IBM WebSphere Commerce

Payments Cassette for BankServACH Supplement

Version 5.5

IBM WebSphere Commerce

Payments Cassette for BankServACH Supplement

Version 5.5

Note

Before using this information and the product it supports, be sure to read the general information under Appendix D, "Notices", on page 65.

Fifth Edition (June 2003)

This edition applies to version 5.5 of IBM WebSphere Commerce Payments and to all subsequent releases and modifications until otherwise indicated in new editions. 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 BankServACH 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 *IBM WebSphere Commerce Administration Guide* and *IBM WebSphere Commerce Installation Guide*. For information about how to install and configure payment cassettes, refer to those documents.

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

Conventions used in this book

This book uses the following highlighting conventions:

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

indicates information specific to the Windows® operating environment.

indicates information specific to AIX®.

indicates information specific to the Solaris Operating Environment.

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

indicates information specific to Linux.

References in this book to *workstation* platforms apply to Windows, AIX, Solaris, and Linux on Intel® platforms (not iSeries).

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 banking industry, including the following:

ACH Operator
The Automated Clearing House (ACH) Operator is the central clearing facility operated by a private organization or a Federal Reserve Bank.

Batch The batch is a collection of financial transactions grouped for administrative and record-keeping purposes.

Originating Depository Financial Institution (ODFI)
The ODFI is an institution that receives payment instructions from the Originator and forwards entries to the ACH Operator.

Originator
The originator initiates ACH entries into the payment system according to an arrangement with a Receiver.

Receiving Depository Financial Institution (RDFI)
The RDFI receives the ACH entries from the ACH Operator and posts the entries to the accounts of its depositors (Receivers).

Receiver
The receiver is the person or organization which has authorized an Originator to initiate an ACH entry to the Receiver's account with the RDFI.

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).

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 BankServACH

BankServ is a payment gateway that interfaces with the Automated Clearing House (ACH) network to support online electronic check payments.

Electronic check payments are made in a single transaction and use the automated ACH system operated by the Federal Reserve to debit and credit the appropriate accounts at different financial institutions. This transfer of funds from the debit account to the credit account is known as settlement. Settlement happens automatically after the transaction has been submitted to the ACH network. Every 24 hours, accepted transactions are sent, as a batch, to the BankServ originating bank (ODFI), where they are introduced into the ACH network. The cutoff point is defined to be the time of day where transactions for the prior 24 hours are sent to the bank (which is 2:30 p.m., PST). All transactions after the cutoff point will be sent in the following batch, occurring up to 24 hours in the future. Once the transactions are presented to the Federal Reserve, they will be settled between the debit account and the credit account. Unless voided, from the merchant standpoint, the transaction is a singular event occurring at the time of the sale, i.e., there is no separate settlement process. All BankServ transactions result in ACH Debit transactions. The BankServ Gateway does not support ACH Credit due to the fact that an ACH Credit cannot be done systematically at this time.

An electronic check transaction occurs by sending a single request that contains the buyer account details, and the corresponding transaction specific details. Once this request has been authorized by BankServ, it is batched and then sent to the ACH system for automatic settlement. This automatic settlement happens once a day, and as mentioned previously, all transactions occurring after the specific settlement time will be settled the next business day. If the transaction returns from the ACH network, it can be automatically re-presented up to two times by the BankServ ACH system. Re-presentment is most commonly used when a transaction is returned due to insufficient funds. In this scenario, a merchant will typically re-present the transaction in the hopes that the additional funds have been placed into the checking account. BankServ allows the merchant to specify, based on the return code, to automatically re-present a transaction or not (if re-presentment is not automatic, then the merchant must manually deal with the returned transaction). For example, a merchant may set up insufficient funds returns to cause automatic re-presentment, but an account closed would be a ″hard return″ requiring manual repair. This gives the merchant the flexibility to define how they want to handle the multitude of return reasons. The number of re-presentments of a transaction to the bank for collection is configurable by the merchant at setup time and is not transaction specific. Transactions should be debited within 48 hours.

Figure 1 on page 2 shows the major components involved with the Cassette for BankServACH in a WebSphere Commerce Payments environment.

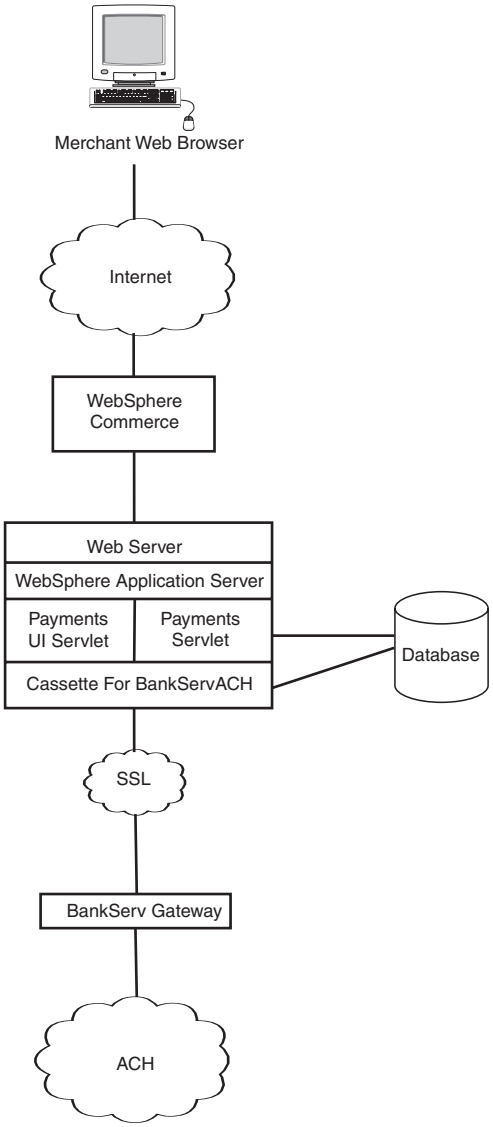

Figure 1. High level overview of BankServ processing

BankServ merchant registration

The BankServ registration process is a manual process between BankServ and the merchant in which the merchant provides:

- A merchant contract.
- A set up form.
- A voided check.
- Financial data.
- The IP address of the machine which will be accessing the server. When accessing the BankServ gateway from behind a firewall using socks or proxy, the merchant must register each of the outgoing IP address(es) of the proxy.

Once BankServ receives and verifies the information, they underwrite the merchant. The BankServACH cassette assumes that this merchant registration process has occurred prior to attempted use of the cassette.

Merchant PIN

As part of merchant registration, BankServ will assign the Merchant a Merchant PIN. The merchant is identified by the merchant PIN. A merchant PIN is a string of up to 200 bytes long which BankServ has issued. This PIN number is associated with the IP addresses of the merchant servers. If a presented merchant PIN fails to originate from a listed merchant IP address, the transaction will be rejected.

Communications

Communication with the BankServ payment gateway is based on the HTTPS protocol, which uses name/value pairs. TCP/IP is the underlying network architecture. Requests to and responses from the BankServ server take the form of name/value pairs.

General flow

Upon receipt of a Deposit request from the merchant, the cassette establishes a connection to the BankServ payment gateway, sends the transaction information to the BankServ gateway, and synchronously receives the response from BankServ. The connection with the BankServ gateway is not long-lived (i.e., a connection is established for each deposit transaction).

Communications via HTTPS

HTTPS is a secure/encrypted version of HTTP. Since all information is sent in an encrypted format, HTTPS is much less vulnerable to network sniffing thus greatly reducing the risks of clandestine theft of information. In addition, HTTPS uses a certificate based identification system to reduce faked identity connections.

Supported functions

The BankServ gateway supports both electronic check and credit card transactions for either known or unknown buyers, only some of which are supported by the BankServACH Cassette.

The following table shows the five types of transactions supported by the BankServ gateway, and whether or not the Cassette for BankServACH supports the transaction:

Table 1. Transactions supported by the BankServ Gateway

Transaction	Description	Supported by BankServACH Cassette?
Electronic Check	Submits transaction into BankServ batch to be entered into the ACH system for automatic settlement (which occurs once a day)	Yes
Credit Card	Credit card authorization transaction	No
Credit Card Settlement	Credit card settlement transaction	No
Transaction Report	Contains a summation of all transactions for a specified merchant PIN on a specified date	No

Table 1. Transactions supported by the BankServ Gateway (continued)

Transaction	Description	Supported by BankServACH Cassette?
Status Transaction	Allows a client to query the status of a specified transaction	Yes (internal use only)

Electronic check and status transactions are defined by a ″rule_set″. The following table lists the rule_sets and indicates which of these rule_sets will be supported by the Cassette for BankServACH. Based on these existing rules, the cassette can only support ACH Debit transactions with a Standard Entry Class Code (SEC Code) of ″WEB.″ ″WEB″ Transactions are consumer ACH Debit transactions authorized over the Internet. These non-recurring debit entries are initiated by the Originator based on authorization and account information obtained from the consumer.

Table 2. BankServ rule sets

Rule set	Meaning	Supported by BankServACH Cassette?
ACHDEBIT	ACH Transaction without credit card information for unknown buyer	Yes
ACHDEBEX	ACH Transaction without credit card information for known buyer	No
TRXSTAT	Transaction status	Yes
TSCC	Credit card transaction for unknown buyer	No
TCCEX	Credit card transaction for known buyer	No
TCCNORQ	Used for internal testing	No
ACHVOID	Void an ACH transaction	Yes

Sensitive data protection

As an option, you can prevent sensitive financial data 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 BankServACH:

Table 3. . Sensitive data processed by Cassette for BankServACH

Data	How data is protected
$CHECKROUTINGNUMBER	Buyer's check routing number. The entire value is masked with asterisks.
$CHECKINGACCOUNTNUMBER	Buyer's checking account number. 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 4.

