

IBM WebSphere Commerce Payments for
Multiplatforms



Cassette for BankServACH Supplement

Version 3.1

IBM WebSphere Commerce Payments for
Multiplatforms



Cassette for BankServACH Supplement

Version 3.1

Note

Before using this information and the product it supports, be sure to read the general information under Appendix D, "Notices" on page 53.

Third Edition (July 2002)

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

©Copyright IBM Corporation 2001, 2002. All rights reserved.

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

Contents

Preface	v
Conventions in this book	v
Terminology	v
Additional information	vi
Chapter 1. Overview of BankServACH	1
Merchant registration process	2
Merchant PIN	2
Communications	3
Chapter 2. BankServ and WebSphere Commerce Payments Concepts	5
A BankServ purchase example	5
WebSphere Commerce Payments object model implementation	7
Administration objects	7
Financial objects	8
Cassette-specific characteristics and behaviors	8
Retry considerations	8
Security Considerations	9
Performance Considerations	10
Cassette for BankServACH payment command summary	10
Summary of state changes	11
Protecting sensitive data	12
Chapter 3. Installing the Cassette for BankServACH	13
Before installing Cassette for BankServACH	13
Installing Cassette for BankServACH (Windows and UNIX platforms)	13
WebSphere Commerce Payments directory structure	14
Installing Cassette for BankServACH (iSeries)	14
Installing the cassette	14
Adding the Cassette for BankServACH to a WebSphere Commerce Payments Instance	15
Uninstalling Cassette for BankServACH on Windows NT	15
Uninstalling Cassette for BankServACH on Windows 2000	16
Uninstalling Cassette for BankServACH on Solaris	16
Uninstalling Cassette for BankServACH on Linux	16
Uninstalling Cassette for BankServACH on AIX	17
Removing Cassette for BankServACH from a WebSphere Commerce Payments Instance for iSeries	17
Uninstalling Cassette for BankServACH for iSeries	17
Chapter 4. Cassette for BankServACH Cashier Profiles	19
Chapter 5. Using WebSphere Commerce Payments commands	21
Cassette for BankServACH Commands	21
AcceptPayment	22
Approve	22
ApproveReversal	22
BatchOpen	22
BatchPurge	22
BatchClose	22
CassetteControl	23
CloseOrder	23
CreateAccount	23

CreatePaySystem	23
DeleteBatch	23
Deposit	23
DepositReversal	23
ModifyAccount	24
ModifyCassette	24
ModifyPaySystem	24
ReceivePayment	24
Refund	24
RefundReversal	25
Chapter 6. Using BankServACH Objects	27
Financial objects used by Cassette for BankServACH	27
BankServACH Order	27
BankServACH Payment	28
BankServACH Batch	28
Administration objects used by Cassette for BankServACH	29
CassetteAdmin	30
AccountAdmin	30
PaySystemAdmin	31
Appendix A. Country Codes and State Codes	33
Appendix B. BankServACH Return Codes	41
Appendix C. Cassette for BankServACH Messages	45
Appendix D. Notices	53
Trademarks	54
Glossary	55
Index	69

Preface

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

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

Conventions in this book

Table 1. Conventions in this book

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

Terminology

Enabling electronic shopping requires different participants and software components:

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

financial network

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

gateway

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

merchant

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

merchant bank

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

merchant server

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

payment server

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

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

ACH Operator

The 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

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

- The *IBM WebSphere Commerce Payments for Multiplatforms Installation Guide, Version 3.1* provides details on installing WebSphere Commerce Payments.

- The *IBM WebSphere Commerce Payments for Multiplatforms Administrator's Guide, Version 3.1* provides details on configuring WebSphere Commerce Payments.
- The *IBM WebSphere Commerce Payments for Multiplatforms Programmer's Guide and Reference, Version 3.1* describes the programming commands that provide the interface to WebSphere Commerce Payments.

