IBM WebSphere Commerce Payments for Multiplatforms

Cassette for SET Supplement

Version 3.1

IBM WebSphere Commerce Payments for Multiplatforms

Cassette for SET Supplement

Version 3.1

Note

Before using this information and the product it supports, be sure to read the general information under Appendix H, “Notices” on page 159.

Fourth Edition (July 2002)

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

Contents

Preface vii
Conventions in this book vii
Terminology vii
Additional information ix
SET Secure Electronic Transaction LLC Documents x

Chapter 1. Introducing the SET Secure Electronic Transaction protocol . . . 1
Understanding the SET protocol 1
Certificate message flows 2
Cardholder – merchant message flows 2
Merchant – financial institution message flows 3
How do SET messages relate to the Application Programming Interface(API) and objects? 4
Merchant and certificate authority interaction 4
Merchant and cardholder interaction 5
Merchant and financial institution interaction 6

Chapter 2. Understanding SET protocol and WebSphere Commerce Payments concepts 9
Administration objects 9
Financial objects 11
SET profiles and acquirer behavior 11
Split payments 12
Batch processing 13
Batch reconciliation 15

Chapter 3. Understanding concepts unique to Cassette for SET 17
Order creation 17
Merchant initiated authorization 17
SaleDetail and AVSData 18
Purchasing cards 18
Levels of purchasing card data supported 19
Purchasing card data processing 19
Relationship to SaleDetail 20
Sale transactions 20
Understanding 2KP and 3KP 21
Extracting the SET initiation message 21
Using the IBM WebSphere Commerce Payments SET JPO Extension 22
Configuring the host address in wakeup message 22
Event notification for SET 22
Protecting sensitive data 23

Chapter 4. Installing the Cassette for SET 25
Before installing the Cassette for SET 25
Cassette for SET migration tips 25
Migration support 25
Time required to migrate 25
PAN and Expiry fields 25
Orders migration 26
Installing Cassette for SET (Windows and UNIX® platforms) 26
WebSphere Commerce Payments directory structure 27
Installing the Cassette for SET (iSeries) 27
Installing the cassette 27

Adding a Cassette for SET to a WebSphere Commerce Payments Instance 28
Configuring the cassette for socks 28
Uninstalling Cassette for SET 29
Uninstalling Cassette for SET on Windows NT or Windows 2000 29
Uninstalling Cassette for SET on Solaris 30
Uninstalling Cassette for SET on AIX 30
Removing Cassette for SET from a WebSphere Commerce Payments Instance for iSeries 31
Uninstalling Cassette for SET from iSeries 31

Chapter 5. Getting started 33
Cassette for SET tutorial 33
Before starting this tutorial 33
Starting the WebSphere Commerce Payments user interface 34
Selecting a WebSphere Commerce Payments merchant and authorizing a cassette 34
Logging in as the Merchant Administrator 35
Creating an Account 35
Managing payment processing 37
Creating orders using the Sample Checkout 38
Approving orders 38
Depositing payments 39
Settling batches 40
Issuing a credit 42
Viewing batch totals 43
Other administration tasks 43

Chapter 6. Cassette for SET Cashier profiles 45

Chapter 7. Using WebSphere Commerce Payments commands with Cassette for SET 47
AcceptPayment 47
Approve 50
ApproveReversal 50
BatchOpen 51
CancelOrder 51
CloseOrder 51
CreateAccount 51
CreateMerchantCassetteObject 54
DeleteBatch 55
DeleteMerchantCassetteObject 56
Deposit 56
ModifyAccount 57
ModifyCassette 58
ModifyMerchantCassetteObject 59
ReceivePayment 62
Refund 65

Chapter 8. Using WebSphere Commerce Payments objects 67
Financial objects used by Cassette for SET 67
Order object 67
Payment object 71
Credit object 74
Batch object 75
Administration objects used by Cassette for SET 76
Cassette object 76

Account object 77
Brand object 79
Address Verification Service (AVS) result codes 88

Appendix A. Certificate utility 91
CertUtil properties 91
Starting the CertUtil program 91
Running the CertUtil program 91

Appendix B. SaleDetail parameters 93

Appendix C. Purchasing card parameters 101

Appendix D. Retry mechanism used in WebSphere Commerce Payments 103
Purchase process overview 103
WebSphere Commerce Payments retry protocol 103
Configuring the retry characteristics 105
Checking for failures 105
Implementing a strategy 106
Other types of errors 106

Appendix E. Configuring the 4758 109
Prerequisite product levels 109
Configuring the 4758 PCI cryptographic coprocessor and CCA support program 109
Coexistence considerations 110
4758 role and profile requirements 110
Expiration of 4758 user profile passwords. 111
Accessing libraries and files. 112
Deleting CDSA ODM objects 112
4758 error codes. 113

Appendix F. Cassette for SET Return Codes 115

Appendix G. Cassette for SET Messages 137

Appendix H. Notices 159
Trademarks. 160

Glossary 161

Index 175

Preface

This book is for users and administrators of the SET Secure Electronic Transaction™ cassette who are responsible for its installation and implementation. This information will help you understand what you need to use the cassette for SET™. Programmers who are responsible for developing applications to manage IBM® WebSphere® Commerce Payments for Multiplatforms transactions that use the SET Secure Electronic Transaction payment protocol (SET protocol) will find the cassette-specific information regarding parameter requirements and cassette specific states useful.

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.
Italics	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 the experience of electronic shopping requires different players and software components:

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

brand Brand recognition and loyalty are the keys to credit card marketing. Some brands are owned by a single financial organization which is also the card issuer. Other brands are owned by bankcard associations, consortiums of financial institutions that promote and advertise the brand, establish operating rules, and provide a network for payment authorization and funds transfers. SET provides controlled access to these networks from the Internet.

cardholder
A person with a valid payment card account who uses software that supports electronic commerce.

certificate authority

Digital certificates are a security measure used to authenticate the different participants in the electronic commerce marketplace. Certificates exchanged among merchants, customers, and payment gateways validate the authenticity of the parties to each other. These certificates are provided by the certificate authority. The certificate authority uses a software application for handling certification management, routing, decrypting and encrypting, and other validation activities.

IBM's solution for the certificate authority software application is the IBM Payment Registry for AIX®.

electronic wallet

The electronic wallet is a software version of the wallet shoppers use to carry credit cards. With this electronic wallet as a browser plug-in or helper application, they can make purchases right on the Internet. The wallet is launched by the cardholder or by the merchant when the cardholder is ready to make a purchase.

IBM's solution for the electronic wallet is IBM Consumer Wallet ™ for Windows ® 95 and Windows NT®.

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

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

merchant server

This server allows merchants to display and sell products to a global market over the Internet. Shoppers can browse the 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.

IBM's solution for the merchant server is the IBM WebSphere Commerce Business Edition and WebSphere Commerce Studio, Business Developer Edition products.

payment gateway

Payment gateways represent the acquiring institution on the Internet. The merchant uses the payment information received from the cardholder to capture credit and reversal transactions and request authorizations from the financial institution. The information is sent to the financial institution and authorization information is received through the payment gateway.

IBM's solution for the payment gateway is the IBM Payment Gateway for AIX.

payment server

The payment server handles and stores payment information. All transactions from the consumer using the wallet flow directly to the merchant's payment server. 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.

IBM's solution for the payment server is the IBM WebSphere Commerce Payments for Multiplatforms.

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

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

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

Capture
Funds can be moved or deposited to the merchant's account. A capture places a transaction into a batch for settlement.

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

Open batch
Either the merchant or the acquirer opens a log in order to group together transactions and process them together at the same time.

Close batch
The merchant or acquirer closes the open log, possibly reconciles the totals against the other party's log, and closes the log. No more transactions are added to the log.

Additional information

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

- The *IBM WebSphere Commerce Payments for Multiplatforms Installation Guide* provides details on installing WebSphere Commerce Payments.
- The *IBM WebSphere Commerce Payments for Multiplatforms Administrator's Guide* provides details on configuring WebSphere Commerce Payments.
- The *IBM WebSphere Commerce Payments for Multiplatforms Programmer's Guide and Reference* provides details on the programming commands that serve as the interface to WebSphere Commerce Payments.

The information provided here supplements the above information; describing cassette-specific programming or administrative considerations and indicating how the cassette is mapped onto the WebSphere Commerce Payments's generic commands.

WebSphere Commerce Payments supports the SET Secure Electronic Transaction protocol to provide identification and enhanced security for payment card transactions conducted over the Internet.

For the latest Cassette for SET information, see the readme.set.html, which you can find using the documentation links on **http://www.ibm.com/software/websphere/paymgr/support** or by following the documentation link from the iSeries Payments task page at http://yoursystemname:2001/. A link to frequently asked questions (FAQs) is also provided on the support Web site.

For additional information on WebSphere Commerce payment processing products, see: **http://www.ibm.com/payment/**

For information on Payment Processing Services that support WebSphere Commerce Payments, see: **http://www.ibm.com/software/commerce/connect/.** This Web site will point you to the Financial Institutions that have established Payment Gateways to handle your WebSphere Commerce Payments payment processing needs.

SET Secure Electronic Transaction LLC Documents

SET was developed by MasterCard and VISA with IBM and other leading technology companies. Additional information concerning the SET protocol can be obtained directly from SET Secure Electronic Transaction LLC Web site:**http://www.setco.org/**. There are a number of documents available for your reference, including:

- *The SET Standard Book 1 Business Description*

  Provides a general overview of SET, its goals and objectives.
- *The SET Standard Book 2 Programmer's Guide*

  Describes fields and messages in detail and outlines processing windows.
- *The SET Standard Book 3 Formal Protocol*

  Provides a very detailed and in depth discussion of SET messages and fields.
- *External Interface Guide to SET Secure Electronic Transaction*

  Discusses issues that exist when developing interoperable SET implementations.

These documents, and others, are available for download from the SET Secure Electronic Transaction LLC Web site. Topics that fall outside the scope of the SET Specification books are addressed in SET Extensions available at that Web site.

Chapter 1. Introducing the SET Secure Electronic Transaction protocol

This chapter, we will look at how WebSphere Commerce Payments uses the SET Secure Electronic Transaction protocol (SET Protocol) to implement the generic payment processing model, commands, and objects defined by the WebSphere Commerce Payments framework. The SET protocol is a request/response protocol, where one entity (for example, a cardholder) makes a request and gets a response from another entity (for example, a merchant). The cardholder, merchant, and financial institution participate in the payment flows that make up a financial transaction.

First, we will look at how financial entities (merchants, cardholders, payment gateways) use the SET protocol and examine the SET messages that flow from:

- Merchant to certificate authority
- Merchant to cardholder
- Merchant to financial institution

Then, we will revisit those categories and show:

- How the message flows relate to WebSphere Commerce Payments API commands and objects
- How API commands lead to these message flows in conjunction with the objects

Understanding the SET protocol

To help you understand the SET protocol, this section provides an overview of the SET message exchanges between the entities involved in financial transactions: cardholders, merchants, financial institutions, and certificate authorities. By definition, overviews are incomplete. For a complete description of the SET protocol, see:

- *The SET Standard Book 1 Business Description*
- *The SET Standard Book 2 Programmer's Guide*
- *The SET Standard Book 3 Formal Protocol*

Figure 1 shows the entities involved in financial transactions:

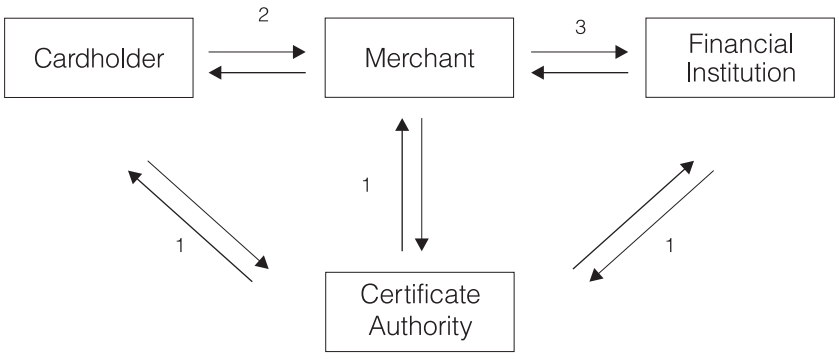


Figure 1. Financial entities

In Figure 1, the numbers represent the logical groups of message exchanges that occur between the entities:

1. “Certificate message flows” describes the messages exchanged where the certificate authority is the server and the financial entity is the client.
2. “Cardholder – merchant message flows” describes the messages exchanged where the cardholder is the client and the merchant is the server.
3. “Merchant – financial institution message flows” on page 3 describes the messages exchanged where the merchant is the client and the financial institution is the server.

Certificate message flows

When using SET protocol, each entity uses certificates to ensure security and data integrity. Before participating in financial transactions, each merchant must obtain certificates from the certificate authority (Note that `certless` transactions are allowed for the cardholder if the gateway supports it.):

1. Certificate requests begin with a merchant sending an initiation message (called a MeAqCInit Request) to the certificate authority, who replies with a MeAqCInit Response, which contains a certificate registration form.
2. The merchant completes the registration form, and sends it in a certificate request (CertReq) to the certificate authority.
3. If the request is successful, the certificate authority replies with a certificate response (CertRes). The CertRes contains the requested certificate information (for example, encryption and signature certificates).

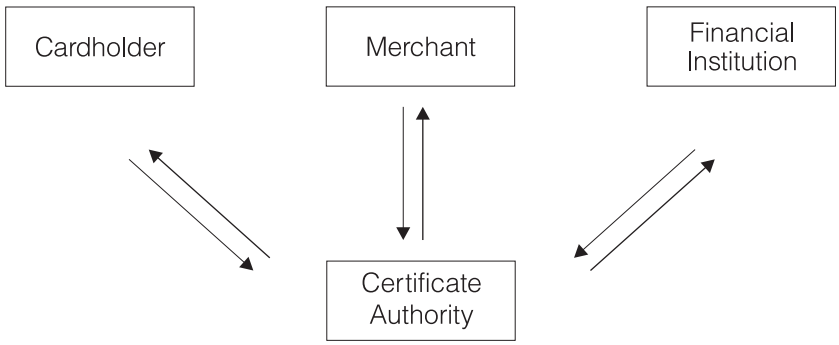


Figure 2. Certificate message flows

Cardholder – merchant message flows

The SET protocol allows consumers to make secure purchases through cardholder and merchant interaction. The cardholder–merchant interaction occurs for a set of flows, called three-party SET transactions, which include a SET wallet application.

Purchase transactions consist of message pairs sent by the cardholder and the merchant:

- When a consumer starts the transaction, an initiation message is sent from the merchant to the consumer that contains all the information about the purchase needed by the consumer to initiate the transaction. Note the initiation message is not part of the SET protocol; it is an out-of-band message.
- Purchase initiation exchange (PInitReq and PInitRes). In the PInitReq, the consumer identifies a brand (for example, a particular credit or debit card) to be used for the purchase. The PInitRes message contains the relevant certificate information needed for authentication.

- Purchase exchange (PReqand PRes). The PReq message is the actual purchase request and uses the certificate information needed for authentication. The PRes is verification to the cardholder and contains the status of the purchase (for example, success or failure).