Table 4. 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 5. 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 (not applicable to this cassette) • 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 (not applicable to this cassette) • 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. BankServ 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 BankServACH presents the BankServ services through the WebSphere Commerce Payments's object model and API set. In addition, cassette-specific behaviors and requirements are discussed.

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

- AcceptPayment creation of orders only. Wallet-driven purchases are not supported.
- Traditional payment-oriented commands.
- A single batch for each account.

A BankServ purchase example

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

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

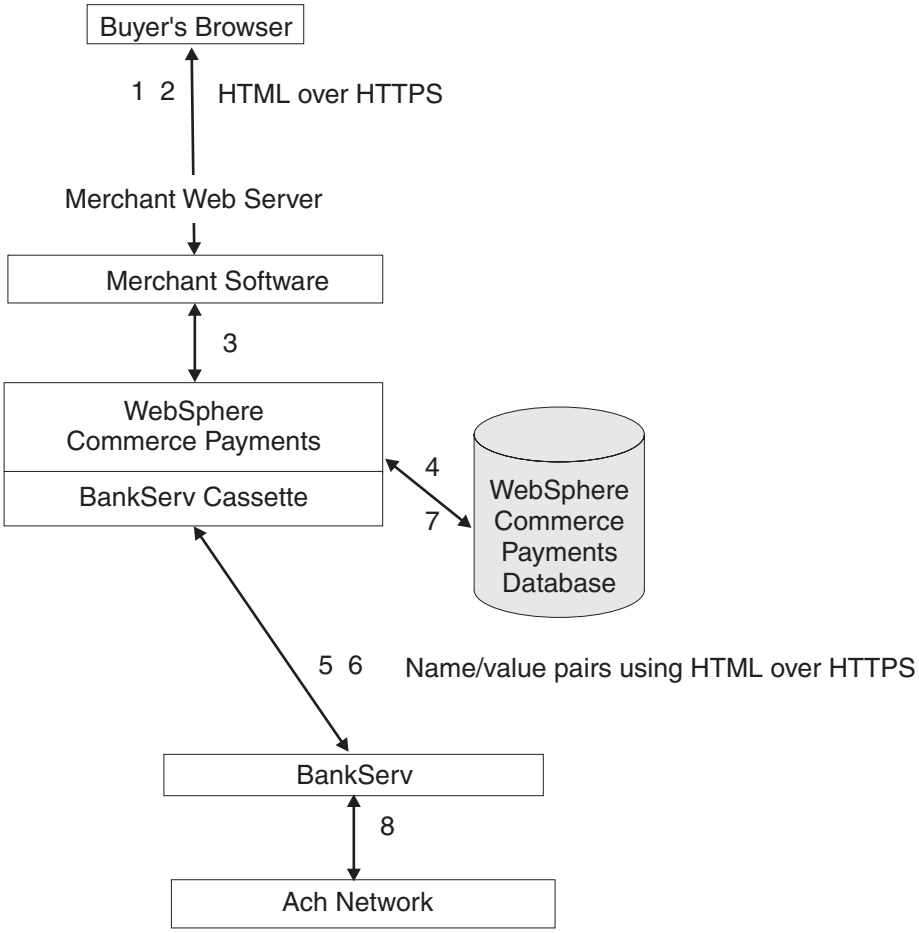

Figure 2. Processing overview

1. The merchant software presents the shopping catalog to the buyer. The buyer shops and then initiates a purchase, typically by clicking the ″Buy″ button on the page.
2. The merchant software then requests the electronic check information from the buyer over a secure (typically SSL protected) connection.
3. The merchant invokes WebSphere Commerce Payments with an AcceptPayment command with AutoAuth and AutoDeposit.
4. WebSphere Commerce Payments and the Cassette for BankServACH update the WebSphere Commerce Payments database with the information needed to perform the transaction.
5. The Cassette for BankServACH builds the Electronic Check request and sends it to the BankServ payment gateway using the HTTPS protocol.
6. The BankServ payment gateway validates the data and returns the transaction status to the cassette.
7. The Cassette for BankServACH updates the WebSphere Commerce Payments database to reflect this status.
8. At a predetermined time, the BankServ host sends all the requests it has accumulated to the ACH processor. Note that no indication of this action is passed back to the cassette.

WebSphere Commerce Payments object model implementation

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

Administration objects

The WebSphere Commerce Payments administration objects are the entities that comprise the system and merchant configuration under which all financial transactions will be performed. Refer to the *WebSphere Commerce Payments Programming Guide and Reference* for a description of the WebSphere Commerce Payments administration objects. The Cassette for BankServACH augments three of the framework administration objects with its own attributes. BankServ Administration objects are described in detail in Chapter 7, "Object reference", on page 39.

Cassette Admin object

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

Account Admin object

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

PaySystem Admin object

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

The Cassette for BankServACH allows the merchant to define a single account in the PaySystemAdmin object.

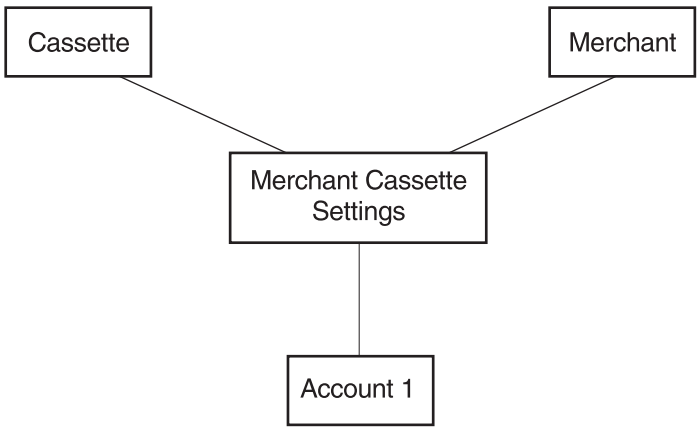

Figure 3. BankServACH PaySystem

Financial objects

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

- Order objects
- Payment objects
- Batch objects

For details on how the Cassette for BankServACH extends these payment objects, see Chapter 7, "Object reference", on page 39. For descriptions of the payment objects and for programming information, see the *WebSphere Commerce Payments Programming Guide and Reference*.

Cassette-specific characteristics and behaviors

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

Retry parameters

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

- Connect Retries
- Connect Timeout
- Read Timeout
- Immediate 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 BankServ cassette icon, then select Advanced Settings) or through the ModifyCassette API command. For more information on the ModifyCassette command, see "ModifyCassette" on page 36.

Connectivity information and retry information will be stored as cassette extensions.

- The $CONNECTRETRIES protocol data is related to establishing the connection. It indicates the number of attempts that will be made to establish a connection with the BankServ gateway.
- The $CONNECTTIMEOUT protocol data indicates the number of seconds to wait while attempting a connection to the BankServ gateway.

The following data relates to communications after the connection has been established.

- The $READTIMEOUT protocol data indicates the number of seconds to wait for a response while communicating with the BankServ gateway.
- The $MAXRETRIES protocol data indicates the maximum number of immediate retries that will occur before the delayed retries take effect.

- The $MAXATTEMPTS protocol data indicates the number of delayed retries that will be attempted.
- The $ATTEMPTINTERVAL protocol data indicates the number of seconds to wait before retrying the communications.

Under direction of these parameters, the cassette will attempt to recover as described in the following paragraphs.

For each command that requires communication with the BankServ server, a HTTPS connection must be established between WebSphere Commerce Payments and the BankServ server. If the initial connection to the server for a given command cannot be established, the cassette can immediately retry the connection as many times as desired. The number of immediate connection retries is specified by the ConnectRetries parameter (this parameter is called $CONNECTRETRIES on the ModifyCassette API command).

Once a connection is established to the server, the cassette sends an appropriate request message to the server and then waits for a predetermined period of time for a response. 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 retry the operation $MAXRETRIES times. If the communication is unsuccessful after all immediate retries have been attempted, the cassette enters ″delayed retry″ logic.

Specifically, delayed retries work as follows:

- The cassette will return a return code that indicates the operation is pending (PRC_OPERATION_PENDING).
- The request message is queued and waits a predetermined amount of time as specified by the cassette setting called AttemptInterval (this parameter is called $ATTEMPTINTERVAL on the ModifyCassette API command).
- Once the attempt interval expires, the request is removed from the internal queue and is retried.
- The process of queuing the request and retrying the operation is repeated until the request is completed or until the maximum number of communication attempts is reached. The maximum number of communication attempts is specified by the MaxAttempts (this parameter is called $MAXATTEMPTS on the ModifyCassette API command) value in the cassette settings.

Security considerations

On workstation systems, the BankServ system uses 128 bit SSL encryption. In order to support this, the cassette uses the `IBMJSSE.JAR` file provided by the IBM version of the Java™ Secure Socket Extension (JSSE). Three actions are required to use this file:

- A protocol package must be specified.
- A ″provider″ must be defined.
- The location of the certificates must be defined.

Within this jar file is a reference implementation of the protocol which has been configured within the cassette by setting the system property **java.protocol.handler.pkgs** to **com.ibm.net.ssl.internal.www.protocol**. The ″provider″ has been specified within the cassette via the following call:

```
Security.addProvider(new com.ibm.jsse.JSSEProvider());
```

Finally, the required certificates can be found in the `cacerts` file in the `<java_home>jre/lib/security/` directory where `<java_home>` is the directory referenced by the JAVA_HOME environment variable. If these system provided certificates are satisfactory, no action is required.