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

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

- **<http://www.ibm.com/payment/>** provides more information on the IBM WebSphere Commerce payment processing products.
- **<http://www.software.ibm.com/commerce/payment/support/serv/index.html>** provides current WebSphere Commerce Payments technical information and links to the latest softcopy views of all WebSphere Commerce Payments documentation.
- **<http://www.ibm.com/software/websphere/appserv/library.html>** provides documentation links for IBM WebSphere Application Server.
- **<http://www.ibm.com/software/data/db2/udb>** provides documentation links for IBM Universal Database.
- **<http://www.ibm.com/software/commerce/connect/>** points to the financial institutions that have established payment gateways to handle WebSphere Commerce Payments payment processing needs.

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-presentation 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-presentation 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-presentation, 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.

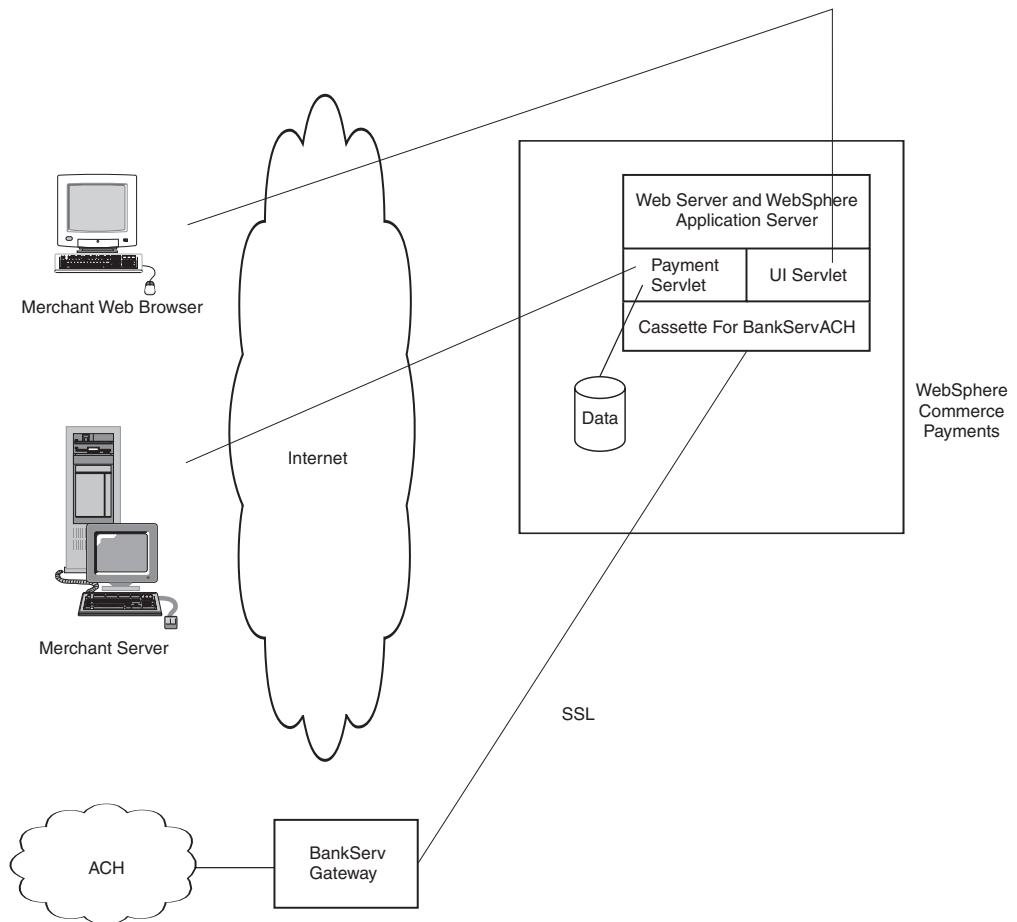


Figure 1. High level overview of BankServ processing

Merchant registration process

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 BankServACH Cassette supports the transaction:

Table 2. 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 merchantPIN on a specified date	No
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 BankServACH cassette. 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 3. BankServ Rule Sets

Rule set	Meaning	Supported by BankServ 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

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.