In this transaction, the Consumer is using a SET protocol electronic wallet application. This diagram shows two entities (a cardholder and a merchant) where funds are to be transferred from the cardholder to the merchant:

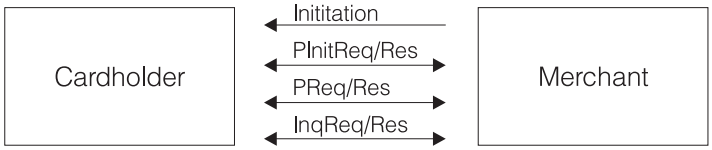


Figure 3. PInitReq/Res and PReq/Res messages

In addition to the purchase transaction, an inquiry exchange (InqReq/Res) can occur, where the cardholder sends an inquiry request (InqReq) to determine the status of a transaction. The merchant returns the status in the inquiry response (InqRes).

Merchant – financial institution message flows

Merchants authorize and capture funds by exchanging messages with a financial institution (or acquirer).

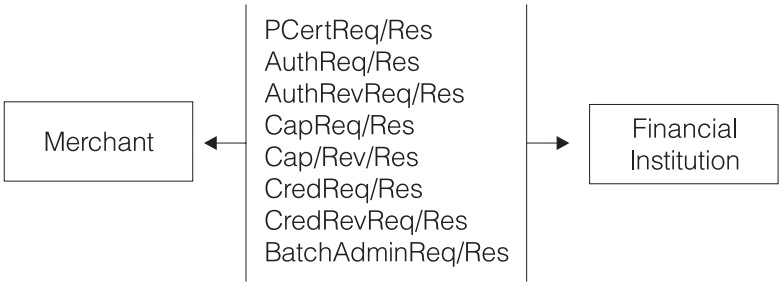


Figure 4. Merchant - financial institution message flows

The message exchanges, as shown in Figure 4, include:

- Purchase certificate request (PCertReq) and purchase certificate response (PCertRes), where the merchant asks for certificate information (in particular, the gateway encryption certificate) from the financial institution using the PCertReq and receives it in the PCertRes.
- Authorization requests (AuthReq) and authorization reversal requests (AuthRevReq) and their responses (AuthRes and AuthRevRes), which are used to authorize payment from the cardholder to the merchant. The AuthReq is a request to authorize transfer of funds from the cardholder to the merchant. The AuthRes contains the status of that request. The AuthRevReq is used to modify the amount of a prior authorization. The AuthRevRes contains the status.
- Capture and Credit exchanges are used to mark payments for settlement:
  - Capture messages arrange the transfer of funds from the cardholder to the merchant. The capture request (CapReq) marks a previously-authorized transaction for capture and associates that transaction with a batch. The capture reversal request (CapRevReq) removes the transaction from the

batch so funds will not be captured in the batch. The responses to these messages (CapRes and CapRevRes) return status of the request.

  - The credit request (CredRes) returns funds to the cardholder from the merchant and puts the transaction in the batch. The credit reversal request (CredRevRes) cancels a credit request and removes the transaction from the batch.

- Batch administration message exchanges are used to open or close batches, exchange status, or purge batches. For more information on batches, see “Batch processing” on page 13.

How do SET messages relate to the Application Programming Interface(API) and objects?

Having looked at the SET message flows, how do they relate to the Cassette for SET objects and the WebSphere Commerce Payments user interface (UI) and API? To help you understand their relationship, this section reviews merchant interactions with the other financial entities:

- “Merchant and certificate authority interaction”
- “Merchant and cardholder interaction” on page 5
- “Merchant and financial institution interaction” on page 6

Merchant and certificate authority interaction

Part of configuring a merchant to accept transactions for a given Brand involves obtaining a certificate from a certificate authority. Cassette for SET uses a brand object to encapsulate certificate and other relevant information. When Brands are manipulated using the WebSphere Commerce Payments UI (or directly by other merchant software that uses API commands), the certificate messages are exchanged.

When you create a Brand (using the UI or API commands), certificate messages flow between the financial entities. For example, Figure 5 on page 5 shows the message flows that occur when a merchant, using WebSphere Commerce Payments and its UI, creates a Brand. When the brand is created, certificate messages (MeAqCInitReq/MeAqCInitRes, and CertReq/CertRes) flow between WebSphere Commerce Payments and the Certificate Authority.

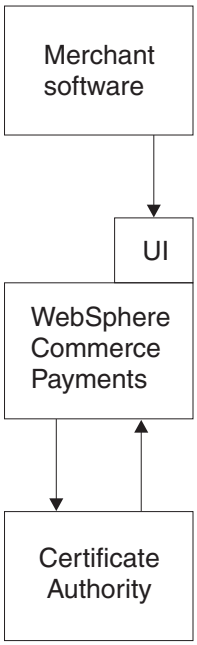


Figure 5. Certificate flows between WebSphere Commerce Payments and the Certificate Authority

Before a Merchant certificate is issued, the request is verified by the Merchant's Acquirer or payment brand authority. The certificates are obtained from the CA using this protocol:

- The Merchant initiates a request for encryption and signature certificates.
- The CA responds with a registration form.
- The Merchant completes and sends the registration form and public keys to the CA for processing.
- The Acquirer or payment brand authority verifies the Merchant request and the CA generates, signs and sends the certificate to the Merchant.

Merchant and cardholder interaction

This scenario shows the message flows involved in a purchase transaction.

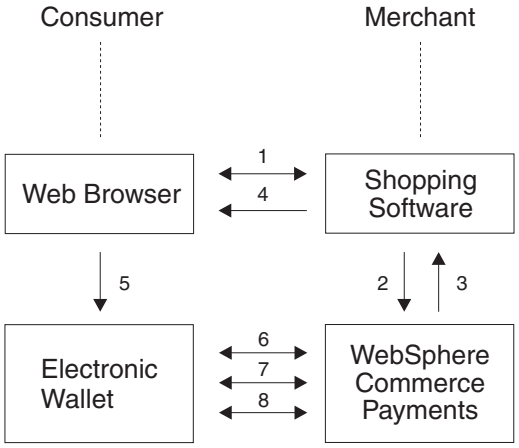


Figure 6. Purchase command sequence message flows

The message flows exchanged, as shown in Figure 6 on page 5, include:

1. Interaction between the Consumer and the Merchant shopping software concludes with the Consumer issuing a payment initiation request to the Merchant (that is, the consumer tells the Merchant that he is ready to purchase).
2. In response to the payment initiation request, the Merchant shopping software sends a payment initiation command (ReceivePayment) to WebSphere Commerce Payments. WebSphere Commerce Payments and the Cassette for SET create a new order with the correct attributes in the requested state.
3. WebSphere Commerce Payments responds with a payment initiation message which contains protocol-specific information for the Consumer's wallet program. The order is now ready for wallet messages.
4. The Merchant software forwards this payment initiation message to the Consumer's browser.
5. Based on the mime-type of the payment initiation message received, the Consumer's browser spawns the Consumer's wallet program. The browser passes the initiation message to the wallet.
6. The wallet and WebSphere Commerce Payments exchange the purchase initiation messages (PInitReq and PInitRes).
7. The wallet sends the purchase request message (PReq) to WebSphere Commerce Payments, which moves the order from Requested to Ordered state and sends the purchase response (PRes) back to the wallet.
8. At any time after the purchase transaction (via the Wallet), an inquiry exchange (InqReq/Res) can occur, where the cardholder sends an inquiry request (InqReq) to determine the status of a transaction. The merchant returns the status in the inquiry response (InqRes).

Notes:

1. Orders can be created without a wallet, using the AcceptPayment command. In this case, all cardholder information normally obtained during wallet flows (for example, brand and card expiration) must be obtained by the merchant directly from the cardholder and passed to WebSphere Commerce Payments on the AcceptPayment command. For more information on commands, see Chapter 7, "Using WebSphere Commerce Payments commands with Cassette for SET" on page 47.
2. After the message flows are complete, the order is in the Ordered state and the merchant can authorize and capture payments for this order with the financial institution. No money has yet changed hands, but will in subsequent message flows.
3. The above scenario shows order creation without interaction with the financial institution. In step 7, WebSphere Commerce Payments can authorize or capture funds between the receipt of the purchase request (PReq) and the sending of the purchase response. In this case, the cardholder will receive additional information regarding authorization and capture in the purchase response.

Merchant and financial institution interaction

This scenario shows the messages exchanged when a Merchant wants to approve funds and later deposit funds:

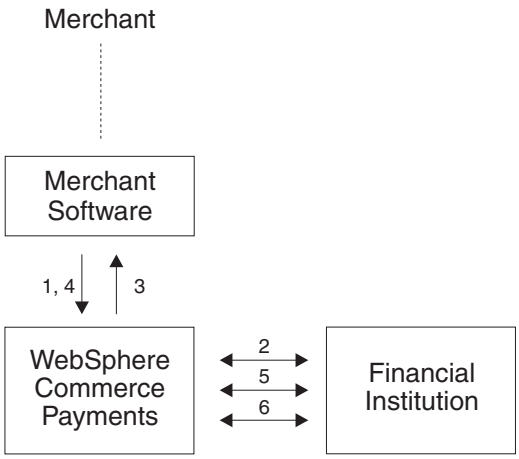


Figure 7. Merchant approval of funds with deposit occurring later

1. The merchant software issues an Approve command (through the UI or API) to WebSphere Commerce Payments.
2. WebSphere Commerce Payments sends an AuthReq message to the financial institution and receives an AuthRes. This is an authorization exchange.
3. If the AuthRes message indicates payment approved, WebSphere Commerce Payments creates a new payment in Approved state and returns a successful response to the merchant software.
4. When the merchant is ready to capture funds for the approve payment, he issues a Deposit command to WebSphere Commerce Payments.
5. The cassette selects a batch into which this payment will be deposited. If no suitable batch is currently open, a batch can be opened with the Batch administration exchange (BatchAdminReq/Res) between WebSphere Commerce Payments and the financial institution.
6. WebSphere Commerce Payments performs a capture exchange (CapReq/CapRes) with the financial institution. This places the payment in the batch for subsequent settlement.
7. If the capture response is successful, the payment moves to Deposited state and success codes are returned to the merchant software.

Notes:

1. Like the Approve command shown above, ApproveReversal, Deposit, DepositReversal, Refund, RefundReversal, BatchOpen, BatchClose, and BatchPurge commands each result in one request/response exchange with the financial institution, as shown in Table 2 on page 8.
2. In the scenario above, the merchant performed authorization and capture at two distinct times. The authorization and capture can occur in one step if the merchant issues an approve with autodeposit (DEPOSITFLAG=1). Depending on the capabilities of the financial institution, authorization may take place in one message exchange (AuthReq/Res) or two exchanges (AuthReq/Res & CapReq/Res).
3. The purchase certificate exchange (PCERT/REQ) occurs between WebSphere Commerce Payments and financial institution, automatically when needed. No action is required by the merchant software.

Mapping WebSphere Commerce Payments commands to SET messages

In the scenarios previously described, the transactions that occur between the financial entities are conducted using WebSphere Commerce Payments commands. Those commands invoke the various financial functions using the SET messages shown in Table 2.

Table 2. WebSphere Commerce Payments commands mapped to SET messages

WebSphere Commerce Payments Command	Financial Function	SET Messages Exchanged
Approve	authorize	AuthReq, AuthRes
ApproveReversal	authorize reversal	AuthRevReq, AuthRevRes
Deposit	capture	CapReq, CapRes
DepositReversal	capture reversal	CapRevReq, CapRevRes
Refund	credit	CredReq, CredRes
RefundReversal	credit reversal	CredRevReq, CredRevRes
BatchOpen	open a new batch	BatchAdminReq, BatchAdminRes
BatchClose	close an existing batch	BatchAdminReq, BatchAdminRes
BatchPurge	remove all payments from an existing batch, leave batch in open state	BatchAdminReq, BatchAdminRes

Chapter 2. Understanding SET protocol 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 abstract away protocol-specific differences so that merchants can ignore differences between protocols.

This section shows how the Cassette for SET organizes SET protocol configuration within the WebSphere Commerce Payments framework:

- "Administration objects" introduces the administration objects, Cassette, Account, and Brand, in which the cassette, financial institution, and certificate configuration is encapsulated.
- "SET profiles and acquirer behavior" on page 11 discusses the different ways the financial institutions use the SET protocol and how the Cassette for SET accommodates those differences.
- "Batch processing" on page 13 concludes with a discussion of batch management.

Administration objects

As described in the *WebSphere Commerce Payments Administrator's Guide* , WebSphere Commerce Payments provides a set of administrative objects that are used for cassette configuration. Each cassette extends these objects, and where appropriate, introduces their own objects. In our case, the Cassette for SET extends the Cassette and Account objects, and introduces the Brand object.

The extensions to the Cassette object are straightforward and contain all cassette-wide configuration. Cassette objects are described in detail in Chapter 8, "Using WebSphere Commerce Payments objects" on page 67.

The Account object represents the relationship between a merchant and financial institution. In the SET world, the financial institution consists of a payment gateway (the Internet point-of-presence for a financial institution) and an acquirer (the financial processor that authorizes and captures transactions). The Cassette for SET extension of the Account object records information describing the payment gateway and the acquirer.

The entities involved in typical financial transactions, and their relationship to WebSphere Commerce Payments administration objects are shown in Figure 8 on page 10. The Account object represents the payment gateway and financial institution. For completeness, Brand objects, which are extensions to the Account object, are also shown and will be discussed later.

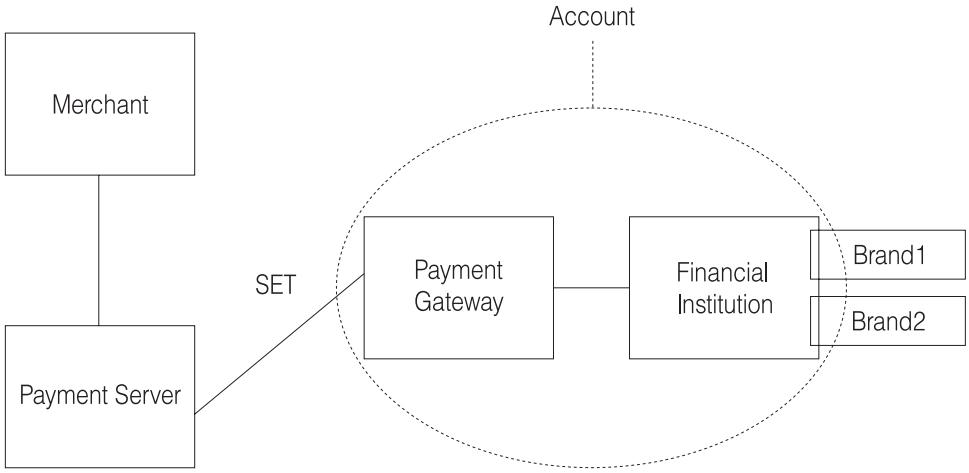


Figure 8. Cassette for SET objects — accounts and brands

The Cassette for SET extension to the Account object includes:

- Payment gateway connectivity information (for example: hostname, port, and URI) describing where WebSphere Commerce Payments sends SET request messages.
- Acquirer behavior: the SET profile. As discussed in “SET profiles and acquirer behavior” on page 11, there are many ways for an acquirer to use the SET protocol. The particular behavior of an acquirer is reflected in the SET profile in the Account extension.