If the system-provided certificates are not satisfactory, you can provide your own set of certificates by creating a `jssecerts` file in the `cacerts` file directory. The implementation firsts looks for a `jssecerts` file in this directory. If found, this file is used. If not found, the `cacerts` file is used. The `jssecerts` file can be created by using either the graphical IBM Key Management tool (ikeyman) or the command line tool (keytool). Refer to the ″Tools for managing certificates and keys″ section of the WebSphere documentation for instructions.

Note: For 400 systems, refer to the latest README file, **readme.BankServACH.html**, accessed through documentation links from the WebSphere Commerce Payments Web site: **http://www.ibm.com/software/webservers/commerce/payments/support.html** and on the CD-ROM containing the Cassette for BankServACH.

Performance considerations

The Cassette for BankServACH uses service threads when communicating with the BankServ host. Since there is a limited number of service threads available (three by default), resources may not be available to handle a communications request. If a service thread is not available, the request waits until one becomes available. Since service threads are also used by the cassette for retries, and by other cassettes, the number of service threads may have to be increased.

On a lightly loaded system, with no other cassettes using the service queue and a quickly responding server, the default value may suffice. On a heavily loaded system with many concurrent requests, a larger value will be required. This value is specified in the Payments instance's configuration through the WebSphere Commerce Configuration Manager (spool size field).

Cassette for BankServACH payment command summary

Table 6 summarizes the way the Cassette for BankServACH 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:

- ″Not supported by cassette″, meaning the cassette does not support that particular command. These commands will always receive return codes PRC_COMMAND_NOT_SUPPORTED, RC_NONE.
- ″Handled by WebSphere Commerce Payments; no message sent″, meaning that the command is processed completely within WebSphere Commerce Payments without communicating with the BankServ payment gateway.
- In any other case, the primary BankServ command (or commands) used to accomplish the function will be shown.

Table 6. Payment command summary

API Command	BankServ message
AcceptPayment	Handled by WebSphere Commerce Payments; no message sent

Table 6. Payment command summary (continued)

API Command	BankServ message
AcceptPayment w/AutoApprove	Handled by WebSphere Commerce Payments; no message sent
AcceptPayment w/AutoApprove and AutoDeposit	Electronic Check Request; rule set = ACHNOCC
Approve	Handled by WebSphere Commerce Payments; no message sent
Approve w/Deposit	Electronic Check Request; rule set = ACHNOCC
ApproveReversal (full reversal)	Handled by WebSphere Commerce Payments; no message sent
ApproveReversal (partial reversal)	Handled by WebSphere Commerce Payments; no message sent
CancelOrder	Handled by WebSphere Commerce Payments; no message sent
CloseOrder	Handled by WebSphere Commerce Payments; no message sent
Deposit	Electronic Check Request; rule set = ACHNOCC
DepositReversal	Electronic Check Request; rule set = VDACH
Refund	Not supported by cassette
RefundReversal	Not supported by cassette
BatchOpen	Not supported (cassette opens batch implicitly, as needed)
BatchClose	Handled by WebSphere Commerce Payments; no message sent
BatchPurge	Not supported by cassette
DeleteBatch	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 7. Summary of State Changes

API command	Object state
AcceptPayment	ORDER_ORDERED
AcceptPayment w/AutoApprove	ORDER_ORDERED PAYMENT_APPROVED
AcceptPayment w/AutoApprove and AutoDeposit	ORDER_ORDERED PAYMENT_DEPOSITED
Approve	PAYMENT_APPROVED
Approve w/Deposit	PAYMENT_DEPOSITED
ApproveReversal (full reversal)	PAYMENT_VOID

Table 7. Summary of State Changes (continued)

API command	Object state
ApproveReversal (partial reversal)	PAYMENT_APPROVED
Cancel Order	ORDER_CANCELED
CloseOrder	ORDER_CLOSED
Deposit	PAYMENT_DEPOSITED
DepositReversal	PAYMENT_APPROVED
Deposit attempt results in delayed retry due to communications failure	PAYMENT_PENDING
Refund	Not supported
RefundReversal	Not supported
BatchOpen	Not supported. Implicit open moves batch to BATCH_OPEN
BatchClose	BATCH_CLOSED PAYMENT_CLOSED
BatchPurge	Not supported
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 BankServACH 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 BankServACH Version 5.5 with earlier versions of the WebSphere Commerce Payments framework.

Before you can configure the Cassette for BankServACH, 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 BankServ 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 BankServACH. 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 BankServACH 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 BankServACH:

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

   After the orders are created, you are ready to begin the following payment-processing tasks that merchants typically perform on a daily basis:
9. Approve orders and deposit payments.
10. Settle batches.
11. View daily batch totals.

Before starting this tutorial

There are some configuration steps that require information from your BankServACH merchant account or your financial institution to do this tutorial. This information consists of your **Merchant PIN**, which was provided as part of the BankServ merchant registration process. The Merchant PIN is entered as one of the settings in the Merchant Cassette Settings page. Refer to the merchant registration process section in Chapter 2 for information on establishing the registration process.

Step 1: Accessing the WebSphere Commerce Payments user interface

Our first task is enabling a merchant to use the Cassette for BankServACH. 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 for SOCKS

Windows AIX Linux Solaris 400 During this tutorial, your WebSphere Commerce Payments will attempt to send messages through the Internet to the BankServ gateway. If your computer is behind a firewall and must use a SOCKS server to access sites outside of your internal network, then you may have to perform certain steps to enable the Cassette for BankServACH for SOCKS.

400 If you have already configured the iSeries TCP/IP support to use a SOCKS server, the iSeries will automatically route connections to the SOCKS server. Thus, you will not have to change the configuration for the Cassette for BankServACH. For more information about configuring the iSeries TCP/IP SOCKS support, see the iSeries Information Center and online library at: **http://publib.boulder.ibm.com.pubs/html/as400/infocenter.htm**. If you choose not to configure SOCKS within iSeries TCP/IP support, then perform the following steps to enable the Cassette for BankServACH for SOCKS.

Notes:

1. If you already have access to sites on the Internet, then you may skip this step.
2. If you use a SOCKS server, you must register the IP address of the server with BankServ.

To configure the cassette for SOCKS, do the following:

1. Click **Cassettes** in the navigation frame.
2. Click the **BankServACH** cassette icon.

3. At the next page, you will see several entry fields that allow you to tailor the way the Cassette for BankServACH communicates with the BankServ gateway. Two of these entry fields are related to SOCKS configuration. Enter the following:

Table 8. SOCKS configuration fields

Field	Description
BankServ URL	The URL to access the BankServ gateway.
SOCKS Host Name	The fully qualified TCP host address for the SOCKS server. **Note:** Enter this name carefully (for example, *mycomputer.city.company.com* is fully qualified).*
SOCKS Port Number	The TCP port of your installation SOCKS server.
*If you enter the wrong SOCKS host name, the Payments application server may fail when a Payments transaction is attempted and you will not be able to use the Payments GUI. If this happens, stop and restart the Payments instance, return to this page, and enter the correct SOCKS host name.	

4. Click **Update** to update your cassette configuration.
5. Click **Stop Cassette** to stop the cassette.
6. 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.

Table 9. Merchant settings: Create a merchant

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

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 **BankServACH**.
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 10. 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 BankServACH, 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 BankServACH** icon in the Test Store.
3. From the Cassette for BankServACH page, click **Merchant Cassette Settings**.

4. At the next page, you will be prompted to enter the following information:

Table 11. Merchant Cassette Settings

Field	Description
Merchant PIN	This is the 1–200 character string (PIN) that was assigned to the merchant by BankServ.

5. Click **Update** to update the merchant cassette settings.

Step 7: Creating an account

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

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; however, the Cassette for BankServACH has a limit of one account per merchant. For the purposes of this tutorial, you will create one account for the Cassette for BankServACH.

To create an account, do the following:

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

Table 12. Add an account settings

Field	Description
Account name	Enter *BankServACH 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. Note that if account number 1 is already defined for the Test Store merchant, then you may choose another number. We will assume for the remainder of this tutorial, however, that you are using account number 1.
Financial Institution name	Enter *BankServ 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 the time in which all open batches should be closed. This is an optional parameter that enables the Merchant to specify a time when any currently open batches will be automatically closed on a daily basis. For information about how to enter the time format, see the online help for this field.

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

Step 8: Creating orders using the Sample Checkout

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

- Merchant-specific administration functions
- All payment processing functions

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

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

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

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

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

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

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

   WAS_installdir/installedApps/*host_name*/*Payments_instance*_Commerce_Payments_App.ear/SampleCheckout.war

   400 For iSeries, the directory path is

   /QIBM/UserData/WebAS5/Base/*WAS_instance*/installedApps/*node_name*/*Payments_instance*_Commerce_Payments_App.ear/SampleCheckout.war

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

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

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

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

3. Save the file.

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

   If you are using SSL with the Payments instance, use https:// instead.
5. At the Sample Checkout page enter the following (note that all fields except the second street address line and email Address are required for the tutorial).