The following diagram shows an overview of the processing.

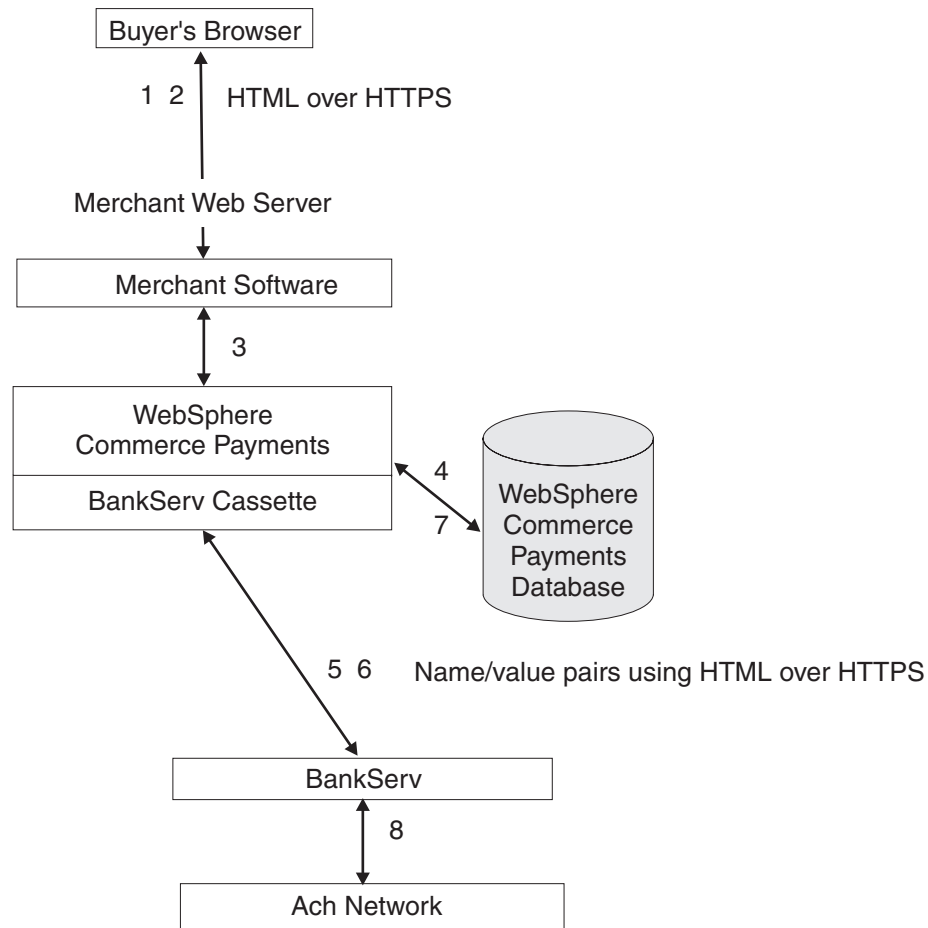


Figure 2. Processing overview

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 BankServ cassette update the WebSphere Commerce Payments database with the information needed to perform the transaction.
5. The BankServACH cassette 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 BankServACH cassette 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 Administrator's Guide* for a complete 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 6, "Using BankServACH Objects" on page 27.

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.