For more information on the Account object, see “Account object” on page 77.

As described in the SET specification, the SET protocol relies on public key encryption that uses financial institution, merchant, and cardholder certificates. In the SET protocol, certificates are distributed by Brand Certification Authority for each brand (for example, VISA and Mastercard). Based on per-brand distribution of certificates, Cassette for SET defines a Brand object to encapsulate the information specific to each brand. Brand objects are always associated with Account objects, as shown in Figure 8. Information defined for the Brand object includes:

- Brand name
- Certificate identifiers (for example, BIN, Merchant ID, Acquirer business ID)
- Certificate authority connectivity information
- Merchant encryption and signature certificates, and gateway encryption certificates

The Brand object is a MerchantCassetteObject, as defined in the *WebSphere Commerce Payments Programmer’s Guide and Reference* . Brand objects can be managed through the WebSphere Commerce Payments user interface, as described in the tutorial in Chapter 5, “Getting started” on page 33. Through the API, Brand objects are manipulated by CreateMerchantCassetteObject, ModifyMerchantCassetteObject, and DeleteMerchantCassetteObject commands. For more information on creating and manipulating brands and certificates through the API commands, see Chapter 7, “Using WebSphere Commerce Payments commands with Cassette for SET” on page 47. For more information on the Brand objects, see “Brand object” on page 79.

Financial objects

The SET cassette provides extensions for the financial objects, which are used to conduct SET financial transactions. Financial objects include:

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

For details on how the SET cassette extends these financial objects, see Chapter 8, “Using WebSphere Commerce Payments objects” on page 67.

For descriptions of the financial objects as viewed by the WebSphere Commerce Payments Framework, see the *WebSphere Commerce Payments Administrator’s Guide* .

For commands provided by the Framework financial programming information, see the *WebSphere Commerce Payments Programmer’s Guide and Reference* .

SET profiles and acquirer behavior

Acquirers use the Secure Electronic Transaction protocol in a variety of ways. In some cases, acquirers require merchants to use particular fields or messages. In other cases, acquirers do not allow merchants to use specific fields or messages. In addition, various acquirers have interpreted parts of the SET specification differently. WebSphere Commerce Payments needs to be able to work in each of these different ways. The SET Profile for an Account tells WebSphere Commerce Payments how to use the SET protocol when sending messages to the acquirer for that account. The SET Profile is selected in the WebSphere Commerce Payments configuration when the account is created. To determine the value for the SET Profile, ask your acquirer which profile corresponds to the behavior expected by your acquirer.

The Cassette for SET has defined several profiles that reflect the behaviors of various acquirers. The merchant must determine the characteristics of the acquirer and specify the appropriate profile. The behaviors encoded in the Cassette for SET profiles are flexible and can expand as new acquirer behaviors become commonly used.

Note: The list of profiles is growing as more acquirers support SET or change their policies. A WebSphere Commerce Payments administrator must periodically check to make sure that the acquirer’s policies still match the SET profile setting, and check the latest profile list when software updates arrive. For a link to where you can find a description of the current list of profiles and their behaviors, see **http://www.ibm.com/software/websphere/paymgr/support**.

Of the currently documented behaviors, these merit closer examination:

Acquirer behavior encoded in a SETProfile

Acquirer Setting	Overview and Implications
Acquirer Supports AuthToken	If the acquirer supports AuthTokens, multiple payments per order can be made. If AuthTokens are not supported, only one payment can be approved per order.

Acquirer behavior encoded in a SETProfile

Batch Control	There are two styles of batch control observed in acquirers and supported by WebSphere Commerce Payments: merchant controls batch (MCB) and acquirer controls batch (ACB), as discussed in “Batch processing” on page 13.
Sale Transactions	An acquirer that supports sale transactions can both authorize and capture a payment with one message. When the merchant specifies AutoDeposit (DEPOSITFLAG=1) on an API command, this setting determines whether WebSphere Commerce Payments sends one message or two to the acquirer.
Batch Status	An acquirer that supports batch status can send totals upon merchant request. If supported, WebSphere Commerce Payments will always balance the batch before closing the batch. If the totals do not agree, the batch will not close.
Batch Purge	An acquirer that supports batch purge will remove all transactions from a batch upon receipt of a purge request from a merchant. The merchant can only issue batch purge requests if batch purge is supported.

Split payments

Suppose a customer contacts an online catalog store and orders $80 worth of merchandise. The merchant checks the inventory and finds that only $60 worth of merchandise is in stock and can be shipped. The merchant would like to collect $60 now and the remaining $20 when the rest of the order is filled. WebSphere Commerce Payments is designed to support payment systems in which customers provide payment information once (for the entire $80) and the merchant collects the funds over time ($60 now and $20 later). This is referred to as *split payments*.

The acquirer must support split payments in order for the merchant to be permitted to approve multiple payments. If your acquirer supports returning AuthTokens, you can select a SET profile where multiple payments are allowed. AuthTokens are required to support split payments.

Split payments are created using the Approve and ApproveReversal API commands with the $SPLITALLOWED parameter set to 1. The basic split payment scenario described above is performed by doing two calls to Approve, one for $60 and one for $20, to create two payments. A more interesting scenario occurs when orders are automatically approved for the full amount and the inventory is checked later. In this case, the first payment for $80 is created automatically. If the inventory check reveals that $20 worth of merchandise is not in stock, the merchant issues an ApproveReversal command call with an amount of $60 (thus changing the amount of the first payment to $60), and then issues an Approve command for $20 to create the second payment.

Batch processing

In the credit card industry, a *batch* is a collection of financial transactions grouped together for administrative and record-keeping purposes. In WebSphere Commerce Payments, a batch is a collection of payments and credits that will be processed and cleared as a group by a single financial institution. *Batch processing* is the term used to refer to the overall process in which batches are created, specified, balanced, and submitted. *Reconciliation* is the process that a merchant and a financial institution use in a situation where the merchant's batch total and the bank's batch total do not match. Reconciliation is often a manual process that involves going through transactions and doing a line-by-line comparison.

In the previous section, we discussed the different ways acquirers use the SET protocol. The biggest difference and the place where the SET protocol is the most vague is in batch processing. For example, when using the SET protocol, batches are opened and closed and identified with a batch ID. Some of the choices you face when using the SET protocol include:

- Which side (merchant or acquirer) is responsible for opening a batch?
- Which side assigns batch IDs?
- Which side selects a batch for captures and credits?
- Which side closes a batch?

To reduce the number of decisions you must make when using SET protocol, IBM has worked with other vendors to define two styles of batch control (also known as acquirer behavior):

merchant controls batch (MCB)
: The merchant is responsible for all batch management decisions. The merchant must open batches explicitly and identify the batch ID (using a batch administration message exchange). The merchant identifies the batch for each payment and credit (by specifying the batch ID in all capture and credit message exchanges). The merchant is responsible for closing all batches (using another batch administration message exchange).

acquirer controls batch (ACB)
: The acquirer is responsible for all batch management decisions. The acquirer opens and closes all batches, the merchant is prohibited from using batch administration messages. The acquirer selects batches for payments or credits. It is assumed that the acquirer has only one open batch at any time. The acquirer is responsible for sending the batch ID to the merchant in capture or credit responses.

Note: With ACB, the acquirer identifies the batch for each transaction, and only one batch can be open at a time. When a new batch ID (that is, a batch ID that has not been previously received) is returned to the merchant from the acquirer, the merchant must assume the previous batch has been closed and a new batch has been opened.

The MCB and ACB are styles of acquirer behavior for using the SET protocol for batch management. We now examine how these two styles of acquirer behavior affect the user interface and API usage.

On the user interface, the only difference between ACB and MCB acquirer behavior is whether the batch can be closed from the user interface Settle screen. Since the merchant is prohibited from closing the batch for many ACB acquirers, batches for those acquirers may not appear on the Settle screen.

With respect to the API, there are many differences in the way the API can be used. In particular, the API commands can be used for explicit batch management and implicit batch management. Acquirer behavior (MCB or ACB) impacts the choice of API style, as well as how the commands are mapped to SET messages:

Explicit API style

Explicit API style can only be used with MCB acquirers. All of the merchant decisions (opening and closing batches, assigning payments and credits to batches) are exported through the API to the merchant software. The merchant opens batches using the BatchOpen command, closes batches using the BatchClose command, and must specify a batch number on all Deposit and Refund commands.

Implicit API style

Implicit API style can be used with both MCB and ACB acquirers. The merchant does not issue batch open commands and does not specify a batch number on deposit or refund commands. For ACB acquirers, WebSphere Commerce Payments forwards requests to the acquirer, allowing the acquirer to manage batches. For MCB acquirers, WebSphere Commerce Payments manages the batches, as necessary.

Figure 9 shows three entities, the merchant software, WebSphere Commerce Payments, and the payment gateway. The entities are divided between merchant and acquirer side and between merchant software or WebSphere Commerce Payments based on where decisions are made. The vertical line divides the merchant and the acquirer. Decisions are made on the merchant side for MCB acquirers, and on the acquirer side for ACB acquirers.

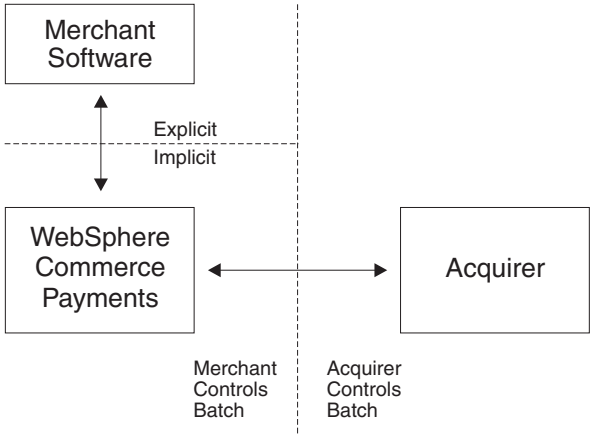


Figure 9. Implicit or explicit batch processing

The horizontal line divides the merchant software and WebSphere Commerce Payments. Decisions are made by the merchant software when explicit API style and by WebSphere Commerce Payments with implicit API style.

The various combinations of API style and acquirer behavior are shown in Table 3.

Table 3. API style and acquirer behavior matrix

API Style	**Acquirer behavior**	
	Acquirer Controls Batch	Merchant Controls Batch
Explicit		x
Implicit	x	x

Although the matrix shows four possible combinations of API style and acquirer behavior, explicit batch processing is not possible with ACB, because the acquirer makes batch decisions and there is no way to export those decisions to merchant software.

The other interesting combination is implicit API style for an MCB acquirer. In this case, WebSphere Commerce Payments handles any needed batch processing on behalf of the merchant software. In particular, on a deposit or refund API command, WebSphere Commerce Payments will attempt to locate a suitable batch for the request. If no batch is available, WebSphere Commerce Payments will open a batch, using a batch administration message exchange.

The behavior of WebSphere Commerce Payments with respect to batch management for an account is determined by the SET profile for that account. Again, there are no choices with regard to the SET profile configuration settings; they are determined by the acquirer. Batch behavior is just one of the financial institution-specific policies that is set in WebSphere Commerce Payments (see "SET profiles and acquirer behavior" on page 11).

Note: The SET Cassette does not currently support implicit MCB with multiple currencies per batch.

Batch reconciliation

At various intervals, the batch is closed and the actual transfer of funds takes place. At this time, the financial institution and the merchant must reconcile the batch before transferring any funds. Batch reconciliation is the process of adjusting the merchant's or financial institution's view of the batch until they agree. If the acquirer supports the exchange of totals and summary information, a simple totals exchange is done before the batch is closed to compare the number of transactions and the total amount in the batch.

If the batch does not balance (the batch totals exchanged do not agree), action is required. The merchant must work directly with the financial institution to determine the cause of the discrepancy. When a merchant establishes a business relationship with an financial institution, provision is made for dealing with errors of this type. In the pre-Internet point-of-sale world, this was frequently done over the phone.

To reconcile a batch that does not balance, the merchant and financial institution must examine their batch contents. You can use the WebSphere Commerce Payments user interface or QUERY commands to view the contents of the batch as seen by the merchant. The merchant can issue payment commands as necessary to correct the batch contents. Deposit and Refund reversals can be used to correct the amounts of specific transactions or remove them entirely from the batch. New Deposit and Refund API commands can be issued to add missing transactions. After these transactions are made, the merchant can try again to close this batch. This process can be repeated until the totals match.

If the acquirer supports this, you can use the BatchPurge command when batches do not balance. The BatchPurge command removes all payments and credits from the batch, returning the batch to an empty state. BatchPurge can be used when the offending payments or credits can not be identified. After using the BatchPurge command, use deposits and refunds to reconstruct the batch.

Note: It is highly recommended that you use the QueryBatches command before purging the batch.

Support for BatchPurge is configured in the SET profile, not all acquirers support this command. For batches that can be purged, there will be a Purge button on the user interface. Programmers can identify batch objects that can be purged because those batch objects will have the PurgeAllowed field set to true.

Chapter 3. Understanding concepts unique to Cassette for SET

This section describes some of the other concepts that are unique to the Cassette for SET support of the SET protocol.

Order creation

The Cassette for SET supports order creation both with wallet participation (using the ReceivePayment command) and without wallet participation (using the AcceptPayment command). Before a merchant creates an order using an AcceptPayment command, the merchant software must collect cardholder information that would otherwise be collected during wallet message exchanges. In particular, the merchant must collect the cardholder's brand, account number, and expiration date, as described in Chapter 7, "Using WebSphere Commerce Payments commands with Cassette for SET" on page 47. This information is passed using $BRAND, $PAN, and $EXPIRY parameters.

Additionally, for both the ReceivePayment and AcceptPayment commands, an order summary must be passed using the $ORDERDESCRIPTION parameter.

Merchant initiated authorization

The Merchant Initiated Authorization (MIA) SET extension permits a merchant to use SET messages for authorization and capture on orders that were placed by the cardholder using a transmission method other than SET. Since these messages will not carry a cardholder certificate, the gateway certificate must indicate that certless purchases are supported.

The MIA extension TransMethod field carries a defined value representing the transmission method used by the merchant to collect card data from the consumer. The value for this field will default to **4** (MIA_otherElectronic). Merchants can override the default using the new protocol data keyword, $TRANSMETHOD. The optional TransCrypto field is always set to null by WebSphere Commerce Payments. For more information, see "AcceptPayment" on page 47.

Prior to the definition of the MIA extension, WebSphere Commerce Payments implemented the IBM-defined Merchant Originated Payment (MOP) extension. To allow time for gateway installations to migrate to the official MIA extension, WebSphere Commerce Payments will build and send both the MIA and the MOP extensions on every merchant originated AuthReq. Every order created in WebSphere Commerce Payments via the AcceptPayment command is considered merchant originated. If a merchant is already using AcceptPayment, no change is required to cause both the MIA and MOP extensions to be sent. Both are noncritical extensions and should be ignored by gateways that do not implement this support.