Table 13. Sample Checkout fields for Cassette for BankServACH

Field	Description
Merchant number	Enter any number to represent a Merchant number.
Order number	Enter any number to represent an Order number.
Amount	Enter any amount to represent the total numeric amount of the order.
Currency	Enter *US dollar*. The currency used to place this order.
Payment method	Choose *BankServACH* as the payment method.
Checking Account Number	Enter the checking account number of the buyer.
Check Routing Number	Enter the check routing number. This is the 9-digit number on the bottom left hand corner of the check, to the left of the account number, which identifies the paying bank that issued the check.
Name	Enter the name of the buyer or check holder.
Street address	Enter the street address of the location of the check holder.
City	Enter the name of the city of the location of the check holder.
State	Enter the name or abbreviation of the state of provice of the location of the check holder.
Zip Code	Enter the zip/postal code of the check holder.
Country/Region	Enter the 2-character country or region code of the location of the check holder. For example, enter US for the United States. See Appendix A, “Country codes and state codes”, on page 45 for values.
Phone	Enter the 9-digit phone number of the check holder (area code and number with no hyphens or spaces).
email Address	Enter the e-mail address of the check holder (optional).
Note: When the BankServACH payment method is selected, additional fields are displayed to accept check holder information (such as address information commonly used in North America, which is shown in this table).	

6. Click **Buy**.

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

Step 9: Approving orders with the Sale function

The sale function allows you to approve and deposit payments with one command. The Cassette for BankServACH 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 page displays the following information:

Table 14. 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 15. Payment details

Field	Description
Merchant	The merchant name.
State	The current state of the Payment (Closed).
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 (BankServACH Account).
Order URL	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 processor's retrieval reference number received when the payment was approved. Not always present. Varies by processor.
Payment Type	The payment cassette or protocol used for this order (BankServACH).
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 BankServACH.
ACH Reference Number	The ACH reference number associated with the transaction, if available.
Response Code	The approval code returned in the Electronic Check response: • AA - Transaction successfully processed. • SF - System failure. Call BankServ. • RE - Transaction rejected due to invalid data, and/or missing values. • DE - Transaction declined due to business reasons.
Transaction Reference Number	The identifier by which this payment transaction is known to BankServ. You will need this identifier if you ever need to contact BankServ about a problem with this payment.
Merchant Transaction ID	The identifier generated by the Cassette for BankServACH for the transaction.
Transaction Status	The status code from the last command sent to the BankServ gateway for this payment.

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

Approve

Approval without deposit is performed through the same windows as the Sale function (that is, the Approve or Order windows). Instead of 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. 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, do the following:

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 undeposited 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 25. Click **Deposit** at the bottom of this screen to deposit all or part of the approved amount.

3. On the Deposit Payment screen, change the deposit amount to a value less than the full approval amount and click **Deposit**.
4. When the deposit has been processed, you will return to the Payment page, which will be updated with the new deposit amount.

Step 10: Settling batches

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

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

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

Table 16. Batch search fields

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

Table 16. Batch search fields (continued)

Field	Description
Account	The account under which this order is being processed (BankServACH Account). If more than 500 accounts have been defined, type the account number in the entry field.

3. Click **Search**.

   Tip: You can also use the before and after fields to narrow search results by excluding certain batches from the search. For example, you could search on all batches opened before 08/01/2003 and after 08/15/2003 thus excluding batches opened between 08/02/2003 and 08/14/2003.
4. Click the batch number to view information about the batch.
5. Click **Batch Details** to see a detailed listing of all payments in this batch. You will see the four payments you just created.
6. Click **Settle** to settle the batch. When processing is complete the settle status is displayed.

Step 11: Viewing batch totals

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

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

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

The Daily Batch Totals report computes the totals for all batches that were closed on the date specified on the Search page. These totals are computed on a per-currency basis, so there is one line per currency. These totals cover all payments made for all payments (not just those made through the Cassette for BankServACH).

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

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

Chapter 5. Cassette for BankServACH 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 BankServACH:

- `WC51_BankServACH.profile`

  If you used the default instance name of **demo**, the profile is stored in the following directory: *WC_installdir*/instances/demo/xml/payment.
- `SampleCheckoutBankServACH.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 SampleCheckout application, the application requires that the Cassette for BankServACH profile be named `SampleCheckoutBankServACH.profile`.

  Initially, the `SampleCheckoutBankServACH.profile` is installed in directory path *Payments_installdir*/cassettes/BankServACH/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.

If necessary, you can edit the profile to set certain parameters, such as APPROVEFLAG and DEPOSITFLAG. (These flags are described in "AcceptPayment" on page 33.) 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 BankServ-specific protocol parameters
- Any special notes related to the Cassette for BankServACH handling of framework parameters.

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

Cassette for BankServACH commands

The following section outlines information specific to the BankServ 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 framework order and a BankServACH cassette order to be created. If the ApproveFlag is set to ″1″, then a framework payment and BankServACH cassette payment are also created. If the DepositFlag is set to ″1″, then the Payment is added to the currently open batch (if there isn't a batch open, one will be created implicitly) and an Electronic Check transaction is sent to the BankServ payment gateway.

Table 17. Required keywords for AcceptPayment command

Name	Value
$BUYERNAME	Specifies the name of the buyer. The value is specified as a 1–80 character string.
$STREETADDRESS	Specifies the first line of the street address. The value is specified as a 1–50 character string.
$CITY	Specifies the city. The value specified is a 1–50 character string.
$STATEPROVINCE	Specifies the 2 character state abbreviation. See Appendix A, "Country codes and state codes", on page 45 for a list of valid state codes.
$POSTALCODE	Specifies the 5 or 9 digit zip code.
$COUNTRYCODE	Specifies the 2 character country code of the buyer. See Appendix A, "Country codes and state codes", on page 45 for a list of valid country codes.
$PHONE	Specifies the phone number of the buyer. The value specified is a 1–10 digit number.
$CHECKROUTINGNUMBER	Specifies the nine digit check routing number of the buyer. The value specified is a 9 digit number.

Table 17. Required keywords for AcceptPayment command (continued)

Name	Value
$CHECKINGACCOUNTNUMBER	Specifies the checking account number of the buyer. The value specified is a 1–17 digit number.

Table 18. Optional keywords for AcceptPayment command

Name	Value
$STREETADDRESS2	Specifies the second line of the street address. The value specified is a 1–50 character string.
$EMAILADDRESS	Specifies the e-mail address of the buyer. The value specified is a 1–49 character string.

Approve

The Approve command causes a framework payment and BankServACH cassette payment to be created. If the DepositFlag is set to ″1″, then the Payment is added to the currently open batch (if there isn't a batch open, one will be created implicitly) and an Electronic Check transaction is sent to the BankServ payment gateway.

ApproveReversal

The ApproveReversal command causes the specified payment to be retrieved and reversed. This command is a ″local operation″ only. There are no messages that flow to the BankServ payment gateway as a result of this command. The BankServACH cassette will support both full and partial reversals. This works as follows:

- If the AMOUNT in the request is ″0″, then a full reversal is done and the payment moves into PAYMENT_VOID state.
- If the AMOUNT is non-″0″, then the payment amount is updated with an amount equal to the amount in the ApproveReversal request. The state of this newly updated payment stays in the PAYMENT_APPROVED state.

BatchOpen

This command is not supported since all batches are opened implicitly. If this command is issued with PAYMENTTYPE set to ″BankServACH″ the command will fail with PRC_COMMAND_NOT_SUPPORTED and RC_NONE.

BatchPurge

The BatchPurge command is not supported. If this command is issued with PAYMENTTYPE set to ″BankServACH″ the command will fail with PRC_COMMAND_NOT_SUPPORTED and RC_NONE.

BatchClose

The BatchClose command is a local operation only since settlement occurs outside the scope of cassette. The way settlement occurs is that every day at 2:30 p.m. PST, all transactions that have been received and authorized by the BankServ gateway in the previous 24 hours are sent, as a batch, to the BankServ originating bank (ODFI), where they are introduced into the ACH network. The cassette is not

involved in this process. The cassette does support the automatic closing of the currently open batch (see “CreateAccount” for details), and it is expected that merchants will configure this to be 2:30 p.m. PST.

When a BatchClose command is received, the current batch will be put in CLOSED state (moving all associated payments to CLOSED state as well) and will no longer accept new transactions.

CassetteControl

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

- PRC_COMMAND_NOT_SUPPORTED
- RC_NONE.

CloseOrder

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

CreateAccount

Table 19. Optional keywords for CreateAccount command

Name	Value
$BATCHCLOSETIME	In the format HHMM, the (local) time of day an automatic BatchClose is to occur.

CreatePaySystem

Table 20. Required keywords for CreatePaySystem command

Name	Value
$MERCHANTPIN	Merchant assigned pin. The value specified is a 1–200 character string.

DeleteBatch

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

Deposit

The Deposit command causes the specified payment to be added to the currently open batch. If a batch is not currently open, one is created. This command causes an Electronic Check transaction to be sent to the BankServ gateway. If the operation is successful, the payment moves from Approved state to Deposited state.

DepositReversal

The DepositReversal command causes the specified payment to be removed from the currently open batch. In addition, it causes the transaction to be removed from the BankServ batch. There is no concept of partial reversals; you are either adding transactions to the batch or you are removing transactions from the batch. Removing a transaction from the batch is accomplished by sending an Electronic Check transaction to the BankServ gateway with the rule set ID = ″ACHVOID″. If

the transaction is successful, then the payment is removed from the batch (both the cassette batch and the BankServ batch) and the payment moves to the PAYMENT_APPROVED state. This command is valid for payments in DEPOSITED state.

ModifyAccount

Table 21. Optional keywords for ModifyAccount command

Name	Value
$BATCHCLOSETIME	In the format HHMM, the (local) time of day an automatic BatchClose is to occur.