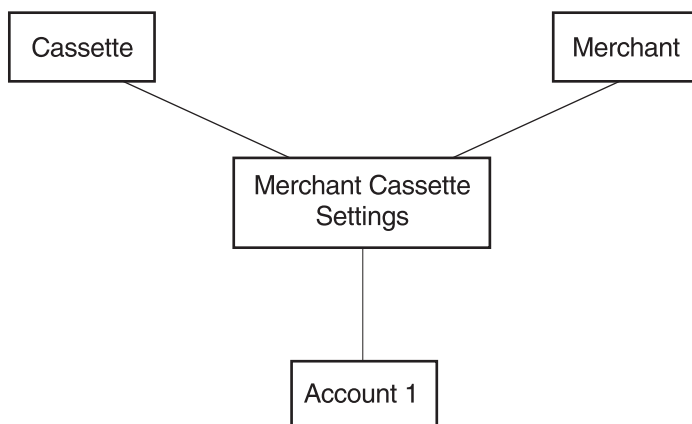


Figure 3. BankServACH PaySystem

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 6, “Using BankServACH Objects” on page 27. For descriptions of the payment objects and for programming information, see the *WebSphere Commerce Payments Programmer’s Guide and Reference*.

Cassette-specific characteristics and behaviors

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

Retry considerations

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 24.

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 affect.

- 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 follows:

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 iSeries™ 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/commerce/payment/support/index.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 (3 by default), resources may not be available to handle a communications he 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 by the eTill.spoolsize argument on the command line used to start WebSphere Commerce Payments (e.g., DeTill.spoolsize=6).

Cassette for BankServACH payment command summary

Table 4 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 4. Payment Command Summary

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

Table 4. Payment Command Summary (continued)

API Command	BankServ message
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 5. 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
ApproveReversal (partial reversal)	PAYMENT_APPROVED
Cancel Order	ORDER_CANCELED

Table 5. Summary of State Changes (continued)

API Command	Object State
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

Protecting sensitive data

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

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

Table 6. . 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.

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

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

Chapter 3. Installing the Cassette for BankServACH

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

Before installing Cassette for BankServACH

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

Note: A prior version of the cassette cannot be installed on top of the WebSphere Commerce Payments Version 3.1.3 Framework. If you currently use a prior version of the Cassette for BankServACH, you must install the PTF for the WebSphere Commerce Payments Version 3.1.3 Cassette for BankServACH software for your cassette data to be migrated and compatible with the WebSphere Commerce Payments Version 3.1.3 Framework.

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

Notes:

1. You must use the WebSphere Application Server administrative console to stop the WebSphere Commerce Payments Application Server before installing the Cassette for BankServACH on Windows and UNIX platforms.
2. iSeries systems do not require that WebSphere Commerce Payments or WebSphere Application Server be ended during installation.

Installing Cassette for BankServACH (Windows and UNIX platforms)

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

- On AIX, Linux and Solaris, you must logon as **root** directory.
- On Windows NT and Windows 2000, you must logon as a user who is a member of the administrator group.

1. Once logged in, insert the CD-ROM containing the WebSphere Commerce Payments Cassette for BankServACH.
 - For Windows NT, go to the *nt* directory.
 - For Windows 2000, go to the *nt* directory.
 - For Solaris, go to the *solaris* directory.
 - For Linux, go to the *linux* directory.
 - For AIX, go to the *aix* directory.
2. Enter **InstallBankServACHCassette** to start the installation. Enter the information requested on the installation screens.
3. The *IBM Cassette for BankServACH Readme* screen indicates that the configuration of the Cassette for BankServACH has successfully completed and lets you display the README, if desired.

WebSphere Commerce Payments directory structure

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

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

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

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

Installing Cassette for BankServACH (iSeries)

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

Installing the cassette

Use the Restore License Program (**RSTLICPGM**) CL command to install the Cassette for BankServACH option of the IBM WebSphere Commerce Payments for iSeries product.

Specify the WebSphere Commerce Payments product number, option 4 for the Cassette for BankServACH and the device from which the product is to be installed. For example: RSTLICPGM LICPGM(5733PY3) DEV(OPT01) OPTION(4).

Adding the Cassette for BankServACH to a WebSphere Commerce Payments Instance

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

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

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

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

Note: Alternately, you can use the **(ADDPYMCSS)** CL command. When executing the ADDPYMCSS command on an iSeries system, use the short name **BNKSRVACH**.

