaymentech2010: An error occurred while attempting to settle a batch for Merchant *merchNum*, **Account** *acctNum*, **Batch** *batchNum*. **The Paymentech 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 Paymentech host representative to determine why the batch already exists.

CEPPaymentech2011: Unable to create the division. A division already exists for merchant number *merchNum* **with the Division Number** *divisionNum*.

Severity: Error

Explanation: The specified merchant already has a division for the specified division number.

User Response: Try the request again, but specify a division number that does not already exist.

CEPPaymentech2012: Unable to create the division. A division already exists for merchant number *merchNum* **with currency code** *currencyCode*.

Severity: Error

Explanation: The specified merchant already has a division defined for the specified currency code.

User Response: Re-attempt the request, specifying a currency code that does not already exist.

CEPPaymentech3000: 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.

CEPPaymentech3001: 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.

CEPPaymentech3002: 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.

CEPPaymentech3003: 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.

CEPPaymentech3004: 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.

CEPPaymentech3005: 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.

CEPPaymentech3006: 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.

CEPPaymentech3007: 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.

CEPPaymentech3008: 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.

CEPPaymentech3009: 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.

CEPPaymentech3010: 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.

CEPPaymentech3011: 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.

CEPPaymentech3012: 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.

CEPPaymentech3013: 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.

CEPPaymentech3014: 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.

CEPPaymentech3015: The settlement file for Merchant Number = *merchNum*. **Batch Number =** *batchNum*. **was sent successfully; however, the cassette was unable to create an RFR request to send to Paymentech.**

Severity: Error

Explanation: This is an internal error in the cassette.

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

CEPPaymentech3016: A format error was encountered with either the RFR Request message or the RFR response message. As a result, the batch for Merchant Number = *merchNum*. **and Batch Number =** *batchNum*. **cannot be reconciled.**

Severity: Error

Explanation: This is an internal error in the cassette.

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

CEPPaymentech3017: An error occurred while attempting to automatically close the batch for Merchant Number = *merchNum* **Batch Number =** *batchNum* **Account Number =** *acctNum* **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.

CEPPaymentech3018: An exception was caught while attempting to contruct a CassetteWorkItem to be sent to the timer queue to schedule an automatic BatchClose request. Merchant Number = *merchNum* **Batch Number =** *batchNum* **Account Number =** *acctNum*

Severity: Error

Explanation: This is an internal error in the cassette.

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

CEPPaymentech3019: An exception was caught while attempting to contruct a CassetteWorkItem to be sent to the service queue to schedule an automatic BatchClose request. Merchant Number = *merchNum* **Batch Number =** *batchNum* **Account Number =** *acctNum***.**

Severity: Error

Explanation: This is an internal error in the cassette.

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

CEPPaymentech3020: An exception was caught while attempting to process the RFR response from Paymentech. Merchant Number = *merchNum* **Batch Number =** *batchNum***.**

Severity: Error

Explanation: This is an internal error in the cassette.

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

CEPPaymentech3021: An RFR response was received in which there is no corresponding batch. The first Merchant Order Number in the RFR Response is: *merchOrderNum***.**

Severity: Error

Explanation: This is an internal error in the cassette.

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

CEPPaymentech3022: An unexpected error occurred while processing an RFR response from Paymentech.

Severity: Error

Explanation: This is an internal error in the cassette.

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

CEPPaymentech4000: 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.

CEPPaymentech4001: 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.

CEPPaymentech4002: 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.

CEPPaymentech4003: 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.

CEPPaymentech4004: 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.

CEPPaymentech4005: 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.

CEPPaymentech4006: 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.

CEPPaymentech4007: 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.

CEPPaymentech5000: 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 payment exists for the merchant and the order.

CEPPaymentech5001: 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.

CEPPaymentech5002: 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.

CEPPaymentech5003: 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.

CEPPaymentech5004: 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.

CEPPaymentech5005: 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.

CEPPaymentech5006: 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.

CEPPaymentech5007: An error occurred while attempting to settle a payment. Response Reason Code = *reasonCode***. Merchant Number =** *merchNum***. Batch Number =** *batchNum***. Order Number =** *orderNum***. Payment Number =** *payNum***. MerchantOrderNumber =** *merOrderNum***.**

Severity: Error

Explanation: The RFR response returned from Paymentech indicated that an error occurred while depositing the indicated payment.

User Response: Contact your Paymentech representative to determine why the deposit was not successful.

CEPPaymentech6000: 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.

CEPPaymentech6001: 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.

CEPPaymentech6002: 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.

CEPPaymentech6003: 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.

CEPPaymentech6004: 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.

CEPPaymentech6005: 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 specified credit exists for the specified merchant and the specified order.

CEPPaymentech6006: An error occurred while attempting to issue a refund. Response Reason Code = *reasonCode*. **Merchant Number =** *merchNum*. **Batch Number =** *batchNum*. **Order Number =** *orderNum*. **Credit Number =** *creditNum*. **MerchantOrderNumber =** *merOrderNum*.

Severity: Error

Explanation: The Request For Response (RFR) response returned from Paymentech indicated that an error occurred while refunding the specified credit.

User Response: Contact your Paymentech representative to determine why the refund was not successful.

CEPPaymentech7000: An SQL exception was caught while processing the result set for a query on cassette orders.

Severity: Error

Explanation: An SQL exception occurred while 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.

CEPPaymentech7001: An SQL exception was caught while processing the result set for a query on cassette payments.

Severity: Error

Explanation: An SQL exception occurred while 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.

CEPPaymentech7002: An SQL exception was caught while processing the result set for a query on cassette credits.

Severity: Error

Explanation: An SQL exception occurred while 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.

CEPPaymentech7003: An SQL exception was caught while processing the result set for a query on cassette batches.

Severity: Error

Explanation: An SQL exception occurred while 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.

CEPPaymentech7004: An SQL exception was caught while processing the result set for a query on cassette accounts.

Severity: Error

Explanation: An SQL exception occurred while 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.

CEPPaymentech7005: An SQL exception was caught while processing the result set for a query on cassette payment systems.

Severity: Error

Explanation: An SQL exception occurred while 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.

CEPPaymentech7006: An SQL exception was caught while processing the result set for a query on cassette configuration.

Severity: Error

Explanation: An SQL exception occurred while 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.

CEPPaymentech8000: An SQL exception was caught while selecting the 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.

CEPPaymentech8001: An SQL exception occurred while updating the 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.

CEPPaymentech9000: 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.

CEPPaymentech9001: 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.

CEPPaymentech9002: 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.

CEPPaymentech9003: 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.

CEPPaymentech9005: An error occurred while attempting to *en_decrypt* **the data.**

Severity: Error

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

User Response: Contact your service representative.

CEPPaymentech0606: An internal error occurred: *exception text***.**

Severity: Error

Explanation: An internal error occurred in the cassette. The exception text will help IBM support to 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 10594-1785
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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligations to you.

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
- OS/400
- pSeries
- S/390
- WebSphere
- zSeries

Microsoft®, Windows, Windows NT®, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

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

SET Secure Electronic Transaction, Secure Electronic Transaction, SET, and the SET Secure Electronic Transaction design mark are trademarks and service marks 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) n 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

A

B

C

D

F

I

K

L

M

N

O

P

Q

R

S

T

U

W

Printed in U.S.A.