ModifyCassette

Table 22. Optional keywords for ModifyCassette command

Name	Value
$READTIMEOUT	Number of seconds to wait while communicating with the BankServ payment gateway.
$CONNECTTIMEOUT	Number of seconds to wait while attempting connection to the BankServ payment gateway.
$CONNECTRETRIES	Number of times to retry a connection attempt to the BankServ payment gateway.
$MAXRETRIES	When a communications error occurs (i.e.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.
$ATTEMPTINTERVAL	When a communications error occurs, the number of seconds to wait before trying the next set of (delayed) retries.
$MAXATTEMPTS	Maximum number of delayed retry sets. Default is 3.
$SOCKSHOSTNAME	TCP Host Address for socks server (0–254 character string). Specify the fully qualified host name carefully. For example, *mycomputer.city.company.com* is fully qualified. If you enter the wrong SOCKS host name, the Payments application server may fail when a Payments transaction is attempted and you will not be able to use the Payments GUI. (To recover, stop and restart the Payments instance.)
$SOCKSPORTNUMBER	Socks port number
$BANKSERVURL	The URL to access the BankServ payment gateway.

ModifyPaySystem

Table 23. Optional keywords for ModifyPaySystem command

Name	Value
$MERCHANTPIN	Merchant assigned PIN.

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 ″BankServ″ the command will fail with:

- PRC_COMMAND_NOT_SUPPORTED and RC_NONE.

Refund

This command is not supported since BankServ ACH transactions do not have the concept of refunds. If this command is issued with PAYMENTTYPE set to ″BankServACH″ the command will fail with PRC_COMMAND_NOT_SUPPORTED and RC_NONE.

RefundReversal

This command is not supported since BankServ ACH transactions do not have the concept of refunds. If this command is issued with PAYMENTTYPE set to ″BankServACH″ the command will fail with PRC_COMMAND_NOT_SUPPORTED and RC_NONE.

Chapter 7. Object reference

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

Financial objects used by Cassette for BankServACH

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

BankServACH Order

Table 24. Cassette properties that belong to a PSOrder Object

Field name	Description
buyerName	Specifies the Buyer's name.
streetAddress	Specifies the Buyer's street address.
streetAddress2	Specifies the Buyer's street address (2nd line).
city	Specifies the Buyer's city.
stateProvince	specifies the Buyer's state.
postalCode	Specifies the Buyer's postal code.
countryCode	Specifies the Buyer's country code.
phoneNumber	Specifies the Buyer's phone number.
emailAddress	Specifies the Buyer's email address.
checkNumber	Specifies the Buyer's check number. The checkNumber is generated by the cassette based on the order number.

BankServACH Order object XML example

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
  <OrderCollection size="1" withCredits="0" withPayments="0">
    <PSOrder ID="O:123456789:1" amount="1200" amountExp10="-2" approvesAllowed="">
     1" currency="840" merchantAccount="1" merchantNumber="123456789" merchantOriginated="1"
     numberOfCredits="0" numberOfPayments="0" orderNumber="1" paymentType="BankServACH"
     state="order_ordered" timeStampCreated="987618551000" timeStampModified=
     "988143427000" unapprovedAmount="0">
      <CassetteExtensionObject>
        <CassetteProperty propertyId="stateProvince" value="NC">
        </CassetteProperty
        <CassetteProperty propertyId="city" value="Raleigh">
        </CassetteProperty
        <CassetteProperty propertyId="phoneNumber" value="5433820">
        </CassetteProperty
```

```
      <CassetteProperty propertyId="countryCode" value="US">
      </CassetteProperty>
      <CassetteProperty propertyId="buyerName" value="Emily McMullen">
      </CassetteProperty>
      <CassetteProperty propertyId="streetAddress" value="432 Main Street">
      </CassetteProperty>
      <CassetteProperty propertyId="postalCode" value="27613">
      </CassetteProperty>
      <CassetteProperty propertyId="checkNumber" value="1">
      </CassetteProperty>
    </CassetteExtensionObject
  </PSOrder>
```

BankServACH Payment

Table 25. Cassette properties that belong to a PSPayment Object

Field name	Description
approvalCode	The ACH reference number associated with the transaction (if available).
authResponseCode	The approval code returned in the Electronic Check response. Responses can be one of the following: **AA** - Transaction successfully processed **SF**- System Failure. Call BankServ **RE**- Transaction rejected due to invalid data, and/or missing values **DE**- Transaction declined due to business reasons
authResponseMsg	A response message that may contain additional error information if the Request was not successful.
trxReferenceNum	The BankServ reference number for the transaction.
merchantTrxID	The cassette generated transaction ID.

Payment object XML example

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="2" primaryRC="0" secondaryRC="0">
  <PaymentCollection size="2" withOrders="0">
      PSPayment ID="P:123456789:1:1" amountExp10="-2" approveAmount="1200"
      currency="840"depositAmount="0" merchantAccount="1" merchantNumber="123456789"
      orderNumber="1" paymentNumber="1" paymentType="BankServACH" state="payment_declined"
      timeStampCreated="987618619000" timeStampModified="988143427000">
      <CassetteExtensionObject>
        <CassetteProperty propertyId="authResponseCode" value="SF">
        </CassetteProperty>
        <CassetteProperty propertyId="paymentNumber" value="1">
        </CassetteProperty>
        <CassetteProperty propertyId="orderNumber" value="1">
        </CassetteProperty>
        <CassetteProperty propertyId="merchantNumber" value="123456789">
        </CassetteProperty>
        <CassetteProperty propertyId="authResponseMsg" value="Test 1200 not found.">
        </CassetteProperty>
        <CassetteProperty propertyId="merchantTrxID" value="11">
        </CassetteProperty>
      </CassetteExtensionObject>
    </PSPayment>
```

BankServACH Batch

Table 26. Cassette properties that belong to a PSBatch Object

Field name	Description
currencyCode	The currency code of the transactions in the batch.

Table 26. Cassette properties that belong to a PSBatch Object (continued)

Field name	Description
batchDate	Batch ID, in the form of a MMDD date. The date corresponds to the day that the batch is closed.

Batch object XML example

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
  <BatchCollection size="1" withCredits="0" withPayments="1">
    <PSBatch ID="B:123456789:1" batchNumber="1" batchStatus="batch_balanced" \
        foceAllowed="0" merchantAccount="1" merchantControl="1" merchantNumber="123456789
        paymentType="BankServACH" purgeAllowed="0" state="batch_closed"
        timeStampClosed="88143801000" timeStampModified="988143801000"
        timeStampOpened="987618918000">
      <BatchTotalCollection size="1">
        <PSBatchTotal amountExp10="-2" creditAmount="0" currency="840" numberOfcredits="0"
         numberOfPayments="1" paymentAmount="5000">
        </PSBatchTotal>
      </BatchTotalCollection>
      <PaymentCollection size="1" withOrders="0">
        <PSPayment ID="P:123456789:2:1" amountExp10="-2" approveAmount="5000"
            batchNumber="1"currency="840" depositAmount="5000" merchantAccount="1"
            merchantNumber="123456789" orderNumber="2" paymentNumber="1" paymentType="BankServ"
            referenceNumber="7788" state="payment_closed" timeStampCreated="987618679000"
            timeStampModified="988143801000">
          <CassetteExtensionObject>
            <CassetteProperty propertyId="authResponseCode" value="AA">
            </CassetteProperty>
            <CassetteProperty propertyId="paymentNumber" value="1">
            </CassetteProperty>
            <CassetteProperty propertyId="orderNumber" value="2">
            </CassetteProperty>
            <CassetteProperty propertyId="merchantNumber" value="123456789">
            </CassetteProperty>
            <CassetteProperty propertyId="authResponseMsg"
                value="Authorized by our default authorizer">
            </CassetteProperty>
            <CassetteProperty propertyId="approvalCode" value="CODE 12">
            </CassetteProperty>
            <CassetteProperty propertyId="trxReferenceNum" value="7788">
            </CassetteProperty>
            <CassetteProperty propertyId="merchantTrxID" value="21">
            </CassetteProperty>
          </CassetteExtensionObject>
        </PSPayment>
      </PaymentCollection>
      <CassetteExtensionObject>
        <CassetteProperty propertyId="currencyCode" value="840">
        </CassetteProperty>
        <CassetteProperty propertyId="batchDate" value="0424">
        </CassetteProperty>
      </CassetteExtensionObject>
    </PSBatch>
  </BatchCollection>
</PSApiResult>
```

Administrative objects used by Cassette for BankServACH

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

- CassetteAdmin
- AccountAdmin
- PaySystemAdmin

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

CassetteAdmin

Table 27. Cassette properties that belong to a PSCassette Object

Field name	Description
readTimeout	Number of seconds to wait while communicating with the BankServ payment gateway.
connectTimeout	Number of seconds to wait while attempting connection to the BankServ payment gateway.
connectRetries	Number of times to retry a connection attempt to the BankServ payment gateway.
maxRetries	When a communications error occurs (i.e., 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.
attemptInterval	When a communications error occurs, the number of seconds to wait before trying the next set of (delayed) retries.
maxAttempts	Maximum number of delayed retry sets.
bankServURL	The URL used to access the BankServ payment gateway
socksHostName	TCP/IP Host address for SOCKS server.
socksPortNumber	TCP/IP port number for SOCKS server.

Cassette object XML example

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
  <CassetteCollection>
    <PSCassette active="1" cassette="BankServACH" changesPending="0" companyPkgName="ibm"
     enabled="1" traceSetting="-1" valid="1">
      <CassetteExtensionObject>
        <CassetteProperty propertyId="socksPortNumber" value="0">
        </CassetteProperty>
        <CassetteProperty propertyId="maxRetries" value="1">
        </CassetteProperty>
        <CassetteProperty propertyId="attemptInterval" value="300">
        </CassetteProperty>
        <CassetteProperty propertyId="readTimeout" value="60">
        </CassetteProperty>
        <CassetteProperty propertyId="connectRetries" value="1">
        </CassetteProperty>
        <CassetteProperty propertyId="maxAttempts" value="3">
        </CassetteProperty>
        <CassetteProperty propertyId="connectTimeout" value="60">
        </CassetteProperty>
        <CassetteProperty propertyId="socksHostName" value="">
        </CassetteProperty>
        <CassetteProperty propertyId="bankServURL"
         value="https://bazilla.bankserv.com/NASApp/hermes/FileSend">
        </CassetteProperty>
      </CassetteExtensionObject>
    </PSCassette>
  </CassetteCollection>
</PSApiResult>
```

AccountAdmin

Table 28. Cassette properties that belong to a PSMerchantAccount

Field name	Description
batchCloseTime	The time of day (in HMM format) in which a BatchClose should be automatically attempted.