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

Uninstalling Cassette for BankServACH on Windows NT

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

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

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

Uninstalling Cassette for BankServACH on Windows 2000

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

1. Use the WebSphere Application Server Administrative console to stop the WebSphere Commerce Payments Application Server.
2. Go to the Windows 2000 Control Panel.
3. Click the **Add/Remove Programs** icon.
4. Select the **WebSphere Commerce Payments Cassette for BankServACH**.
5. Click **Add/Remove**.

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

Uninstalling Cassette for BankServACH on Solaris

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

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

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

```
./UninstallBankServACH.sh
```

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

Uninstalling Cassette for BankServACH on Linux

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

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

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

```
./UninstallBankServACH.sh
```

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

Uninstalling Cassette for BankServACH on AIX

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

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

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

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

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

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

To remove the Cassette for BankServACH from a WebSphere Commerce Payments instance:

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

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

Uninstalling Cassette for BankServACH for iSeries

To uninstall the Cassette for BankServACH option from the system, use the Delete License Program (**DLTLICPGM**) CL command to delete option 4 from WebSphere Commerce Payments. For example:

```
DLTLICPGM LICPGM(5733PY3) OPTION(4)
```

.

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

Chapter 4. Cassette for BankServACH Cashier Profiles

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

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

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

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

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

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

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

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

The Cassette for BankServACH Cashier profile is in the directory:

`<<<Payments_installdir>>/profiles/`. For details on designing and tailoring profiles, please see Chapter 3 of the *WebSphere Commerce Payments Programmer's Guide and Reference*, Version 3.1. The Cassette for BankServACH Cashier profile is **SampleCheckoutBankServACH.profile**.

Chapter 5. Using WebSphere Commerce Payments commands

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

- 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 Programmer's Guide and Reference* for a complete listing 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 Programmer's 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.

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 33 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 33 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.
\$CHECKINGACCOUNTNUMBER	Specifies the checking account number of the buyer. The value specified is a 1–17 digit number.

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" on page 23 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

Optional keywords for CreateAccount command

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

CreatePaySystem

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

Optional keywords for ModifyAccount command

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

ModifyCassette

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	Socks host name
\$SOCKSPORTNUMBER	Socks port number
\$BANKSERVURL	The URL to access the BankServ payment gateway.

ModifyPaySystem

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 6. Using BankServACH Objects

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

Cassette properties that belong to a POrder 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="0:123456789:1" 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>
      </CassetteExtensionObject>
    </PSOrder>
  </OrderCollection>
</PSApiResult>
```