More information on SET extensions and on the MIA extension can be found on the SET Secure Electronic Transaction LLC Web site at **http://www.setco.org/**.

SaleDetail and AVSData

SaleDetail and AVSData are optional structures defined by the SET protocol. Both provide a mechanism to specify an additional level of order-specific information. AVSData is used to specify the cardholder's billing address. It is sent to the Payment Gateway in the AuthReq. The financial institution can compare this address to the cardholder's address on record, providing a higher level of security. SaleDetail collects data regarding items like a cardholder's hotel charges, auto rental charges, or transport charges. In WebSphere Commerce Payments, AVSData and SaleDetail information is collected at order creation time and passed to the payment gateway in the AuthReq and CapReq messages, respectively.

SaleDetail and AVSData appear as optional protocol data parameters on ReceivePayment and AcceptPayment commands. For a complete listing of SaleDetail parameters, see Appendix B, "SaleDetail parameters" on page 93.

Although optional, certain dependencies exist within the SaleDetail and AVSData structures. For a listing of required, conditionally required, and mutually exclusive fields, see the *SET Secure Electronic Transaction Specification Book 2: Programmer's Guide.*

Purchasing cards

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

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

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

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

1. An employee makes a purchase directly from the supplier through a purchasing card.

2. The employee receives a monthly purchasing card statement, verifies the charges, and sends it on to accounts payable.
3. Accounts payable reviews and approves the statement. The statement may include general information about the purchase (such as the purchase date, amount, commodity code, merchant or supplier information, tax information). It may also include details about the line items associated with the purchase (such as the item quantity, unit of measure, part numbers, description, unit cost, and tax information). This purchase information is known in the e-commerce industry as Level I, Level II, and Level III information.

The benefits of using purchasing cards include:

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

Levels of purchasing card data supported

The Cassette for SET supports three levels of purchasing card information as follows:

Level I and Level II data

Standard commercial transaction data for the purchase (beyond what the WebSphere Commerce Payments Framework provides) including: freight amount, duty amount, commodity code, merchant identification and location, reference number for the charge or credit, and other data elements. Also, additional data about the purchase, such as: merchant type, merchant tax ID, sales tax amount, merchant zip code, and other data elements.

Level III data

Full line-item detail in addition to the data in Level II which includes unit cost, quantities, unit of measure, product codes, product descriptions, tax amount and rate, discount amount, and other data elements.

For a list of purchasing card data supported by the Cassette for SET, see Appendix C, “Purchasing card parameters” on page 101.

Purchasing card data is logically associated with a payment, but is not displayed in the WebSphere Commerce Payments graphical user interface.

Purchasing card data processing

Purchasing card data can be passed on the AcceptPayment, ReceivePayment, and/or Deposit commands only.

- When purchasing card data is passed in on the AcceptPayment or ReceivePayment command, the data is associated with the order, and multiple payments are not allowed. The $SPLITALLOWED keyword must be ″0″ (indicating multiple payments are not allowed) on the AcceptPayment and ReceivePayment API commands.

At Capture (Deposit) time, if no purchasing card data is specified on the Deposit command, the Cassette for SET sends the purchasing card data authorized with the cassette's Order.

- When purchasing card data is passed in on the Deposit command, the data is associated with the payment. Purchasing card data passed in with the Deposit command takes precedence over purchasing card data passed in through the AcceptPayment command.

Therefore, if you need to support multiple payments per order and purchasing cards, purchasing card data must be passed in at Deposit time. If you need to support purchasing cards and do automatic authorization or automatic deposits with the AcceptPayment and ReceivePayment commands respectively, purchasing card data must be passed at AcceptPayment or ReceivePayment time.

Relationship to SaleDetail

Purchasing card data is actually a subset of the SaleDetail data structure defined by the SET protocol. When purchasing card protocol data is specified, it is sent to the Payment Gateway in the SaleDetail data structure. Although there is some overlap of the type of data supported, you should be aware of these differences:

- The SaleDetail data structure includes both travel and expense (T&E) and non-T&E data. For the Cassette for SET, purchasing card data does *not* include travel and expense information. Therefore, if you need to pass travel and expense purchasing card data (information about auto rentals and hotel stays), you must use the SaleDetail protocol data parameters.
- The SaleDetail protocol data does not support multiple line items. If you were to use SaleDetail protocol data, only one line item can be associated with the Order. If you need to pass information about more than one line item in an Order, you must use the purchasing card protocol data parameters.
- The SaleDetail protocol limits all character data to ASCII. The purchasing card protocol supports ASCII and UTF-8, as appropriate (as defined by the SET protocol).

Sale transactions

A sale transaction refers to the authorization and capture of a transaction with a single SET message. Some acquirers support (or require) that the merchant authorize and capture transactions at the same time by sending a SaleTransaction request, which is done using an AuthReq (authorization request) message with the CaptureNow flag set to true.

The SET Cassette knows when Sale Transactions are required by the value of the SET Profile in the associated account. When the SET Profile indicates that the acquirer supports Sale Transactions, Approve API commands with auto-Deposit set to true cause a sale transaction to be sent to the Acquirer. For acquirers that do *not* support sale transactions, the cassette sends an AuthReq followed by a CapReq. This could result from one of these API commands:

- ReceivePayment with AutoApprove and AutoDeposit set to true
- AcceptPayment with AutoApprove and AutoDeposit set to true
- Approve with AutoDeposit set to true

Note: References to the *AutoApprove* option means that APPROVEFLAG = 1 or 2. The *AutoDeposit* option means that DEPOSITFLAG = 1.

When the API command with AutoDeposit set to true completes successfully, the payment will be in Deposited state. A transaction in Deposited state can be reversed by issuing a DepositReversal command. If this was a Sale Transaction, the DepositReversal will result in the reversal of both parts of the Sale Transaction: the authorization and the capture. The SET Cassette will send an AuthRevReq message with CaptureNow set to true to the acquirer. When the reversal completes successfully, the payment will be moved to Void state. Both the authorization and capture will have been reversed. Note that for non-Sale transactions only the deposit is reversed, placing the payment back into approved state.

In summary, for sale transactions:

- Make sure the SET profile indicates that the acquirer supports sale transactions.
- To send a Sale Transaction, issue an Approve command with DepositFlag=1.
- To reverse a Sale Transaction, issue a DepositReversal.

Understanding 2KP and 3KP

SET payment flows use certificates to identify the parties involved with a particular payment. The merchant and gateway are always required to have certificates. The presence of a cardholder certificate is optional. Payment flows which involve certificates for all three parties in a transaction are often referred to as *3KP*, where **KP** stands for *key pairs*. Payments which do not involve a cardholder certificate are referred to as *2KP*. 2KP is also known as ″certless purchase″. The gateway certificate includes a flag which indicates whether or not it will accept 2KP transactions.

All merchant-initiated authorizations are 2KP flows. Merchant-initiated transactions are transactions in which cardholder data is collected by the merchant, through means other than the exchange of SET messages. A consumer with a SET wallet may also initiate a 2KP flow by choosing to pay with a credit card, for which no SET certificate has been acquired. Some merchants who support MIA or MOP also want to be able to enforce consumer use of a SET certificate when a SET wallet is used. Since the gateway certificate in this case indicates that 2KP flows are accepted, the certificate cannot be used to reject certless purchases from a SET wallet.

To enforce 3KP transactions with SET purchases involving wallets, a merchant may specify the $REQUIRECARDCERT boolean parameter on the ReceivePayment command. Setting $REQUIRECARDCERT to 1 indicates that an incoming PReq from a wallet must contain a cardholder certificate. If the PReq does not contain a certificate, a SET error message is returned to the wallet.

Extracting the SET initiation message

When using the ReceivePayment command to create orders with wallet participation, WebSphere Commerce Payments composes a SET initiation message (that is, Wakeup), which must be sent to the cardholder. This initiation message is an HTML page (as described in the *SET External Interface Guide*), which is included as packaged content in the XML response to the ReceivePayment command. The XML response is URL-encoded. Before sending the initiation message to the card holder, the merchant software must decode the message so the cardholder receives content (for example, OrderDescription) identical to that sent to the merchant by the cardholder software. The merchant software must:

1. Process the XML response
2. Extract the packaged content.

3. Forward the initiation message to the cardholder.

Note that merchant applications which use the Java™ Client Application Library (CAL) do not have to perform these steps since CAL does it for them.

For descriptions of the rules for URL encoding and decoding, see the *WebSphere Commerce Payments Programmer's Guide and Reference* .

Using the IBM WebSphere Commerce Payments SET JPO Extension

The Japanese Payment Option Extension is an extension to the SET protocol that supports a variety of payment options that are available at the point of sale in Japan. The SET protocol is used differently in Japan by cardholders, merchants and financial institutions. In particular, the Japanese payment option (JPO) is used in both cardholder merchant flows and merchant financial institution flows. The JPO Extensions are documented in ″SET Secure Electronic Transaction Specification Support for Japanese Requirements″ available at:
http://www.setco.org/extensions.html

The IBM WebSphere Commerce Payments SET JPO Extension is provided with the Japanese version of the Cassette for SET. For instructions on how to enable the JPO Extension, see the ″Japanese Payment Option Installation and Configuration″ at **http://www.setco.org/extensions.html**.

Configuring the host address in wakeup message

If your WebSphere Commerce Payments is behind a firewall, the hostname in the SET initiation message must be different than the hostname of the machine on which WebSphere Commerce Payments is installed. WebSphere Commerce Payments provides an ETillHostname field, which is intended to be used as the domain name server (DNS) name to send messages to WebSphere Commerce Payments. If the ETillHostname field is set, the Wakeup message includes it as the hostname in the SET-SET-URL and the SET-Query-URL. The Cassette for SET uses this field to override the DNS lookup.

To change the ETillHostname field, use the ModifyPayServer administration command. For details on viewing this field using query commands, see the WebSphere Commerce Payments Basic Settings window in the WebSphere Commerce Payments user interface (note that you must have Payments administrator access to view this window).

Event notification for SET

Cassette for SET supports two types of events:

State change event
When the state of the financial objects belonging to the Cassette for SET change, a state change event will be sent to the interested merchant software. Note, however, that these events are driven by changes in the Framework objects, even though the Cassette for SET maintains its own interval state in its own objects. For the mappings between Cassette for SET and Framework states, see Chapter 8, “Using WebSphere Commerce Payments objects” on page 67.

SET Cassette specific event
Cassette for SET supports one cassette specific event. This event signifies

completion of Payment Initialization (PI) with the wallet. Specifically, when the Cassette for SET receives a PInitReq, it sends a cassette specific event with these name and value pairs:

- EventType=2
- TimeGenerated=<time_stamp>
- OrderNumber=<order_number>
- MerchantNumber=<merchant_number>
- Message=″PInitReq received″
- CassetteName=SET

One potential use of this event is to let the merchant know that the customer did not abandon the order and allow the merchant software to update his browser with this information. For example, when the merchant software receives this event, the merchant software could send a MIME multi-part message back (including the wallet invocation) that displays a new screen and shows that the PI has been accepted.

For detailed information on event types, event contents, and how to register to receive the events, see the *WebSphere Commerce Payments Programmer's Guide and Reference* .

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 SET:

Table 4. Sensitive data processed by Cassette for SET

Data	How data is protected
$PAN	Cardholder's card number. All but the last 4 digits of the card number are masked with asterisks.
$EXPIRY	Card expiration date. The entire value is masked with asterisks.
$CARDVERIFYCODE	Verification code for the payment card. The entire value is masked with asterisks.
$AVS data	Address verification service (AVS) data. The entire value is masked with asterisks. (For a list of AVS data elements, see the description of the AcceptPayment or ReceivePayment API commands in Chapter 7, "Using WebSphere Commerce Payments commands with Cassette for SET" on page 47.)

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*.

Chapter 4. Installing the Cassette for SET

WebSphere Commerce Payments must be installed before the Cassette for SET can be installed. The WebSphere Commerce Payments installation will ensure that all prerequisite products are available. 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 the Cassette for SET

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

  Note: iSeries does not require that WebSphere Commerce Payments or WebSphere Application Server be ended during installation.
- Understand the migration considerations discussed in the *WebSphere Commerce Payments Installation Guide*. See also the Cassette for SET migration tips that follow.

  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 SET, you must install the version 3.1.3 Cassette for SET software for your cassette data to be migrated and compatible with the version 3.1.3 Framework.

Cassette for SET migration tips

In addition to the information provided in the *WebSphere Commerce Payments Installation Guide*, the following information is important to your Cassette for SET migration.

Migration support

The WebSphere Commerce Payments installation program migrates existing Payment Manager Version 2.2.x data. The installation program does not support migration from Payment Manager Version 2.1.x or Payment Server 1.2.

Time required to migrate

The time required to migrate WebSphere Commerce Payments (or Payment Manager) data is dependent on the number of orders in your existing payments database. Migrating a database with 1000 orders will take approximately 30 minutes.

PAN and Expiry fields

After migration, the PAN and Expiry fields will show up as null values in the WebSphere Commerce Payments user interface for migrated orders with multiple payments and for migrated orders on which an approveReversal has occurred. These orders are still fully functional.

Orders migration

1. Orders that are in WakeupSent or PinitResSent state do not get migrated.
2. If the order is in Pending state, we migrate the order but change the state as follows:
   - The order moves into Ordered state if there are no payments associated with the order.
   - The order moves into Refundable state if the Acquirer controls the batch and the payment has been captured, cap reversed, or closed.
   - The order moves into Refundable state if the merchant controls the batch and the payment is closed.

Installing Cassette for SET (Windows and UNIX® platforms)

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

Then, follow these procedures to install the Cassette for SET:

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

1. Insert the CD-ROM containing the Cassette for SET.
2. Select the directory for your platform:
   - For Windows NT and Windows 2000, got to the `nt` directory
   - For Solaris, got to the `solaris` directory
   - For AIX, got to the `aix` directory
3. Enter **InstallSETCassette** to start the installation. Enter the information requested on the installation windows.
4. The *Cassette for SET README* window indicates that the configuration of the Cassette for SET has successfully completed and allows you to display the README, if desired.

Note: The *IBM SET Cassette Configuration Type* screen enables you to specify whether to use default (Typical) or specific (Custom) configuration values for the given SET environment:

- Select **Typical** to display the *IBM SET Cassette Configuration Information*, which prompts you for the key database password. Enter the password to be used for the key database containing SET merchant certificates. Click **Next** to select the default SET values.
- Select **Custom** to display the *IBM SET Cassette Configuration Information* screen, which prompts you for the TCP Ports. Enter the port numbers chosen for the SET Payment TCP Port and the SET Inquiry TCP Port. Click **Next** to enter unique values, which include Certificate Database Type:

– Use the *Certificate Database Type* to designate whether your SET certificates will be stored in your database and accessed via ODBC (only if you are using DB2) or whether they will be stored in a flat file database:
  - If you choose flat file, you do not need a separate driver for the install. Choosing flat file may cause a degradation in performance as compared to ODBC.
  - If you choose ODBC, you must know the ODBC driver name for your database and the ODBC Version Number.