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
  <MerchantAccountCollection>
    <PSMerchantAccount active="1" apApproveFlag="0" apDepositFlag="0"
        cassette="BankServACH"changesPending="0" enabled="1"
        financialInstName="Bank of Raleigh" merchantAccount="1"
        merchantAccountName="Emily" merchantNumber="123456789" valid="1">
      <CassetteExtensionObject>
        <CassetteProperty propertyId="batchCloseTime" value="2330">
        </CassetteProperty>
        <CassetteProperty propertyId="merchantNumber" value="123456789">
        </CassetteProperty>
        <CassetteProperty propertyId="accountNumber" value="1">
        </CassetteProperty>
      </CassetteExtensionObject>
    </PSMerchantAccount>
  </MerchantAccountCollection>
</PSApiResult>
```

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 BankServACH PaySystem data:

Table 29. Cassette properties that belong to PSMerchantCassetteSettings

Field Name	Description
merchantPIN	BankServ assigned Merchant PIN.

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
  <MerchantCassetteSettingsCollection>
    <PSMerchantCassetteSettings active="1" cassette="BankServACH" changesPending="0">
        enabled="1" merchantNumber="123456789" valid="1">
      <CassetteExtensionObject>
        <CassetteProperty propertyId="merchantNumber" value="123456789">
        </CassetteProperty>
        <CassetteProperty propertyId="merchantPIN" value="123">
        </CassetteProperty>
      </CassetteExtensionObject>
    </PSMerchantCassetteSettings>
  </MerchantCassetteSettingsCollection>
</PSApiResult>
```

Appendix A. Country codes and state codes

Table 30. Country codes

Country/region names	Country/region codes
AFGHANISTAN	AF
ALBANIA	AL
ALGERIA	DZ
AMERICAN SAMOA	AS
ANDORRA	AD
ANGOLA	AO
ANGUILLA	AI
ANTARCTICA	AQ
ANTIGUA AND BARBUDA	AG
ARGENTINA	AR
ARMENIA	AM
ARUBA	AW
AUSTRALIA	AU
AUSTRIA	AT
AZERBAIJAN	AZ
BAHAMAS	BS
BAHRAIN	BH
BANGLADESH	BD
BARBADOS	BB
BELARUS	BY
BELGIUM	BE
BELIZE	BZ
BENIN	BJ
BERMUDA	BM
BHUTAN	BT
BOLIVIA	BO
BOSNIA AND HERZEGOVINA	BA
BOTSWANA	BW
BOUVET ISLAND	BV
BRAZIL	BR
BRITISH INDIAN OCEAN TERRITORY	IO
BRUNEI DARUSSALAM	BN
BULGARIA	BG
BURKINA FASO	BF
BURUNDI	BI
CAMBODIA	KH

Table 30. Country codes (continued)

Country/region names	Country/region codes
CAMEROON	CM
CANADA	CA
CAPE VERDE	CV
CAYMAN ISLANDS	KY
CENTRAL AFRICAN REPUBLIC	CF
CHAD	TD
CHILE	CL
CHINA	CN
CHINA (HONG KONG S.A.R)	HK
CHINA (MACAO S.A.R)	MO
CHRISTMAS ISLAND	CX
COCOS (KEELING) ISLANDS	CC
COLOMBIA	CO
COMOROS	KM
CONGO	CG
CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD
COOK ISLANDS	CK
COSTA RICA	CR
CÔTE D'IVOIRE	CI
CROATIA	HR
CUBA	CU
CYPRUS	CY
CZECH REPUBLIC	CZ
DENMARK	DK
DJIBOUTI	DJ
DOMINICA	DM
DOMINICAN REPUBLIC	DO
EAST TIMOR	TP
ECUADOR	EC
EGYPT	EG
EL SALVADOR	SV
EQUATORIAL GUINEA	GQ
ERITREA	ER
ESTONIA	EE
ETHIOPIA	ET
FALKLAND ISLANDS (MALVINAS)	FK
FAROE ISLANDS	FO
FIJI	FJ
FINLAND	FI
FRANCE	FR

Table 30. Country codes (continued)

Country/region names	Country/region codes
FRENCH GUIANA	GF
FRENCH POLYNESIA	PF
FRENCH SOUTHERN TERRITORIES	TF
GABON	GA
GAMBIA	GM
GEORGIA	GE
GERMANY	DE
GHANA	GH
GIBRALTAR	GI
GREECE	GR
GREENLAND	GL
GRENADA	GD
GUADELOUPE	GP
GUAM	GU
GUATEMALA	GT
GUINEA	GN
GUINEA-BISSAU	GW
GUYANA	GY
HAITI	HT
HEARD ISLAND AND MCDONALD ISLANDS	HM
HOLY SEE (VATICAN CITY STATE)	VA
HONDURAS	HN
HUNGARY	HU
ICELAND	IS
INDIA	IN
INDONESIA	ID
IRAN, ISLAMIC REPUBLIC OF	IR
IRAQ	IQ
IRELAND	IE
ISRAEL	IL
ITALY	IT
JAMAICA	JM
JAPAN	JP
JORDAN	JO
KAZAKSTAN	KZ
KENYA	KE
KIRIBATI	KI
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP
KOREA, REPUBLIC OF	KR
KUWAIT	KW

Table 30. Country codes (continued)

Country/region names	Country/region codes
KYRGYZSTAN	KG
LAO PEOPLE'S DEMOCRATIC REPUBLIC	LA
LATVIA	LV
LEBANON	LB
LESOTHO	LS
LIBERIA	LR
LIBYAN ARAB JAMAHIRIYA	LY
LIECHTENSTEIN	LI
LITHUANIA	LT
LUXEMBOURG	LU
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	MK
MADAGASCAR	MG
MALAWI	MW
MALAYSIA	MY
MALDIVES	MV
MALI	ML
MALTA	MT
MARSHALL ISLANDS	MH
MARTINIQUE	MQ
MAURITANIA	MR
MAURITIUS	MU
MAYOTTE	YT
MEXICO	MX
MICRONESIA, FEDERATED STATES OF	FM
MOLDOVA, REPUBLIC OF	MD
MONACO	MC
MONGOLIA	MN
MONTSERRAT	MS
MOROCCO	MA
MOZAMBIQUE	MZ
MYANMAR	MM
NAMIBIA	NA
NAURU	NR
NEPAL	NP
NETHERLANDS	NL
NETHERLANDS ANTILLES	AN
NEW CALEDONIA	NC
NEW ZEALAND	NZ
NICARAGUA	NI
NIGER	NE

Table 30. Country codes (continued)

Country/region names	Country/region codes
NIGERIA	NG
NIUE	NU
NORFOLK ISLAND	NF
NORTHERN MARIANA ISLANDS	MP
NORWAY	NO
OMAN	OM
PAKISTAN	PK
PALAU	PW
PALESTINIAN TERRITORY, OCCUPIED	PS
PANAMA	PA
PAPUA NEW GUINEA	PG
PARAGUAY	PY
PERU	PE
PHILIPPINES	PH
PITCAIRN	PN
POLAND	PL
PORTUGAL	PT
PUERTO RICO	PR
QATAR	QA
RÉUNION	RE
ROMANIA	RO
RUSSIAN FEDERATION	RU
RWANDA	RW
SAINT HELENA	SH
SAINT KITTS AND NEVIS	KN
SAINT LUCIA	LC
SAINT PIERRE AND MIQUELON	PM
SAINT VINCENT AND THE GRENADINES	VC
SAMOA	WS
SAN MARINO	SM
SAO TOME AND PRINCIPE	ST
SAUDI ARABIA	SA
SENEGAL	SN
SEYCHELLES	SC
SIERRA LEONE	SL
SINGAPORE	SG
SLOVAKIA	SK
SLOVENIA	SI
SOLOMON ISLANDS	SB
SOMALIA	SO

Table 30. Country codes (continued)