```

    <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

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

Cassette properties that belong to a PSBatch Object

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

Cassette properties that belong to a PSBatch Object

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>
```

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

CassetteAdmin

Cassette properties that belong to a PSCassette Object

Field	Description
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

Cassette properties that belong to a PSMerchantAccount

Field Name	Description
------------	-------------

Cassette properties that belong to a PSMerchantAccount

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:

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 7. 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
CAMEROON	CM

Table 7. Country Codes (continued)

Country/Region Names	Country/Region Codes
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 7. 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

Table 7. Country Codes (continued)

Country/Region Names	Country/Region Codes
KUWAIT	KW
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

Table 7. Country Codes (continued)

Country/Region Names	Country/Region Codes
NICARAGUA	NI
NIGER	NE
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

Table 7. Country Codes (continued)

Country/Region Names	Country/Region Codes
SLOVENIA	SI
SOLOMON ISLANDS	SB
SOMALIA	SO
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

Table 7. Country Codes (continued)

Country/Region Names	Country/Region Codes
VIRGIN ISLANDS, BRITISH	VG
VIRGIN ISLANDS, U.S.	VI
WALLIS AND FUTUNA	WF
WESTERN SAHARA	EH
YEMEN	YE
YUGOSLAVIA	YU
Zaire see CONGO, THE DEMOCRATIC REPUBLIC OF THE	
ZAMBIA	ZM
ZIMBABWE	ZW

Table 8. 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

Table 8. State Codes (continued)

State Names	State Codes
NORTH DAKOTA	ND
NEBRASKA	NE
NEW HAMPSHIRE	NH
NEW JERSEY	NJ
NEW MEXICO	NM
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. 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 *IBM WebSphere Commerce Payments Programmer's Guide*, 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 *IBM WebSphere Commerce Payments Programmer's Guide*, for a list.) The following table lists BankServACH-specific errors and their definitions.

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.
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.
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 CasetteAdmin data.
SRC_CASSETTE_ADMIN_UPDATE_SQL_FAILURE	27001	Database error occurred while attempting to update the CasetteAdmin 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.
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 too 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.

CEPBankServACH1000 This method has not been implemented yet. Method name = *methodName*.

Severity: Information

Explanation:

User Response:

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 service 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 shutdown.

Severity: Information

Explanation: The BankServACH Cassette is no longer active.

User Response: None

CEPBankServACH2000 A SQL exception was caught while selecting existing accounts from the database.

Severity: Error

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

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

machine and the database server.

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: Re-attempt the CreateAccount request, specifying 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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

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 Payment Server database. This could be due to an error connecting to the database.

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

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 Payment Server database. This could be due to an error connecting to the database.

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

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 Payment Server database.

User Response: Ensure that the specified batch exists for the specified 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 Payment Server database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

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

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

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 Payment Server database. This could be due to an error connecting to the database.

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

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 service 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 service 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 service 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 service 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 service 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 service representative.

CEPBankServACH3016 An exception was caught while attempting to construct 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 service representative.

CEPBankServACH3017 An exception was caught while attempting to construct 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 service 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 Payment Server database.

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

CEPBankServACH4001 A SQL exception was caught while selecting existing orders from the database.

Severity: Error

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

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

CEPBankServACH4002 A SQL exception was caught while closing a SELECT statement.

Severity: Error

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

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

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

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 Payment Server database. This could be due to an error connecting to the database.

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

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 service 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 Payment Server database.

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

CEPBankServACH5001 A SQL exception was caught while selecting existing payments from the database for Merchant *merchNum*, Order *orderNum* and Payment *paymentNum*.

Severity: Error

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

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

CEPBankServACH5002 A SQL exception was caught while closing a SELECT statement for Merchant *merchNum*, Order *orderNum* and Payment *paymentNum*.

Severity: Error

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

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

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

machine and the database server.

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

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 Payment Server database. This could be due to an error connecting to the database.

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

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 service representative.

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

Severity: Error

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

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

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

Severity: Error

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

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

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

Severity: Error

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

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

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

Severity: Error

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

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

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

Severity: Error

Explanation: An SQL exception occurred processing a result set that was obtained by querying a database

view. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

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

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

Severity: Error

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

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

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

Severity: Error

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

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

CEPBankServACH8000 A SQL exception was caught while selecting existing cassette configuration from the database.

Severity: Error

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

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

CEPBankServACH8001 An SQL exception occurred while updating Cassette Configuration in the database.

Severity: Error

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

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

CEPBankServACH9000 A SQL exception was caught while selecting existing payment systems from the database.

Severity: Error

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

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

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

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

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 Payment Server database. This could be due to an error connecting to or writing to the database, or due to an error in the content of the data that is being written to the database.

User Response: Check the connection to the

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

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 Payment Server database. This could be due to an error connecting to the database.

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

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 service representative.

CEPBankServACH0606 An internal error occurred: *exception text*.

Severity: Error

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

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

CEPBankServACH9006 The value *value* specified 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
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

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