If you choose to store SET certificates in the database, the installation grants public access for select, insert, update, and delete operations to these tables: KEY, KEYPAIR, CRL, BCI, TABLEHEADER and DATABASEHEADER. In a production environment, **public** access to these tables may not be the appropriate level of security. If you change access to these tables, make sure the database administrator's user ID has the authority to perform the stated operations.

WebSphere Commerce Payments directory structure

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

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

```
<Payments_installdir>\eTillSETClasses.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 the Cassette for SET (iSeries)

This section describes the procedure for installing the Cassette for SET on 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 SET option of the WebSphere Commerce Payments for iSeries product.

- Specify the WebSphere Commerce Payments product number, option 1, for the Cassette for SET and the device from which the product is to be installed. For example,

  ```
  RSTLICPGM LICPGM(5733PY2) DEV(OPT01) OPTION(1).
  ```

Adding a Cassette for SET to a WebSphere Commerce Payments Instance

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

- The WebSphere Commerce Payments instance
- The HTTP Server that processes payment requests for WebSphere Commerce Payments
- The WebSphere application server that processes payment requests for WebSphere Commerce Payments

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

- Access the iSeries tasks web 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 **SET** from the cassette list and press the **Add** button to add the cassette.

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

  When the process for adding the cassette has completed:
  - The cassette-specific database tables will have been added to the WebSphere Commerce Payments instance database collection.
  - The password for the key database containing the SET merchant certificates is defaulted to the WebSphere Commerce Payments password currently configured for the WebSphere Commerce Payments instance.

    Note: The key database password can be changed from the WebSphere Commerce Payments user interface.
  - A unique TCP port number is configured for the SET Payment TCP port. Use the WebSphere Commerce Payments user interface to display information about the TCP port.

Configuring the cassette for socks

If your computer is behind a firewall and must use a socks server to access a certificate authority and Payment Gateway outside of your internal network, then specify a socks host name and socks port number to enable the Cassette for SET for socks. To do this, follow this procedure to specify the Java property configuration for the socks server after installing the WebSphere Commerce Payments Framework and the Cassette for SET:

1. Open the WebSphere Application Server administrative console.
2. Expand **Nodes**.

3. Expand the host name for the system where WebSphere Commerce Payments is installed.
4. Under Application Servers, select **WebSphere Commerce Payments**. If you are using an iSeries system, click **WPM** <*instance*> **WebSphere Commerce Payments**, where <*instance*> is the name of the WebSphere Commerce Payments instance.
5. On the **JVM Settings** tab page, click the **Advanced JVM Settings** button.
6. In the command line arguments field, add the following properties. If properties already exist in the field, you can add these to the end of the field (add a blank space before entering the new values).

   ```
   -Dwpmset.socksHost=hostname -Dwpmset.socksPort=portnumber
   ```

   hostname is the TCP host name or IP address for your socks server (and is required). *portnumber* is the TCP port number for your socks server. The port number property is optional; if not specified, the socks port number defaults to 1080.
7. Click **OK**, and then click **Apply** on the JVM Settings tab page to apply the configuration changes.
8. Stop and restart the WebSphere Commerce Payments Application Server in the WebSphere Application Server administrative console for your changes to take effect.

After following this procedure, WebSphere Commerce Payments should be able to use the socks server to access the desired SET certificate authority or Payment Gateway through the firewall.

Uninstalling Cassette for SET

To remove Cassette for SET, use the process for your operating system:

- “Uninstalling Cassette for SET on Windows NT or Windows 2000”
- “Uninstalling Cassette for SET on Solaris” on page 30
- “Uninstalling Cassette for SET on AIX” on page 30
- “Uninstalling Cassette for SET from iSeries” on page 31

If your certificates are stored in a flat file database, those files will not be removed when Cassette for SET is uninstalled. You can either manually delete the files after uninstalling or leave them available if you plan to reinstall WebSphere Commerce Payments. Certificates that are stored using ODBC are always removed when Cassette for SET is uninstalled.

Uninstalling Cassette for SET on Windows NT or Windows 2000

On Windows NT or Windows 2000, use these steps to remove the Cassette for SET:

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

6. Database files that were created after installation must be removed manually before a successful reinstallation of WebSphere Commerce Payments can be achieved.

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

Uninstalling Cassette for SET on Solaris

On Solaris, use these steps to remove the Cassette for SET:

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 Solaris console window. Be sure to be logged on as user root.
4. Enter `cd /var/sadm/pkg`
5. Enter `pkginfo itj*`
6. Record the **itj** numbers associated with the IBM WebSphere Commerce Payments Cassette for SET.
7. Enter `pkgrm itj <Cassette_for_SET_itjnumber1>`
8. Database files that were created after installation must be removed manually before a successful reinstallation of WebSphere Commerce Payments can be achieved.
9. Now remove the other package. Enter `pkgrm itj <Cassette_for_SET_itjnumber2>`

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

Uninstalling Cassette for SET on AIX

On AIX, use these steps to remove the Cassette for SET:

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. Click all components that begin with **IBM.WebSphere.Payment.Cassette.for.SET** and click **OK**.
6. Make sure the preview only field is set to **No**. Select **OK** to remove the Cassette for SET.
7. Database files that were created after installation must be removed manually before a successful reinstallation of WebSphere Commerce Payments can be achieved.

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

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

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

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

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

- Access the iSeries tasks web 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 **SET** from the cassette list and press the **Remove** button to remove the cassette.

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

Uninstalling Cassette for SET from iSeries

To uninstall the Cassette for SET option from the system, use the Delete License Program **(DLTLICPGM)** CL command. For example,