Country/region names	Country/region codes
SOUTH AFRICA	ZA
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS
SPAIN	ES
SRI LANKA	LK
SUDAN	SD
SURINAME	SR
SVALBARD AND JAN MAYEN	SJ
SWAZILAND	SZ
SWEDEN	SE
SWITZERLAND	CH
SYRIAN ARAB REPUBLIC	SY
TAIWAN	TW
TAJIKISTAN	TJ
TANZANIA, UNITED REPUBLIC OF	TZ
THAILAND	TH
TOGO	TG
TOKELAU	TK
TONGA	TO
TRINIDAD AND TOBAGO	TT
TUNISIA	TN
TURKEY	TR
TURKMENISTAN	TM
TURKS AND CAICOS ISLANDS	TC
TUVALU	TV
UGANDA	UG
UKRAINE	UA
UNITED ARAB EMIRATES	AE
UNITED KINGDOM	GB
UNITED STATES	US
UNITED STATES MINOR OUTLYING ISLANDS	UM
URUGUAY	UY
UZBEKISTAN	UZ
VANUATU	VU
Vatican City State see HOLY SEE	
VENEZUELA	VE
VIET NAM	VN
VIRGIN ISLANDS, BRITISH	VG
VIRGIN ISLANDS, U.S.	VI
WALLIS AND FUTUNA	WF
WESTERN SAHARA	EH

Table 30. Country codes (continued)

Country/region names	Country/region codes
YEMEN	YE
YUGOSLAVIA	YU
Zaire see CONGO, THE DEMOCRATIC REPUBLIC OF THE	
ZAMBIA	ZM
ZIMBABWE	ZW

Table 31. State Codes

State Names	State codes
ALASKA	AK
ALABAMA	AL
ARKANSAS	AR
ARIZONA	AZ
CALIFORNIA	CA
COLORADO	CO
CONNECTICUT	CT
DELAWARE	DE
FLORIDA	FL
GEORGIA	GA
HAWAII	HI
IOWA	IA
IDAHO	ID
ILLINOIS	IL
INDIANA	IN
KANSAS	KS
KENTUCKY	KY
LOUISIANA	LA
MASSACHUSETTS	MA
MARYLAND	MD
MAINE	ME
MICHIGAN	MI
MINNESOTA	MN
MISSOURI	MO
MISSISSIPPI	MS
MONTANA	MT
NORTH CAROLINA	NC
NORTH DAKOTA	ND
NEBRASKA	NE
NEW HAMPSHIRE	NH
NEW JERSEY	NJ
NEW MEXICO	NM

Table 31. State Codes (continued)

State Names	State codes
NEVADA	NV
NEW YORK	NY
OHIO	OH
OKLAHOMA	OK
OREGON	OR
PENNSYLVANIA	PA
RHODE ISLAND	RI
SOUTH CAROLINA	SC
SOUTH DAKOTA	SD
TENNESSEE	TN
TEXAS	TX
UTAH	UT
VIRGINIA	VA
VERMONT	VT
WASHINGTON	WA
WISCONSIN	WI
WEST VIRGINIA	WV
WYOMING	WY

Appendix B. Cassette for BankServACH return codes

Almost all of the error conditions raised by the Cassette for BankServACH 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 BankServACH are defined by the framework. (See the *WebSphere Commerce Payments Programming Guide and Reference* for a list.) The following table lists BankServACH-specific errors and their definitions.

Secondary Return Code	Value	Description
SRC_CASSETTE_MERCHANTPIN	10001	Refers to $MERCHANTPIN
SRC_CASSETTE_ATTEMPTINTERVAL	10002	Refers to $ATTEMPTINTERVAL protocol data
SRC_CASSETTE_MAXATTEMPTS	10003	Refers to $MAXATTEMPTS protocol data
SRC_CASSETTE_BANKSERVURL	10004	Refers to $BANKSERVURL protocol data
SRC_CASSETTE_CONNECTTIMEOUT	10005	Refers to $CONNECTTIMEOUT protocol data
SRC_CASSETTE_SOCKSHOSTNAME	10006	Refers to $SOCKSHOSTNAME protocol data
SRC_CASSETTE_SOCKSPORTNUMBER	10007	Refers to $SOCKSPORTNUMBER protocol data
SRC_CASSETTE_READTIMEOUT	10008	Refers to $READTIMEOUT
SRC_CASSETTE_MAXRETRIES	10009	Refers to $MAXRETRIES protocol data
SRC_DEPOSIT_AMOUNT	10010	Refers to $DEPOSITAMOUNT
SRC_CASSETTE_CONNECTRETRIES	10011	Refers to $CONNECTRETRIES
SRC_METHOD_NOT_IMPLEMENTED	20000	The BankServ cassette does not support the framework method (e.g., CreateProtocolRequest).
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 BankServ accounts.
SRC_ACCOUNT_CREATE_SQL_FAILURE	21001	Database error occurred while attempting to create BankServ account.
SRC_ACCOUNT_UPDATE_SQL_FAILURE	21002	Database error occurred while attempting to update a BankServ account.

Secondary Return Code	Value	Description
SRC_ACCOUNT_DELETE_SQL_FAILURE	21003	Database error occurred while attempting to delete a BankServACH account.
SRC_ACCOUNT_MORE_THAN_ONE_BATCH	21004	There is more than one batch open for this account.
SRC_ACCOUNT_NULL_BATCH_NUMBER	21005	Cannot retrieve a batch since the batch number is NULL.
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 BankServ batch
SRC_BATCH_CREATE_SQL_FAILURE	22003	Database error occurred while attempting to create a BankServ batch
SRC_BATCH_UPDATE_SQL_FAILURE	22004	Database error occurred while attempting to update a BankServ batch
SRC_BATCH_DELETE_SQL_FAILURE	22005	Database error occurred while attempting to delete a batch
SRC_BATCH_NULL_ORDER_FOR_PAYMENT	22006	Framework order associated with the cassette's batch payment cannot be retrieved
SRC_BATCH_NULL_PAYMENT	22007	Batch payment cannot be retrieved.
SRC_BATCH_BAD_BATCH_IN_PAYMENT	22008	The cassette's batch number for payment and framework's batch number for payment are not the same.
SRC_BATCH_TIMER_WORK_ITEM_FAILURE	22009	A failure occurred while attempting to automatically close the batch.
SRC_BATCH_CLOSE_WORK_ITEM_FAILURE	22010	A failure occurred while attempting to schedule an automatic batch close.
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 an order
SRC_ORDER_CREATE_SQL_FAILURE	23002	Database error occurred while attempting to create an order
SRC_ORDER_UPDATE_SQL_FAILURE	23003	Database error occurred while attempting to update an order
SRC_ORDER_DELETE_SQL_FAILURE	23004	Database error occurred while attempting to delete an order
SRC_ORDER_SELECT_CLOSE_FAILURE	23005	Database error occurred while attempting to end a query
SRC_ORDER_UNABLE_TO_OBTAIN_CRYPTO_KEY	23006	Unable to obtain crypto key to encrypt sensitive data
SRC_PAYMENT_SELECT_PAYMENT_MISSING	24000	The payment cannot be resurrected since it is not in the database.

Secondary Return Code	Value	Description
SRC_PAYMENT_SELECT_SQL_FAILURE	24001	Database error occurred while attempting to retrieve a payment
SRC_PAYMENT_CREATE_SQL_FAILURE	24002	Database error occurred while attempting to create a payment
SRC_PAYMENT_UPDATE_SQL_FAILURE	24003	Database error occurred while attempting to update a payment
SRC_PAYMENT_DELETE_SQL_FAILURE	24004	Database error occurred while attempting to delete a payment
SRC_PAYMENT_SELECT_CLOSE_FAILURE	24005	Database error occurred while resurrecting the payment from the database.
SRC_PENDINGOP_CREATE_SQL_FAILURE	25000	Database error occurred while attempting to create a pending operation.
SRC_PENDINGOP_UPDATE_SQL_FAILURE	25001	Database error occurred while attempting to update a pending operation.
SRC_PENDINGOP_DELETE_SQL_FAILURE	25002	Database error occurred while attempting to delete a pending operation.
SRC_QUERY_ORD_SELECT_SQL_FAILURE	26000	A database error occurred while processing the result set for a query on cassette orders.
SRC_QUERY_PAY_SELECT_SQL_FAILURE	26001	A database error occurred while processing the result set for a query on cassette payments.
SRC_QUERY_BAT_SELECT_SQL_FAILURE	26002	A database error occurred while processing the result set for a query on cassette batches.
SRC_QUERY_ACC_SELECT_SQL_FAILURE	26003	A database error occurred while processing the result set for a query on cassette accounts.
SRC_QUERY_CASSETTE_SELECT_SQL_FAILURE	26004	A database error occurred while processing the result set for a query on cassette configuration.
SRC_QUERY_PAYSYS_SELECT_SQL_FAILURE	26005	A database error occurred while processing the result set for a query on cassette pay systems.
SRC_CASSETTE_ADMIN_SELECT_SQL_FAILURE	27000	Database error occurred while attempting to retrieve the CassetteAdmin data.
SRC_CASSETTE_ADMIN_UPDATE_SQL_FAILURE	27001	Database error occurred while attempting to update the CassetteAdmin data.
SRC_PAYSYS_NOT_VALID	28000	PaymentSystem is not in a valid state. It is missing required protocol data.
SRC_PAYSYS_CREATE_SQL_FAILURE	28001	An error occurred when adding the PaymentSystem to the database.
SRC_PAYSYS_UPDATE_SQL_FAILURE	28002	An error occurred when updating the PaymentSystem in the database.