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

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

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

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

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

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

IBM Corporation

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

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

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

Trademarks

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

- AS/400
- DB2
- IBM
- WebSphere

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

Microsoft, Windows NT, and the Windows logo are trademarks 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.

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

Glossary

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

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

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

The following cross-references are used in this dictionary:

Contrast with:

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

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

Obsolete term for:

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

A

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

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

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

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

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

ADSM. See ADSTAR Distributed Storage Manager.

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

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

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

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

approve all. Selects all orders displayed for approval.

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

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

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

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

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

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

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

B

balance. Within IBM e-commerce, an attribute of a WebSphere Commerce Payments Batch object. Indicates whether the merchant and financial institution agreed on the contents of the batch when it was closed. See 66 for more details.

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

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

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

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

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

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

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

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

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

BCD. See binary-coded decimal notation.

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

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

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

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

browser. See Web browser.

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

C

CA. See certificate authority.

capture. (1) In SETCo., a transaction sent after the merchant has shipped the goods. This transaction will trigger the movement of funds from the Issuer to the Acquirer and then to the merchant account. (2) In the credit card world, payment is actually made when the funds are *captured*. All payments must be authorized *and* captured (although this work can be done at the same time). Note that all payments are associated with a batch (see “void payment” on page 67) and that the actual capture occurs when the associated batch is closed.

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

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

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

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

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

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

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

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

- signature
- key encipherment
- certificate signature
- CRL signature

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

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

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

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

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

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

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

certification authority. See certificate authority.

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

CGI. See Common Gateway Interface.

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

CGI script. See CGI program.

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

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

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

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

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

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

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

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

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

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

constructor method. See constructor.

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

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

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

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

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

CSP. See commerce service provider.

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

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

D

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

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

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

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

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

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

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

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

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

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

DTD. See document type definition.

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

E

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

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

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

ECML. See Electronic Commerce Modeling Language.

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

electronic commerce. See e-commerce.

Electronic Commerce Modeling Language (ECML).

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

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

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

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

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

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

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

F

failover. See fallover.

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

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

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

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

force. Within IBM e-commerce, a WebSphere Commerce Payments verb. An attempt to settle a batch (see 66). 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).

FQDN. See fully qualified domain name.

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

H

HACMP. See high-availability cluster multiprocessing.

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

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

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

hash. See root key hash.

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

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

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

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

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

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

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

- mycomputer.city.company.com
- mycomputer

HTML. See Hypertext Markup Language.

HTTP. See Hypertext Transfer Protocol.

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

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

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

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

identify. To establish the identity.

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

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

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

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

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

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

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

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

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

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

IPC. See interprocess communication.

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

J

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

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

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

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

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

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

JDBC. See Java Database Connectivity.

JDK. See Java Development Kit.

JRE. See Java Runtime Environment.

JVM. See Java Virtual Machine.

K

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

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

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

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

L

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

M

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

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

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