```
DLTLICPGM LICPGM(5733PY2) OPTION(1).
```

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

Chapter 5. Getting started

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

- Installed the WebSphere Commerce Payments Framework
- Created a WebSphere Commerce Payments instance (iSeries only)
- Installed the Cassette for SET
- Added the SET Cassette to the WebSphere Commerce Payments instance (iSeries only)
- Started WebSphere Application Server and the Web server
- Started WebSphere Commerce Payments
- Defined a WebSphere Commerce Payments user with administrative authority
- Created a merchant and Merchant administrator for that merchant

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

Note: Whenever a WebSphere Commerce Payments URL is listed in this manual, this implies that WebSphere Commerce Payments uses a default URL which corresponds to one using default HTTP port number 80. If your configuration is such that WebSphere Commerce Payments is running on a port other than default port number 80, you may need to specify a port number in the URL.

Cassette for SET tutorial

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

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

1. Create an account
2. Create a brand

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

Before starting this tutorial

There are a number of configuration steps that require information from your financial institution. In particular, you will need this information describing the acquirer and certificate authority (CA):

1. Acquirer information for each account:
   - Payment gateway hostname, port and uniform resource index (URI)
   - SET profile

- Signing brand

2. Certificate information for each brand:
   - Merchant ID
   - Brand ID
   - BIN
   - CA Request URL
   - CA root hash
   - Language used
   - Information to complete certificate registration form

Note: For educational purposes, IBM provides a test acquirer and CA that you can use with this tutorial. To use the IBM tutorial test infrastructure and CA, see Web site: www.ibm.com/software/webservers/paymgr/support/paydemo.html. You can complete the tutorial with either the test information or valid information obtained from your financial institution and acquirer.

Starting the WebSphere Commerce Payments user interface

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

To start the WebSphere Commerce Payments user interface:

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

Selecting a WebSphere Commerce Payments merchant and authorizing a cassette

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

1. From the navigation frame, click **Merchant Settings** under the Administration section.
2. From the Merchant Settings window, select the Test store merchant created during the Test cassette tutorial, or create a new merchant with merchant number 123456789.

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

3. On the next window, you will be permitted to authorize use of the Cassette for SET:

Authorized cassettes	Check the box next to *SET*. Checking this box authorizes the merchant to use the Cassette for SET.

4. When you have entered the requested information, click **Create Merchant** to save the merchant configuration.

5. You will also have to give the user ID Merchant Administrator authority for this merchant. For instructions on assigning roles, see the *WebSphere Commerce Payments Administrator's Guide*.

Logging in as the Merchant Administrator

To log off and log in again:

1. Click **Logoff admin** on the navigation frame of the WebSphere Commerce Payments user interface, and you will return to the main WebSphere Commerce Payments Login window.
2. From the main WebSphere Commerce Payments Login window, type the user ID (with Merchant Administrator authority) and the password and click **OK**.

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

Creating an Account

So far, you have granted the test store merchant permission to use the Cassette for SET. Your first task as the Merchant administrator is to create an *account* for the Cassette for SET.

An account is a relationship between the merchant and the financial institution that processes transactions for that merchant. Each payment cassette can have multiple accounts. For this tutorial, you will create one account for the Cassette for SET.

To create an account:

1. Click **Merchant Settings** on the navigation frame of the WebSphere Commerce Payments user interface.
2. From the Merchant Settings window, click the Cassette for SET icon in the Test Store window.
3. From the Cassette for SET window, click **Accounts**.
4. Click **Add an Account** on the Accounts window.
5. On the next window, enter the following information (for this tutorial, you *must* enter the italicized text exactly as shown):

Account name	Enter *SET Account*. This is the name that you assign to the account. This field is used only for display purposes in the user interface.
Account number	Enter *1*. This is a number that you (that is, either the hosting service provider or the Merchant administrator) assign which uniquely identifies the account in all transaction data. Used for tracking purposes.
Financial Institution name	Enter *SET Bank*. This is the name of the financial institution with which you hold this account. This field is used only for display purposes in the user interface.
Gateway hostname	Enter <payment_gateway_hostname>. This is the name of the payment gateway machine from the tutorial information from the IBM Web site (normally, this is supplied by your financial institution).

Gateway port	Enter <payment_gateway_number>. Port at the payment gateway to which requests should be sent. Get from the tutorial information from the IBM Web site (normally, this is supplied by your financial institution).
Gateway URI	Enter <payment_gateway_URI>. Receives merchant payment requests for your financial institution. Get this value from the tutorial information from the IBM Web site (usually, this is supplied by your financial institution).
Signing brand	Enter *ROBO*. This is the brand used to sign messages to the acquirer and is from the tutorial information from the IBM Web site (usually, this is supplied by your financial institution).
SET Profile	Enter <SET_profile_number>. This is from the tutorial information from the IBM Web site (normally, this is supplied by your acquirer).

6. Click **Create account** to create an account for the Cassette for SET.

Creating a Brand

Now that you have created the account for the Cassette for SET, you need to create a *Brand* for the account. A Brand is a credit card type, such as VISA or MasterCard. The Brand, or Brands, that you define for an account are based upon the terms of the account as defined by the financial institution. This tutorial uses the fictitious brand `ROBO`.

To create a brand:

1. Click **Merchant Settings** on the navigation frame of the WebSphere Commerce Payments user interface.
2. From the Merchant Settings window, click the Cassette for SET icon in the Test Store window.
3. From the Cassette for SET window, click **Accounts**.
4. Click **SET Account** on the Accounts window.
5. From the SET Account window, click **Brands**.
6. Click **Add a brand** on the Brands window.
7. On the next window, enter the brand information. For this tutorial, you **must** enter the *italicized* text exactly as shown:

Brand name	Enter *ROBO*. This is the local payment card brand for this account. This tutorial requires ROBO as the local brand name. Note that this identifier is used only as the brand identifier within WebSphere Commerce Payments. This value does not necessarily have to match the certificate brand ID.
Certificate ID	Enter <certificate_identifier>. The brand ID is specified in the certificate. You must enter the ID from the tutorial information from the IBM Web site (usually, this is supplied by your financial institution).
CA Request URL	Enter <certificate_authority_URL>. The Uniform Resource Locator of the Certificate Authority, from the tutorial information from the IBM Web site (usually, this is supplied by your financial institution).

Merchant ID	Enter <merchant_ID>. The merchant identifier from the tutorial information from the IBM Web site (usually, this is supplied by your financial institution). Enter a unique identifier for this merchant.
BIN	Enter *<bank_identification_number>*. The six-digit bank identification number, from the tutorial information from the IBM Web site (usually, this is supplied by your financial institution).
Language used	Enter <language>. The language used for certificate flow, provided in the tutorial information from the IBM Web site (usually, this is supplied by your financial institution).
Wallet purchases	Indicates if this brand should be presented to wallets. Check this box for this tutorial. Only one per cert brand ID.

8. Click **Create brand** to create a brand for the SET Account.
9. At this point, WebSphere Commerce Payments will start the certificate request process. If necessary, you will be asked to provide a root hash. The root hash is provided by your service provider, financial institution, or the tutorial information from the IBM Web site.
10. Next, you will be asked to fill out a registration form, the contents of which may vary. Fill out and submit the form to complete your end of the certificate request process and create the brand.

Managing payment processing

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

- Merchant-specific administration functions
- Payment processing functions (all)

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

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

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

For more information on these tasks, see the *WebSphere Commerce Payments Administrator's Guide*.

Creating orders using the Sample Checkout

As previously discussed, a real business environment features a customer who creates orders using a merchant's Internet storefront and a merchant who processes payments for those orders using WebSphere Commerce Payments. To walk through WebSphere Commerce Payments processing functions, you must create orders that require payment processing. To simulate a merchant's Internet

storefront and help you create orders, the Cassette for SET provides a Sample Checkout.To access the Sample Checkout and create orders:

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

Merchant number	Enter any number to represent a Merchant number.
Order number	Enter any number to represent an Order number.
Amount	Enter any amount to represent the total numeric amount of the order.
Currency	Enter *US dollar*. The currency used to place this order.
Payment method	Choose *SET* as the payment method.
Brand	Choose *ROBO* as the brand.
Credit card number	Enter *777777*. The ROBO credit card number.
Expiry Date	Highlight the expiration month and year for your ROBO credit card. **Note:** You can choose any month and year combination for this tutorial.
Card verification value	Enter the 3- or 4-digit verification code printed on the signature panel of the card.
Street address	Enter the street address of the location of the cardholder.
State or Province	Enter the name or abbreviation of the state or province of the location of the cardholder.
Cardholder's zip/postal code	Enter the postal code of the location of the cardholder.
Country	Select the country of the location of the cardholder.
Note: When the SET payment method is selected, additional fields are displayed to accept credit card and cardholder information (such as the cardholder address information commonly used in North America as AVS data, which is shown in this table). The Sample Checkout application can be modified to display other countries in the drop-down list for the Country field. (To do this, modify the SampleCheckoutSET.properties file for a given locale. More information about customizing properties files is provided in the *WebSphere Commerce Payments Administrator's Guide* .)	

3. Click **Buy**.
4. Repeat these steps twice (each time with a different Order number) so that you have three orders for which to process payments.

Approving orders

After creating three orders using the Sample Checkout, return to the browser window where the WebSphere Commerce Payments user interface is displayed. If you used the same browser window to access the Sample Checkout, you will need to point your browser once again to the WebSphere Commerce Payments URL (that is, **http://<hostname>/webapp/PaymentManager/**) and logon with Merchant Authority. To approve an order:

1. From the navigation frame, click **Approve** under the Payment Processing section.
2. From the Approve window, check the box next to the order you want to approve (select only one order for this exercise) and click **Approve Selected**.

3. On the Approve Results window, you will see the status of your approve request. When processing is complete, success or failure status will appear next to each order submitted for approval. When your approval is complete, click **Return to the Approve screen**.

Two orders still await your approval. You could have approved them all at once (for their full amount) by clicking **Approve All** from the Approve window. To demonstrate the many facets of the Approve function, you will work with each order individually.

Approving orders from the Order window

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

1. From the Approve window, click the **Order number** for one of the remaining orders awaiting approval.
2. From the Order window, you can view order details. Information unique to SET protocol (for example, PAN and expiration date) is shown along with data common to all cassettes.
3. Click **Approve** to approve this order.
4. From the Order Approve window, change the approval amount to `3.00` and click **Approve** to approve this order for three dollars.

When approval processing has completed, you will be returned to the Order window and notified of approval success or failure. You will notice in the order details that the approved amount has been updated to reflect the three dollars we specified in the previous step.

Using the Sale Function to approve orders

Because you approved only *part* of the last order you worked with, you still have two order entries in the Approve window. In this exercise, you will use the *sale* function to approve the remaining orders.

The sale function allows you to approve an order and move it directly into Deposited state, bypassing Approved state. The sale function automatically performs an Approve and a Deposit on your order payment (thus, you can think of sale as Approve with AutoDeposit). Perform a sale as follows:

1. From the navigation frame, click **Approve** under the Payment Processing section.
2. Click **Sale All** from the Approve window.
3. On the Approve Results window, you will see a progress bar indicating the status of your sale request. When processing is complete, success or failure status will appear next to each order submitted for sale.
4. When your sale is complete, click **Return to the Approve Screen**.

Depositing payments

Deposit allows you to deposit order payments. As demonstrated in “Approving orders from the Order window”, a single order number can have multiple payments associated with it. You may see the same order number appear multiple times in the same list, each time with different payment information. To deposit a payment:

1. From the navigation frame, click **Deposit** under the Payment Processing section.
2. Check the box next to one of the payments listed and click **Deposit Selected**.
3. When processing is complete, success or failure status will appear in the Deposit Results window next to the payment submitted for deposit.

4. When your sale is complete, click **Return to the Deposit Screen**.

Note: You can deposit only *part* of a payment, in much the same way you approved part of an order:

1. From the Deposit window, click the **Payment number** for one of the payments awaiting deposit.
2. The next window is the Payment window. From the Payment window, you can view payment details. Information unique to the SET protocol (for example, SET Authorization code and Gateway AuthTime) is shown along with data common to all cassettes.
3. Click **Deposit** to deposit this payment.
4. On the Order Payment window, change the deposit amount from `5.00` to `3.00` and click **Deposit** to deposit this payment for three dollars.

Settling batches

Whether or not a merchant is permitted to settle batches is determined by the Acquirer behavior in the SET profile. For ACB acquirers, the acquirer is responsible for closing batches, and the Settle option is not presented in the user interface. The account you created earlier uses a test acquirer which implements MCB behavior; so, for this tutorial, you will be able to settle the batch.

To view batches:

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

Merchant	The name of the merchant whose batch you are searching for. **Note:** If there are fewer than 500 merchants in the WebSphere Commerce Payments database, you will select the merchant name from the drop-down list. If there are more than 500 merchants in the WebSphere Commerce Payments database, you will enter the merchant number.
Batch Number	The number that uniquely identifies the batch within the merchant. Assigned when the payment is deposited.
State	The state of the batch: • Open • Closed For more information on batch states, see “Batch states” on page 76.
Balance Status	The balance status of this batch: • **Balanced :** the batch has been successfully balanced (that is, all totals agree). • **Out of balance:** an unsuccessful attempt has been made to balance this batch (that is, all totals do not agree).
Payment Type	Identifies the payment type, or protocol, used to place the order (SET, in this case).

Batch Open Date	Use the *after* and *before* fields below to search for batches opened during the specified range in time: • **After**: Specify a date to search for all batches opened on and after this date. • **Before**: Specify a date to search for all batches opened on and before this date.
Batch Closed Date	Use the *before* and *after* fields below to search for batches closed during the specified range in time: • **After**: Specify a date to search for all batches closed on and after this date. • **Before**: Specify a date to search for all batches closed on and before this date.

3. Click **Search** to initiate a batch search.

   Note: In addition to using the *after* and *before* fields to specify a time range for the batch search (such as, 08/01/99 to 08/15/99). These fields can also be used to narrow search results by *excluding* certain batches from the search. For example, you could search on all batches opened *before* 08/01/99 and all batches opened *after* 08/15/99, thus excluding batches opened between 08/02/99 and 08/14/99.
4. Click the batch number to view information about the batch.
5. From the Batch window, you can view useful batch information, including the total number and amount of both payments and credits in the batch. In addition to information common to all cassettes, information unique to the SET protocol (for example, SET Batch ID) is also shown.
6. Click **Batch Details** to see a detailed listing of all payments and credits in this batch.
7. Since the tutorial uses MCB acquirer, a **Settle** button will appear in the Settle Results window. Click **Settle**. When processing is complete, success or failure status will appear in the Settle Results window.

Issuing a credit

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

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

Merchant	The name of the merchant whose order you are searching for. **Note:** If there are fewer than 500 merchants in the WebSphere Commerce Payments database, you will select the merchant name from the drop-down list. If there are more than 500 merchants in the WebSphere Commerce Payments database, you will enter the merchant number.
Order Number	A number assigned by the merchant that uniquely identifies the order.

State	The state of the order: • Ordered • Refundable • Canceled • Closed For more information on order states, see “Order states” on page 70
Payment Type	Identifies the payment type, or protocol, used to place the order (SET, in this case).
Order Date	Use the *after* and *before* fields below to search for orders opened during the specified range in time: • **After**: Specify a date to search for all orders opened on and after this date. • **Before**: Specify a date to search for all orders opened on and before this date.
Order Amount	• **Currency**: The currency used to place this order. Select the currency type from the drop-down list. • **Greater than**: Specify a value to retrieve all orders with order amounts that are greater than or equal to the value you specify. • **Less than**: Specify a value to retrieve all orders with order amounts that are less than or equal to the value you specify.

3. Click **Search** to initiate an order search.

   Note: In addition to using the *after* and *before* fields to specify a time range for the order search (such as, 08/01/99 to 08/15/99). These fields can also be used to narrow search results by *excluding* certain orders from the search. For example, you could search on all orders opened *before* 08/01/99 and all orders opened *after* 08/15/99, thus excluding orders opened between 08/02/99 and 08/14/99.
4. From the next window, click an order number for an order in Refundable state (see “Order states” on page 70) to view the details of that order.
5. From the Order window, click **Credit** to create a credit against this order.
6. On the Create Credit window, the following information displays:

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

   Enter the credit amount (any amount up to the deposited amount of the order) and click **Credit**.

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

Viewing batch totals

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

1. From the navigation frame, click **Reports** under the Payment Processing section.
2. From the Reports window, click **Daily Batch Totals**.
3. At the Batch Totals Report window, you will be prompted to enter a date for which the totals are to be computed. *Leave this field blank to generate a report for the current date.*
4. You will also be prompted to enter the date for which you would like a Batch Totals Report. If you want a report for the current date, leave this field blank.
5. Click **Search** to generate the batch totals report.

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

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

Other administration tasks

Other Cassette for SET administration tasks are performed in the same manner as you would for any WebSphere Commerce Payments cassette. For more information, see the *WebSphere Commerce Payments Administrator's Guide*.

Chapter 6. Cassette for SET Cashier profiles

The Cashier is WebSphere Commerce Payments code 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 and ReceivePayment APIs. 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

Following are the Cassette for SET Cashier profiles:

Table 5. Cassette for SET Cashier Profiles

SET Cashier profile	Function
SampleCheckoutSET.profile	Uses MIA (AcceptPayment command) with optional AVS information
SampleCheckoutSETWallet.profile	Uses Wallet (ReceivePayment command) with optional AVS information

On iSeries, the Cassette for SET Cashier profiles are in the directory: `/QIBM/UserData/PymSvr/profiles/`.

Chapter 7. Using WebSphere Commerce Payments commands with Cassette for SET

Cassette for SET uses the standard WebSphere Commerce Payments commands. Information specific to the Cassette for SET's use of WebSphere Commerce Payments financial and administrative commands are described here. This information should be used in conjunction with the command descriptions in the *WebSphere Commerce Payments Programmer's Guide and Reference* .

Note: The Cassette for SET does not support the CassetteControl, CreateSystemCassetteObject, DeleteSystemCassetteObject, ModifySystemCassetteObject commands.

AcceptPayment

Note: In addition to the following parameters for ACCEPTPAYMENT, there are a number of parameters for the following:

- SaleDetail in Appendix B, "SaleDetail parameters" on page 93
- Purchasing cards in Appendix C, "Purchasing card parameters" on page 101

Required keywords for AcceptPayment command.

Keywords	Description	Value Type and Range
$BRAND	This field contains the card brand.	Value is specified in ASCII, 1 to 40 characters.
$EXPIRY	Specifies the card expiration date.	The value is specified as 6 numeric characters in the form `yyyymm`. Example value: 199911 (for November 1999).
$ORDERDESCRIPTION	Specifies details about an order.	Can be in binary format, 1 to 16,000 bytes.
$PAN	Specifies the PAN used in payment initiation message.	The value specified as 6 to 19 numeric characters.
PAYMENTTYPE	The name of the Cassette you are using to process the command.	Must be the ASCII character string **SET**.

Optional keywords for AcceptPayment command.

Keywords	Description	Value Type and Range
$AGENTNUM	A merchant terminal ID field that is optional but may be required by the Acquirer. Integer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.

Optional keywords for AcceptPayment command.

$AVS.COUNTRYCODE	Required for AVSData.	Value can be 1 to 999. (ISO-3166 country code).
$AVS.STREETADDRESS	Required for AVSData.	The value is specified as a string, 1 to 128 ASCII characters.
$AVS.STREETADDR	Alias for $AVS.STREETADDRESS	The value is specified as a string, 1 to 128 ASCII characters.
$AVS.CITY	Required for AVSData.	The value is specified as a string, 1 to 50 ASCII characters.
$AVS.STATEPROVINCE	Required for AVSData.	The value is specified as a string, 1 to 50 ASCII characters.
$AVS.STATEPROV	Alias for $AVS.STATEPROVINCE.	The value is specified as a string, 1 to 50 ASCII characters.
$AVS.POSTALCODE	Required for AVSData.	The value is specified as a string, 1 to 14 ASCII characters.
$AVS.LOCATIONID	Required for AVSData.	The value is specified as a string, 1 to 10 ASCII characters.
$CARDVERIFYCODE	Some payment cards are issued with a verification code. The verification code is generated by the issuing bank and can be verified by the bank. The account number followed by the three or four digit verification code is printed on the signature panel of the card.	The value must be a 3 or 4 character numeric string. Example values: 1234 or 321.
$CHAINNUM	An optional merchant terminal ID field that may be required by the Acquirer. Integer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$CHARSET	Indicates the content type and character set of the $ORDERDESCRIPTION parameter.	If a null value is specified, content type defaults to text/plain and the character set to U. A string, 1 to 1000 ASCII characters.
$MERCHCATCODE	One of two fields in the merchData structure; when specified, the cassette will use it.	Value must be a 4-character numeric string. Set this value if required by your Acquirer.
$MERCHGROUP	One of two fields in the merchData structure; when specified, the cassette will use it.	Integer. Value must be a numeric between ″1″ and ″8″. Semantics for possible values are described in the SET specification. Set this value if required by your Acquirer.

Optional keywords for AcceptPayment command.

$MERORDERNUM	The MerOrderNum field is in the SaleDetail structure of CapReq and AuthReq (with capture now) messages.	Value must be a character string, 1 to 24 ASCII characters.
$SPLITALLOWED	Indicates whether or not a merchant may approve additional payments.	Supported values are: 0-Indicates that this is the final payment for the order. 1-(Default) Indicates that the merchant may approve additional payments for the order. **Note:** Additional payments are allowed only if the SET profile specified in the Acquirer settings supports it.
$STORENUM	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer.	Integer. If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$TERMINALID	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer.	Integer. If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$TRANSMETHOD	Used to specify the value of the TransMethod field passed in the MIA (Merchant Initiated Authorization) extension.	Supported values are: • MIA_channelEncryption=0 • MIA_unencryptedWWW=1 • MIA_encryptedEMail=2 • MIA_unencryptedEMail=3 • MIA_otherElectronic=4(default) • MIA_mail=5 • MIA_telephone=6 • MIA_fax=7 • MIA_faceToFace=8 • MIA_0therNonElectronic=9
Sale detail keywords	For more information on these keywords, see Appendix B, “SaleDetail parameters” on page 93.	

Optional keywords for AcceptPayment command.

Purchasing card keywords	For more information on these keywords, see Appendix C, "Purchasing card parameters" on page 101.	

Approve

Optional keyword for Approve command.

Keywords	Description	Value Type and Range
$SPLITALLOWED	Indicates whether or not a merchant may approve additional payments.	Supported values are: 0-Indicates that this is the final payment for the order. 1-(Default) Indicates that the merchant may approve additional payments for the order. **Note:** Additional payments are allowed only if the SET profile specified in the Acquirer settings supports it.