Secondary Return Code	Value	Description
SRC_PAYSYS_DELETE_SQL_FAILURE	28003	An error occurred when deleting the PaymentSystem from the database.
SRC_NO_RESPONSE_TO_REQUEST_RETRIES_EXHAUSTED	30000	No authorization response received by BankServ host and internal recovery attempts have been exhausted.
SRC_CONNECT_FAILED_TO_HOST	30001	Unable to open a connection with the configured BankServ host.
SRC_HTTP_IO_ERROR	30002	Error trying to send data to, or receive data from, the BankServ host.
SRC_HTTP_RESPONSE_FAILURE	30003	An http response code other than 200 was received
SRC_UNSUPPORTED_CONTENT_TYPE	30004	The response contained an unsupported content type.
SRC_UNSUPPORTED_ENCODING	30005	The response contained an unsupported content encoding (e.g. other than us-ascii)
SRC_UNSUPPORTED_DOCUMENT_TYPE	30006	The BankServ host returned an unsupported document type
SRC_EMPTY_DOCUMENT	30007	The BankServ host returned an empty document
SRC_INVALID_DOCUMENT	30008	The BankServ host returned an invalid document (e.g. required field not present).
SRC_INVALID_DOCUMENT_CONTENT	30009	The BankServ host returned a valid document but the data was invalid (e.g. field to long)
SRC_MALFORMED_URL	30010	The URL used to connect to the BankServ host is not valid
SRC_MISSING_RESPONSE_BODY	30011	The data portion of the message from the BankServ host is missing.
SRC_TRUNCATED_RESPONSE_BODY	30012	The data portion of the message from the BankServ host is too short indicating data has been lost.
SRC_INTERNAL_FAILURE	30013	A request cannot be completed because a required resource is no longer available.
SRC_FORMAT_ERROR	30014	Either the message cannot be sent due to invalid data, or the received message cannot be processed due to invalid data.

Appendix C. Cassette for BankServACH messages

This chapter contains the Cassette for BankServACH-specific messages.

CEPBankServACH1001: The BankServACH 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.

CEPBankServACH1002: The BankServACH Cassette has started.

Severity: Information

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

User Response: None

CEPBankServACH1003: The BankServACH Cassette has shut down.

Severity: Information

Explanation: The BankServACH Cassette is no longer active.

User Response: None

CEPBankServACH2000: 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.

CEPBankServACH2001: Unable to create the account. Account number *acctNum* **for merchant number** *merchNum* **already exists.**

Severity: Error

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

User Response: Try to create the account again, but specify an account number that does not already exist.

CEPBankServACH2002: 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.

CEPBankServACH2003: 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.

CEPBankServACH2004: 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.

CEPBankServACH2005: 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.

CEPBankServACH2006: 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.

CEPBankServACH2007: 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.

CEPBankServACH2008: 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.

CEPBankServACH2009: 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.

CEPBankServACH2010: An error occurred while attempting to settle a batch for Merchant *merchNum*, **Account** *acctNum*, **Batch** *batchNum*. **The BankServ 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 BankServ host representative to determine why the batch already exists.

CEPBankServACH2011: Unable to create the account. Merchant number *merchNum* **already has an Account defined.**

Severity: Error

Explanation: The specified merchant already has an account. There can be only one account defined per merchant.

User Response: Use the account that is already defined for this merchant.

CEPBankServACH3000: 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 batch exists for the merchant.

CEPBankServACH3001: A 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.

CEPBankServACH3002: 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.

CEPBankServACH3003: 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.

CEPBankServACH3004: 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.

CEPBankServACH3005: 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.

CEPBankServACH3006: 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.

CEPBankServACH3007: 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.

CEPBankServACH3011: 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.

CEPBankServACH3013: 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.

CEPBankServACH3015: 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.

CEPBankServACH3016: 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.

CEPBankServACH3017: 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.

CEPBankServACH4000: 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.

CEPBankServACH4001: 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.

CEPBankServACH4002: An SQL exception was caught while closing a SELECT statement.

Severity: Error

Explanation: An SQL exception occurred while attempting to close an 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.

CEPBankServACH4003: 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.

CEPBankServACH4004: 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.

CEPBankServACH4005: 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.

CEPBankServACH4006: 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.

CEPBankServACH5000: 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.

CEPBankServACH5001: 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.

CEPBankServACH5002: 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 an 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.

CEPBankServACH5003: 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.

CEPBankServACH5004: 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.

CEPBankServACH5005: 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.

CEPBankServACH5006: 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.

CEPBankServACH7000: An SQL exception was caught while processing the result set for a query on cassette orders.

Severity: Error

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

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

CEPBankServACH7001: An SQL exception was caught while processing the result set for a query on cassette payments.

Severity: Error

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

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

CEPBankServACH7002: An SQL exception was caught while processing the result set for a query on cassette credits.

Severity: Error

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

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

CEPBankServACH7003: An SQL exception was caught while processing the result set for a query on cassette batches.

Severity: Error

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

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

CEPBankServACH7004: An SQL exception was caught while processing the result set for a query on cassette accounts.

Severity: Error

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

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

CEPBankServACH7005: An SQL exception was caught while processing the result set for a query on cassette payment systems.

Severity: Error

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

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

CEPBankServACH7006: An SQL exception was caught while processing the result set for a query on cassette configuration.

Severity: Error

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

User Response: Check the connection to the database

to make sure that there is not a problem with the communication between the WebSphere Commerce Payments machine and the database server.

CEPBankServACH8000: 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.

CEPBankServACH8001: 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.

CEPBankServACH9000: 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.

CEPBankServACH9001: 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.

CEPBankServACH9002: 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.

CEPBankServACH9003: 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.

CEPBankServACH9005: An error occurred while attempting to *encryptordecrypt* **the data.**

Severity: Error

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

User Response: Contact your IBM support representative.

CEPBankServACH0606: An internal error occurred: *exception text***.**

Severity: Error

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

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

CEPBankServACH9006: The value *value* **specifed for** *parameter* **is not valid.**

Severity: Error

Explanation: An incorrect value was entered in the referenced protocol data.

User Response: Reissue the command with a valid value.

Appendix D. Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department TL3B/Building 503
PO Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195

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

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

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

Trademarks

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

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

Microsoft, Windows NT, and the Windows logo are 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 and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

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

Glossary

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

A

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

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

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

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

approve. Within IBM e-commerce, a WebSphere Commerce Payments verb. A merchant issues this verb to create a Payment object. For cassettes that implement credit card protocols, this verb will likely map to authorization (see authorize). Other cassettes may implement the approval process differently.

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

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

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

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

B

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

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

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

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

with these guidelines. (2) Within IBM e-commerce, one of the fundamental WebSphere Commerce Payments objects is the Batch. A Batch is an object with which Payment and Credit objects are associated. Transfer of funds is to occur when the batch is closed. (3) A group of records or data processing jobs brought together for processing or transmission.

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

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

C

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

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

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

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

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

- signature
- key encipherment
- certificate signature
- CRL signature

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

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

client. (1) A functional unit that receives shared services from a server. For example, a personal computer requesting HTML documents from a Web server is a client of that server. (2) A computer system or process that requests a service of another computer system or process that is typically referred to as a server. Multiple clients may share access to a common server.

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

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

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

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

D

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

document type definition (DTD). The rules that specify the structure for a particular class of SGML or XML documents. The DTD defines the structure with

elements, attributes, and notations, and it establishes constraints for how each element, attribute, and notation may be used within the particular class of documents. A DTD is analogous to a database schema in that the DTD completely describes the structure for a particular markup language.

DTD. See document type definition.

E

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

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

encryption. (1) In computer security, the process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process. (2) The conversion of data into a form that cannot be easily understood so as to prevent unauthorized access, especially during transmission.

event. (1) A representation of a change that occurs to a part. The change enables other interested parts to receive notification when something about the part changes. For example, a push button generates an event by signalling that it has been clicked, which may cause another part to display a window. (2) Any significant change in the state of a system resource, network resource, or network application. An event can be generated for a problem, for the resolution of a problem, or for the successful completion of a task.

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

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

F

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

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

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

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

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

G

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

H

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

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

- `mycomputer.city.company.com`
- `mycomputer`

HTML. See Hypertext Markup Language.

HTTP. See Hypertext Transfer Protocol.

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

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

I

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

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

internet. (1) In TCP/IP, a collection of interconnected networks that functions as a single, large network. (2) A collection of interconnected networks that use the Internet suite of protocols. The internet that allows universal access is referred to as the Internet (with a capital "I"). An internet that provides restricted access (for example, to a particular enterprise or organization) is frequently called an intranet, whether or not it also connects to the public Internet.

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

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

J

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

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

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

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

K

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

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

L

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

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

M

merchant. A seller of goods, services, and/or other information who accepts payment for these items electronically. The merchant may also provide electronic selling services and/or electronic delivery of items for sale. The merchant supervises the overall store objectives and management, in addition to tracking the store sales.

merchant bank. An Acquiring Financial institution. A merchant bank acquires merchant business by supplying the merchant with the means to accept credit cards for payment. The financial institution charges the merchant a fee for providing these services.

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

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

N

number of credits. A credit is a transaction sent when the merchant needs to return money to the cardholder (via the Acquirer and the Issuer) following a valid capture message, such as when goods have been returned or were defective. Credits can be for up to the

total amount of all payments associated with an Order. There can be zero or more Credits per Order.

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

O

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

order. In WebSphere Commerce Payments, an order represents all the instructions and information needed from the consumer (payer) in order for the merchant (payee) to collect money.

order amount. The amount of the order.

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

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

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

P

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

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

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

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

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

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

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

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

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

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

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

public key. (1) 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

P

Q

R

S

T

U

W

Printed in U.S.A.