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

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

MIME. See Multipurpose Internet Mail Extensions.

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

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

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

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

N

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

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

node. See cluster node.

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

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

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

O

ODBC. See Open Database Connectivity.

ODBC bridge. See Java Database Connectivity.

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

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

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

order amount. The amount of the order.

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

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

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

P

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

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

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

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

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

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

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

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

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

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

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

PGTM. See Payment Gateway Transaction Manager.

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

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

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

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

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

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

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

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

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

R

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

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

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

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

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

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

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

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

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

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

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

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

S

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

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

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

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

Secure Electronic Transaction. See SET Secure Electronic Transaction.

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

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

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

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

SET. See SET Secure Electronic Transaction.

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

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

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

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

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

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

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

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

SMIT. See System Management Interface Tool.

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

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

socksified. See socks-enabled.

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

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

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

SSL. See Secure Sockets Layer.

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

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

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

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

T

TEC. See Tivoli Enterprise Console.

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

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

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

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

time opened. The time that the batch was created.

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

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

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

Tivoli GEM. See Tivoli Global Enterprise Manager.

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

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

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

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

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

trust chain. In SETCo., a synonym for certificate chain. See 58.

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

U

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

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

URL. See uniform resource locator.

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

V

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

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

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

W

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

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

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

Web. See World Wide Web.

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

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

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

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

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

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

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

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

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

WWW. See World Wide Web.

X

XML. See Extensible Markup Language.

68 Cassette for BankServACH Supplement

Numerics

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

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

Index

A

AcceptPayment 21
Account 30
AccountAdmin object 7
ACH Operator
 definition vi
administration objects 7
 account 7
 cassette 7
 PaySystemAdmin 7
Approve 22
ApproveReversal 22

B

BankServ Order 27
BankServ Payment 28
Batch 28
 definition vi
Batch Object
 XML example 29
Batch objects 8
batch, delete 23
BatchOpen 22
BatchPurge 22

C

cardholder
 definition v
Cassette for BankServACH
 installation 13
 uninstalling on Windows NT 15
Cassette properties
 accountadmin 30
 BankServ payment 28
 batch 28
 order 27
 paysystemadmin 31
CassetteAdmin 30
CassetteAdmin object 7
CassetteControl 23
closeorder
 delete option 23
communication parameters 8
CreateAccount 23

D

delete batch 23
delete option, closeorder 23
depositreversal 23
document, PSApiResult 27

F

financial network
 definition v
Framework object
 extensions 29

G

gateway
 definition v

I

IBM Universal Database
 web site vii
IBM WebSphere Application Server
 web site vii

K

keywords, Optional
 acceptpayment 22
 modifyaccount 24
 modifycassette 24
 modifypaysystem 24
keywords, Required
 acceptpayment 21
 createaccount 23

M

MaxRetries 30
merchant
 definition vi
merchant server
 definition vi
model, object 27
 implementation 7
models, object 5
ModifyAccount 24
ModifyCassette 24

N

Notices 53

O

object model 27
 implementation 7
object models 5
object, account 7
Object, Batch
 XML example 29
object, cassette 7

- object, Framework
 - extensions 29
- object, Order
 - XML example 27
- Object, Payment
 - XML example 28
- objects, administration 7
 - account 7
 - cassette 7
 - PaySystemAdmin 7
- objects, Batch 8
- objects, Order 8
- objects, payment
 - batch 8
 - credit 8
 - order 8
 - payment 8
- objects, Payment 8
- ODFI
 - definition vi
- Optional keywords
 - acceptpayment 22
 - modifyaccount 24
 - modifycassette 24
 - modifypaysystem 24
- Order object
 - XML example 27
- Order objects 8
- Originator
 - definition vi

P

- parameters, communication 8
- parameters, protocol 21
- Payment Object
 - XML example 28
- payment objects
 - batch 8
 - credit 8
 - order 8
 - payment 8
- Payment objects 8
- payment processing
 - implementation 5
- payment processing, WebSphere Commerce Payments
 - web site vii
- payment server
 - definition vi
- properties, Cassette
 - accountadmin 30
 - BankServ payment 28
 - batch 28
 - order 27
 - paysystemadmin 31
- protocol parameters 21
- PSApiResult document 27
- Purchase example 5

Q

- query call 27

R

- RDFI
 - definition vi
- ReadTimeout 30
- ReceivePayment 24
- Receiver
 - definition vi
- Refund 24
- RefundReversal 25
- Required keywords
 - acceptpayment 21
 - createaccount 23
- Retry 30
- reversal, deposit 23

T

- technical information, WebSphere Commerce Payments
 - web site vii
- trademarks 54

U

- Universal Database, IBM
 - web site vii

W

- Web sites
 - IBM Payment Suite vii
- WebSphere Application Server, IBM
 - web site vii
- WebSphere Commerce Payments payment processing
 - web site vii
- WebSphere Commerce Payments technical information
 - web site vii
- WebSphere Commerce Payments Web site 13

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

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

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

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

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

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

Please tell us how we can improve this book:

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

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

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

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



Fold and Tape

Please do not staple

Fold and Tape



Printed in U.S.A.