ApproveReversal

Optional keyword for ApproveReversal command.

Keywords	Description	Value Type and Range
$SPLITALLOWED	Indicates whether or not a merchant may approve additional payments.	Supported values are: 0-Indicates that this is the final payment for the order. 1-(Default) Indicates that the merchant may approve additional payments for the order. **Note:** Additional payments are allowed only if the SET profile specified in the Acquirer settings supports it.

BatchOpen

Required keyword for BatchOpen command.

Keywords	Description	Value Type and Range
PAYMENTTYPE	The name of the Cassette you are using to process the command.	Must be the ASCII character string **SET**.

Optional keyword for BatchOpen command.

Keywords	Description	Value Type and Range
BATCHID	If present, the value used for the SET BATCHID in communication with the acquirer.	Integer. 0 to max integer.

CancelOrder

Cassettes are responsible for deleting ancillary objects. For the Cassette for SET, all related records in the SETOrder, SETPayment, SETCredit, BinaryData, and the SETMessages tables are deleted when CancelOrder command is issued with DeleteOrder = 1.

CloseOrder

Cassettes are responsible for deleting ancillary objects. For the Cassette for SET, all related records in the SETOrder, SETPayment, SETCredit, BinaryData, and the SETMessages tables are deleted when CloseOrder command is issued with DeleteOrder = 1.

CreateAccount

Required keywords for CreateAccount command

Keywords	Description	Value Type and Range
$ACQUIRESETPROFILE	SET profile.	Must be a positive integer. Get this value from your Acquirer.
$GATEWAYHOSTNAME	Hostname for Payment Gateway you are using.	ASCII string between 1 and 255 characters. Get this value from your Acquirer.
$GATEWAYPORT	If present, the value passed in will replace the GatewayPort field of the Account object.	Integer. The value must be a positive integer. Value is in range 1 to 65535.
$SIGNINGBRANDID	If present, the value passed in will replace the SigningBrandID field of the Account object.	ASCII string between 1 and 40 characters.

Optional keywords for CreateAccount command

Keywords	Description	Value Type and Range
$DELAYEDRETRYINTERVAL	If present, the value passed in will replace the DelayedRetryInterval field of the Account object.	Integer. Must be a nonnegative integer.

Optional keywords for CreateAccount command

$GATEWAYURI	If present, the value will replace the GatewayURI field of the Account object.	ASCII string that is either null or between 1 and 255 characters.
$MAXCONNECTIONS	If present, the value passed in will replace the MaxConnections field of the Account object.	Integer. Must be a positive integer.
$MAXDELAYEDRETRIES	If present, the value passed in will replace the MaxDelayedRetries field of the Account object.	Integer. Must be a positive integer.
$MAXIMMEDIATERETRIES	If present, the value passed in will replace the MaxImmediateRetries field of the Account object.	Integer. Must be a positive integer.
$READTIMEOUT	If present, the value passed in will replace the ReadTimeOut field of the Account object.	Integer. Must be a positive integer.

CreateMerchantCassetteObject

Use this command to create Brand objects. This command is only valid when the cassette is active.

Required keywords for CreateMerchantCassetteObject command

Keywords	Description	Value Type and Range
ACCOUNTNUMBER	Part of the Brand identifier-identifies the account for this brand. Required at brand creation time and then unchangable.	Integer and ASCII character string from 1–999999999.
CASSETTENAME	The name of the Cassette (SET)	ASCII character string from 1–64. (In this case, it must be ″SET″.)
MERCHANTNUMBER	Part of the Brand identifier-identifies the merchant for this account (and brand). Required at brand creation time and then unchangable.	WebSphere Commerce Payments merchant number from 1–999999999.
OBJECTNAME	The ASCII string ″Brand ″.	ASCII character string from 1–1000. (In this case, it must be ″Brand″.)

Required keywords for CreateMerchantCassetteObject command

$BIN	Bank Identification Number of the Merchant. This is one of the identifiers used in SET certificates. Required at brand creation time and then unchangable.	String of 6 numeric characters.
$BRANDID	Local identifier of the newly created brand object. Required at brand creation time and then unchangable.	String, 1 to 40 ASCII characters.
$CAURL	The URL of the Certificate Authority. Required at brand creation time and then unchangable.	String, 1 to 255 ASCII characters.
$LANGUAGE	Desired natural language for the certificate flow. Required at brand creation time and then unchangable.	String, 1 to 35 ASCII characters.
$MERCHANTID	Merchant ID assigned by Acquirer. Required at brand creation time and then unchangable. **Note:** A brand certificate's merchant ID is unique for each merchant. Do not create two merchants configured with the same brand and merchant ID. WebSphere Commerce Payments chooses a merchant certificate based on the brand and merchant ID.	This 1 to 30 ASCII character alphanumeric string is one of the identifiers used in SET certificates.

Optional keywords for CreateMerchantCassetteObject command

Keywords	Description	Value Type and Range
ENABLED	Configurable property that indicates whether or not the system should attempt to activate the object. On startup WebSphere Commerce Payments will try to initialize all objects that have their enabled property set to true. When the value of enabled is modified through the API, WebSphere Commerce Payments will attempt to start or stop the object for true and false values, respectively.	ASCII character string. Value will be either ″0″ or ″1″ where ″0″ is false and ″1″ is true.
$AGENTNUM	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$CERTBRAND	Brand identifier in SET certificates.	Defaults to value of $BRANDID.
$CHAINNUM	An optional MerTermID field that may be required by the Acquirer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$MAXIMMEDIATERETRIES	Number of times to retry a request/response flow.	Default is 0.

Optional keywords for CreateMerchantCassetteObject command

$PRESENTTOWALLETS	Indicates whether or not the brand is to be presented to wallets in the SET initiation message.	Boolean value. If true (1), this brand is to be presented to wallets in the SET initiation message. If false (0), the brand is not presented.
$READTIMEOUT	ReadTimeout for Certificate Authority TCP connection and the interval between retry attempts.	Defaults to 60.
$STORENUM	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$TERMINALID	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.

DeleteBatch

Cassettes are responsible for deleting ancillary objects. For the Cassette for SET, all related records in the SETBatch, BatchItem, SETBatchItem and SETMessages tables are deleted when DeleteBatch command is issued.

DeleteMerchantCassetteObject

Use this command to delete Brand objects. This command is only valid when the cassette is active.

Required keywords for DeleteMerchantCassetteObject command

Keywords	Description	Value Type and Range
ACCOUNTNUMBER	Part of the Brand identifier-identifies the account for this brand. Required at brand creation time and then unchangable.	Integer and ASCII character string from 1–999999999.
$BRANDID	BrandID of certificate requested (which is also the same thing as the BrandID for the SETBrandAdmin object.) Required at brand creation time and then unchangable.	String, 1 to 40 ASCII characters.
CASSETTENAME	The name of the Cassette (SET).	ASCII character string from 1–64. (In this case, it must be ″SET″.)
MERCHANTNUMBER	Part of the Brand identifier-identifies the merchant for this account (and brand). Required at brand creation time and then unchangable.	WebSphere Commerce Payments merchant number from 1–999999999.
OBJECTNAME	The ASCII string ″Brand ″.	ASCII character string from 1–1000. (In this case, it must be ″Brand″.)

Deposit

Note: In addition to the following parameters for DEPOSIT, there are a number of parameters purchasing cards in Appendix C, “Purchasing card parameters” on page 101.

Optional keyword for Deposit command

Keywords	Description	Value Type and Range
BATCHNUMBER	For merchant controls batch (MCB) acquirers, the merchant may set the batch number using this parameter.	ASCII string from 1–999999999.
Purchasing card keywords	For more information on these keywords, see Appendix C, “Purchasing card parameters” on page 101.	

ModifyAccount

Optional keywords for ModifyAccount command

Keywords	Description	Value Type and Range
$DELAYEDRETRYINTERVAL	If present, the value passed in will replace the DelayedRetryInterval field of the Account object.	Integer. Must be a positive integer.
$GATEWAYHOSTNAME	If present, the value will replace the GatewayHostname field of the Account object.	ASCII string between 1 and 255 characters.
$GATEWAYPORT	If present, the value passed in will replace the GatewayPort field of the Account object. Value is a range 1 to 65535.	Integer. The value or a positive integer.
$GATEWAYURI	If present, the value passed in will replace the GatewayURI field of the Account object.	ASCII string that is either null or between 1 and 255 characters.
$MAXCONNECTIONS	If present, the value passed in will replace the MaxConnections field of the Account object.	Integer. Must be a positive integer.
$MAXDELAYEDRETRIES	If present, the value passed in will replace the MaxDelayedRetries field of the Account object.	Integer. Must be a positive integer.
$MAXIMMEDIATERETRIES	If present, the value passed in will replace the MaxImmediateRetries field of the Account object.	Integer. Must be a positive integer.
$READTIMEOUT	If present, the value passed in will replace the ReadTimeOut field of the Account object.	Integer. Must be a positive integer.
$SIGNINGBRANDID	If present, the value passed in will replace the SigningBrandID field of the Account object.	ASCII string between 1 and 40 characters.

ModifyCassette

Optional keywords for ModifyCassette command

Keywords	Description	Value Type and Range
$CERTFLATFILEPATH	If present, the value passed in will replace the CertFlatfilePath field of the Cassette object.	ASCII string between 1 and 150 characters.
$CERTPASSWORD	If present, the value passed will modify the password for the certificate database.	ASCII string between 1 and 24 characters.
$HWCRYPTOID	If present, the value passed in will replace the HWCryptoID field of the Cassette object.	ASCII string that is either null or between and 1 and 50 characters.
$HWCRYPTOPWD	If present, the value passed in will replace the HWCryptoPwd field of the Cassette object. This write-only field is not exported.	ASCII string that is either null or between and 1 and 24 characters.
$READTIMEOUT	Identifies, in seconds, the read timeout to be used in socket communication with SET wallets.	Integer. Must be a positive integer.
$SETFLATFILEPATH	If present, the value passed in will replace the SETFlatfilePath field of the Cassette object.	ASCII string between 1 and 150 characters.
$WAKEUPMIMETYPE	If present, the value passed in will replace the WakeupMIMETYPE field of the Cassette object.	ASCII string that is either null or between 1 and 40 characters.

ModifyMerchantCassetteObject

Use this command to modify Brand objects. This command is only valid when the cassette is active.

Required keywords for ModifyMerchantCassetteObject command

Keywords	Description	Value Type and Range
ACCOUNTNUMBER	Part of the Brand identifier-identifies the account for this brand. Required at brand creation time and then unchangable.	Integer and ASCII character string from 1–999999999.

Required keywords for ModifyMerchantCassetteObject command

$BRANDID	BrandID of certificate requested (which is also the same thing as the BrandID for the SETBrandAdmin object.) Required at brand creation time and then unchangable.	String, 1 to 40 ASCII characters.
CASSETTENAME	The name of the Cassette (SET).	ASCII character string from 1–64. (In this case, it must be ″SET″.)
MERCHANTNUMBER	Part of the Brand identifier-identifies the merchant for this account (and brand). Required at brand creation time and then unchangable.	WebSphere Commerce Payments merchant number from 1–999999999.
OBJECTNAME	The ASCII string ″Brand ″.	ASCII character string from 1–1000. (In this case, it must be ″Brand″.)

Optional keywords for ModifyMerchantCassetteObject command

Keywords	**Description**	**Value Type and Range**
$ABORT	Used to indicate that the operation should be cancelled.	Boolean: 1=true, 2=false.
$ACCOUNTDATA	An additional registration form answer that is unique to the merchant as defined by the payment card brand and Acquirer.	String, 1 to 74 ASCII characters.
$AGENTNUM	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer. If this value is specified, then the value overrides what is stored in the Brand object.	If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$CHAINNUM	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$MAXIMMEDIATERETRIES	Number of times to retry a request/response flow.	Default is 0.

Optional keywords for ModifyMerchantCassetteObject command

$PRESENTTOWALLET	Indicates whether or not the brand is to be presented to wallets in the SET initiation message.	Boolean value. If true (1), this brand is to be presented to wallets in the SET initiation message. If false (0), the brand is not presented.
$READTIMEOUT	ReadTimeout for Certificate Authority TCP connection and the interval between retry attempts.	Defaults to 60.
$REGITEM.FIELDVALUE.n	Registration form answers. There can be up to 50 (REGITEM.FIELDVALUE.1-REGITEM.FIELDVALUE.50) of these specified. A REGITEM.FIELDVALUE is required when the CA indicates in the Registration form that a field is required.	The field values have a one-to-one correspondence to the Registration form, that is, REGITEM.FIELDVALUE.7 corresponds to the RegForm_field[06]. String, 1 to 128 unicode characters.
$RENEWCERTS	Used to indicate that this is a certificate renewal request.	Boolean: 1=true, 2=false.
$RETRIEVEGATEWAYCERTS	Used to retrieve the Gateway encryption certificate (through the PCertReq) for the brand, as specified by the $BRANDID parameter.	Boolean: 1=true, 2=false. When this parameter is set to true (1) and the Brand has valid merchant certificates, the Gateway certificate will be refreshed by sending a PCertReq. This can also be achieved by selecting the ″Retrieve Gateway Certificate″ button on the Brand screen. When the button is selected, the WebSphere Commerce Payments user interface sends a ModifyMerchantCassetteObject command with this flag set to true, which causes the PCert exchange to occur.
$ROOTHASH	A root hash will be passed into the cassette when a root hash is required as specified in the MeAqCInitRes message.	Must be 40 ASCII characters.
$STORENUM	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer. If this value is specified, then the value overrides what is stored in the Brand object.	If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.

Optional keywords for ModifyMerchantCassetteObject command

$TERMINALID	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.

ReceivePayment

Note: In addition to the following parameters for RECEIVEPAYMENT, there are a number of parameters for the following:

- SaleDetail in Appendix B, “SaleDetail parameters” on page 93
- Purchasing cards in Appendix C, “Purchasing card parameters” on page 101

Required keywords for ReceivePayment command.

Keywords	Description	Value Type and Range
$ORDERDESCRIPTION	Specifies details about an order.	Can be in binary format. 1 to 16000 bytes.
$CANCELURL	The URL displayed by the shopper’s browser when the shopper has cancelled a payment prior to completion.	This is a required field and must be 1 to 254 U.S. ASCII characters.
$FAILUREURL	The URL displayed by the shopper’s browser when payment processing has not completed successfully.	This is a required field and must be 1 to 254 U.S. ASCII characters.
$SERVICEURL	The URL displayed by the shopper’s browser for service purposes.	This is a required field and must be 1 to 254 U.S. ASCII characters.
$SUCCESSURL	The URL displayed by the shopper’s browser when payment processing is successfully completed.	This is a required field and must be 1 to 254 U.S. ASCII characters.

Note: When a consumer uses SET to make a purchase, the last message they receive from the Payment Server is the Payment Response (PRes). The PRes carries a CompletionCode with the status of the message. The Wallet will redirect the browser to either the success, failure, or cancel URL, depending on the value of this CompletionCode. The possible values of the CompletionCode, the meaning of each value, and the URL displayed are detailed in the table below.

CompletionCode	Meaning	URL
meaninglessRatio	Purchase amount is zero.	failure

CompletionCode	Meaning	URL
orderRejected	The AuthCode in the AuthRes was not equal to success, or the brand in the PReq did not match the brand in the PInitReq.	failure
orderReceived	Order received; no further processing has been completed.	success
orderNotReceived	Response to a PInq when the order is unknown.	No URL after inquiry
authorizationPerformed	Authorization has been successfully completed.	success
capturePerformed	Capture has been successfully completed.	success
creditPerformed	Credit has been successfully completed.	success

In most situations, the behavior of the wallet is easy to understand. However, one scenario that results in behavior that many people find confusing occurs when the merchant attempts to send the Authorization Request (AuthReq) to the gateway but receives no response. This could happen when there is a communication error or an error at the gateway which prevents it from sending the Authorization Response (AuthRes) back to the merchant.

In this case, WebSphere Commerce Payments does not know whether the authorization completed successfully at the gateway, was rejected by the gateway, or failed to be processed by the gateway. The only correct CompletionCode is OrderReceived. The CompletionCode cannot be set to OrderRejected or AuthorizationPerformed since no AuthRes has been received that would provide this information. The order has been successfully received and the Authorization may be sent again later using the automatic retry logic in WebSphere Commerce Payments.

The consumer will see the Success URL and must check back later if they want to see if the authorization went through. Neither the merchant nor the consumer can know the status of the authorization until an AuthRes has been successfully received. If the merchant chooses not to retry the AuthReq and consider the order rejected, they risk leaving an authorization in place on the consumer's account.

Optional keywords for ReceivePayment command.

Keywords	Description	Value Type and Range
$AGENTNUM	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer. If this value is specified, then the value overrides what is stored in the Brand object.	If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.

Optional keywords for ReceivePayment command.

$AVS.CITY	Required for AVSData.	Value is specified as a string with a maximum of 50 characters.
$AVS.COUNTRYCODE	Required for AVSData.	Value can be 1 to 999. (ISO-3166 country code).
$AVS.LOCATIONID	Required for AVSData.	The value is specified as a string with a maximum of 10 characters.
$AVS.POSTALCODE	Required for AVSData.	The value is specified as a string with a maximum of 14 characters.
$AVS.STATEPROVINCE	Required for AVSData.	The value is specified as a string with a maximum of 50 characters.
$AVS.STREETADDRESS	Required for AVSData.	Value is specified as a string with a maximum of 128 characters.
$CHAINNUM	An optional MerTermID field that may be required by the Acquirer. Set this value if required by your Acquirer. If this value is specified, then the value overrides what is stored in the Brand object.	If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$CHARSET	Indicates the content type and character set of the ORDERDESCRIPTION parameter.	If a null value is specified, content type defaults to text/plain and the character set to US ASCII.
$MERCHCATCODE	One of two fields in the merchData structure; when specified, the Cassette will use it.	Value must be a 4 character numeric string. **Note:** The requirement to use this field will come from the acquirer.
$MERORDERNUM	Is in the SaleDetail, structure of CapReq and AuthReq (with capture now) messages.	Value should be a character string with a maximum length of 24 characters. **Note:** The requirement to use this field will come from the acquirer.
$MERCHGROUP	One of two fields in the merchData structure; when specified, the Cassette will use it.	Value must be a numeric string with values between ″1″ and ″8″. Semantics for possible values are described in the SET specification. **Note:** The requirement to use this field will come from the acquirer.

Optional keywords for ReceivePayment command.

$REQUIRECARDCERT	Used to indicate that an incoming PReq from a Wallet must contain a cardholder certificate. If this keyword is omitted, purchases without certificates are allowed.	Valid values are: 0: Indicates that cardholder certificates are not required. 1: Indicates that cardholder certificate are required.
$SPLITALLOWED	Indicates whether or not a merchant may approve additional shipments.	Supported values are: 0-Indicates that this is the final payment for the order. 1-(Default) Indicates that the merchant may approve additional split payments for the order. **Note:** Additional payments are allowed only if the SET profile specified in the Acquirer settings supports it.
$STORENUM	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer. If this value is specified, then the value overrides what is stored in the Brand object.	If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
$TERMINALID	A MerTermID field that is optional to the SET message but may be required by the Acquirer. Set this value if required by your Acquirer.	If this value is specified, then the value overrides what is stored in the Brand object. If this value is *not* specified, then Brand values are used. This value will be associated with the order, thus it will be used on all subsequent SET messages to the Acquirer (for example, AuthReq, CapReq, etc . .) for the order.
Sale detail keywords	For more information on these keywords, see Appendix B, “SaleDetail parameters” on page 93.	
Purchasing card keywords	For more information on these keywords, see Appendix C, “Purchasing card parameters” on page 101.	

Refund

Optional keyword for Refund command

Keywords	Description	Value Type and Range
BATCHNUMBER	For merchant controls batch (MCB) acquirers, the merchant may set the batch number using this parameter.	ASCII string from 1–999999999.

Chapter 8. Using WebSphere Commerce Payments objects

This chapter describes Cassette for SET extensions to payment and administration objects. We present the syntax and description of all cassette-specific attributes and show examples of these attributes as viewed through WebSphere Commerce Payments query commands.

For a complete description of WebSphere Commerce Payments Framework objects, see the *WebSphere Commerce Payments Programmer's Guide and Reference* .

Financial objects used by Cassette for SET

The Cassette for SET extends the WebSphere Commerce Payments financial objects, Orders, Payments, Credits, and Batches. This section defines all attributes and states of these object extensions.

Order object

Table 6. WebSphere Commerce Payments Order Object Attributes Cassette for SET extensions

Field Name	Description
PAN	Cardholder's card number.
brandId	Cardholder's chosen payment card brand.
cardExpiry	Identifies the card expiration date.
cardVerifyCode	Cardholder's verification code. Found on the signature panel following the account number.
orderDescription	Summary of order contents.
contentTypeCharset	Identifies the content type and character set of the SET order description.
XID	Global unique identifier for the order.
BIN	Bank Identification Number from the cardholder's account number (first 6 digits).
Success URL Failure URL Cancel URL Service URL	Uniform resource locators presented to the cardholder. The success, failure and cancel URLs are presented upon completion of cardholder to merchant flows.

Table 6. WebSphere Commerce Payments Order Object Attributes Cassette for SET extensions (continued)

currentState	The state of the order: • Reset • Wakeup sent • PInitRes sent • Ordered • Refundable • Rejected • Pending • Canceled • Closed For more information on Order states, see "Order states" on page 70.
Address Verification Service (AVS)-related fields	
avsStreetAddress	Street address of the location of the cardholder.
avsCountryCode	Country code of the location of the cardholder.
avsCity	City name of the location of the cardholder.
avsStateProvince	Name or abbreviation of the state or province of the location of the cardholder.
avsPostalCode	Postal code of the location of the cardholder.
avsLocationID	An identifier used to specify a location.
Purchasing card-related fields	
shippingAmount	Total shipping/freight amount for the order.
dutyAmount	Total amount of duties or tariff for the order.
dutyReference	Reference number assigned to the duties or tariff for the order.
nationalTaxAmount	Total amount of national tax (sales or VAT) applied to the order.
localTaxAmount	Total amount of local tax applied to the order.
otherTaxAmount	Total amount of other taxes applied to the order.
totalTaxAmount	Total amount of all taxes applied to the order.
merchantTaxId	Tax identification number of the merchant.
merchantDutyTariffReference	Duty or tariff reference number assigned to the merchant.
customerDutyTariffReference	Duty or tariff reference number assigned to the cardholder.
summaryCommodityCode	Commodity code that applies to the entire order.
merchantType	Type of merchant.
merchantCountryCode	The ISO country code portion of the merchant's location.
merchantCityCode	City name portion of the merchant's location.
merchantStateProvince	Name or abbreviation of the state or province of the merchant's location.
merchantPostalCode	Postal code of the merchant's location.

Table 6. WebSphere Commerce Payments Order Object Attributes Cassette for SET extensions (continued)

merchantLocationId	Identifier that the merchant uses to specify one of its locations.
shipFromCountryCode	The ISO country code portion of the location where the goods are shipped from.
shipFromCityCode	City name portion of the location where the goods are shipped from.
shipFromStateProvince	Name or abbreviation of the state or province of the location where the goods are shipped from.
shipFromPostalCode	Postal code of the location where the goods are shipped from.
shipFromLocationId	An identifier that the merchant uses to specify one of its locations where the goods are shipped from.
shipToCountryCode	The ISO country code portion of the location where the goods are shipped to.
shipToCityCode	City name portion of the location where the goods are shipped to.
shipToStateProvince	Name or abbreviation of the state or province of the location where goods are shipped to.
shipToPostalCode	Postal code of the location where the goods are shipped to.
shipToLocationId	An identifier that the merchant uses to specify the location where the goods are shipped to.
merchantOrderNumber	Merchant order number.
customerReferenceNumber	Reference number assigned to the order by the cardholder.
orderSummary	Summary description of the order.
customerServicePhone	Merchant's customer service telephone number.
printCustomerServicePhoneNumber	Specifies if the issuer may print the merchant's customer service phone number on the cardholder's statement.
Purchasing card – line item data	
itemCommodityCode.*n*	Commodity code for the line item.
itemProductCode.*n*	Product code for the line item.
itemDescriptor.*n*	A description of the line item.
itemQuantity.*n*	The quantity for the line item.
itemUnitCost.*n*	Unit cost of the line item.
itemUnitOfMeasure.*n*	Unit of measure for the line item.
itemNetCost.*n*	Net cost per unit of the line item.
itemDiscountAmount.*n*	Amount of discount applied to the line item.
itemDiscountIndicator.*n*	Indicates if a discount was applied.
itemNationalTaxAmount.*n*	Amount of national tax (sales or VAT) applied to the line item.
itemNationalTaxRate.*n*	National tax (sales or VAT) rate applied to the line item.
itemNationalTaxType.*n*	Type of national tax applied to the line item.

Table 6. WebSphere Commerce Payments Order Object Attributes Cassette for SET extensions (continued)

itemLocalTaxAmount.*n*	Amount of local tax applied to the line item.
itemOtherTaxAmount.*n*	Amount of other taxes applied to the line item.
itemTotalCost.*n*	The total cost of the line item.

Order object XML example

WebSphere Commerce Payments provides a set of query commands that allow financial applications to search for and retrieve WebSphere Commerce Payments data. The query commands accept calls through an HTTP Post and return data in the form of an XML PSApiResult document. You can use the query commands to view the WebSphere Commerce Payments objects.

This XML example shows a Cassette for SET Order and its cassette extension object:

```
Order
    <PSOrder ID="O:123456789:36" amount="7000" amountExp10="-2" approvesAllowed="1"
           brand="VISA" currency="840" merchantAccount="111111111"
           merchantNumber="123456789" merchantOriginated="1" numberOfCredits="0"
           numberOfPayments="0" orderNumber="36" paymentType="SET" state="order_ordered"
           timeStampCreated="938116543000" timeStampModified="938116633000"
           unapprovedAmount="0">
      <CassetteExtensionObject>
        <CassetteProperty propertyId="PAN" value="2222222222222222">
        </CassetteProperty>
        <CassetteProperty propertyId="contentTypeCharset" value="text/plain">
        </CassetteProperty>
        <CassetteProperty propertyId="brandId" value="VISA">
        </CassetteProperty>
        <CassetteProperty propertyId="currentState" value="order_ordered">
        </CassetteProperty>
       <CassetteProperty propertyId="cardVerifyCode" value="4321">
        </CassetteProperty>
        <CassetteProperty propertyId="cardExpiry" value="199912">
        </CassetteProperty>
        <CassetteProperty propertyId="BIN" value="222222">
        </CassetteProperty>
      </CassetteExtensionObject>
      <CassetteObject key="orderDescription" objectId="orderDescription">
        <PackagedContent content="Cassette Object" transform="URLEncoded">
This+is+the+order        </PackagedContent>
      </CassetteObject>
    </PSOrder>
```

Order states

The state of an object determines what actions are *legal* for the object. The state of an object is determined by the action, or *command*, that was last performed on it (for example, a Payment that was approved, moves into Approved state). Orders are in one of these states:

State	Description
Reset	Initial order state.
Wakeup sent	A ReceivePayment command has been successfully processed and a SET initiation message has been returned to the cardholder wallet. The generic order will be in Requested state.

State	Description
PInitRes sent	The purchase initiation exchange (PInitReq/Res) has completed and WebSphere Commerce Payments is waiting for the purchase request for this order. The generic order will be in Requested state.
Ordered	The generic order will be in Ordered state.
Refundable	The generic order will be in Refundable state.
Pending	The generic order will be in Pending state.
Rejected	The generic order will be in Rejected state.
Cancelled	The generic order will be in Cancelled state.
Closed	The generic order will be in Closed state.

Payment object

The Payment object represents 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. The attributes for the Payment object are:

Table 7. WebSphere Commerce Payments Payment Object Attributes Cassette for SET extensions

Field Name	Description
approvalCode	Approval code assigned to the transaction by the Issuer.
authCode	Enumerated code indicating status of payment authorization processing. See SET Book 2 for valid values.
authTime	When the payment was authorized.
capCode	Enumerated code indicating status of capture. For information on valid values, see SET Book 2.
capTime	When the payment was captured.
cardCurrency	The ISO 4217 currency code of the cardholder.
currConvRate	The value with which to multiply the payment amount to calculate an amount in the cardholder's currency.
avsCode	Enumberated response code indicating the status of an Address Verification Service (AVS) request. For information on valid values, see SET Book 2.
authRetNum	ID of auth request used within the financial network. For more information, see SET Book 2.
authRevCode	Enumerated code for auth Reversals. For more information, see SET Book 2.
capRevCode	Enumerated code for auth capture. For more information, see SET Book 2.
Purchasing card-related fields	

Table 7. WebSphere Commerce Payments Payment Object Attributes Cassette for SET extensions (continued)

shippingAmount	Total shipping/freight amount for the order.
dutyAmount	Total amount of duties or tariff for the order.
dutyReference	Reference number assigned to the duties or tariff for the order.
nationalTaxAmount	Total amount of national tax (sales or VAT) applied to the order.
localTaxAmount	Total amount of local tax applied to the order.
otherTaxAmount	Total amount of other taxes applied to the order.
totalTaxAmount	Total amount of all taxes applied to the order.
merchantTaxId	Tax identification number of the merchant.
merchantDutyTariffReference	Duty or tariff reference number assigned to the merchant.
customerDutyTariffReference	Duty or tariff reference number assigned to the cardholder.
summaryCommodityCode	Commodity code that applies to the entire order.
merchantType	Type of merchant.
merchantCountryCode	The ISO country code portion of the merchant's location.
merchantCityCode	City name portion of the merchant's location.
merchantStateProvince	Name or abbreviation of the state or province of the merchant's location.
merchantPostalCode	Postal code of the merchant's location.
merchantLocationId	Identifier that the merchant uses to specify one of its locations.
shipFromCountryCode	The ISO country code portion of the location where the goods are shipped from.
shipFromCityCode	City name portion of the location where the goods are shipped from.
shipFromStateProvince	Name or abbreviation of the state or province of the location where the goods are shipped from.
shipFromPostalCode	Postal code of the location where the goods are shipped from.
shipFromLocationId	An identifier that the merchant uses to specify one of its locations where the goods are shipped from.
shipToCountryCode	The ISO country code portion of the location where the goods are shipped to.
shipToCityCode	City name portion of the location where the goods are shipped to.
shipToStateProvince	Name or abbreviation of the state or province of the location where goods are shipped to.
shipToPostalCode	Postal code of the location where the goods are shipped to.
shipToLocationId	An identifier that the merchant uses to specify the location where the goods are shipped to.
merchantOrderNumber	Merchant order number.

Table 7. WebSphere Commerce Payments Payment Object Attributes Cassette for SET extensions (continued)

customerReferenceNumber	Reference number assigned to the order by the cardholder.
orderSummary	Summary description of the order.
customerServicePhone	Merchant's customer service telephone number.
printCustomerServicePhoneNumber	Specifies if the issuer may print the merchant's customer service phone number on the cardholder's statement.
Purchasing card – line item data	
itemCommodityCode.*n*	Commodity code for the line item.
itemProductCode.*n*	Product code for the line item.
itemDescriptor.*n*	A description of the line item.
itemQuantity.*n*	The quantity for the line item.
itemUnitCost.*n*	Unit cost of the line item.
itemUnitOfMeasure.*n*	Unit of measure for the line item.
itemNetCost.*n*	Net cost per unit of the line item.
itemDiscountAmount.*n*	Amount of discount applied to the line item.
itemDiscountIndicator.*n*	Indicates if a discount was applied.
itemNationalTaxAmount.*n*	Amount of national tax (sales or VAT) applied to the line item.
itemNationalTaxRate.*n*	National tax (sales or VAT) rate applied to the line item.
itemNationalTaxType.*n*	Type of national tax applied to the line item.
itemLocalTaxAmount.*n*	Amount of local tax applied to the line item.
itemOtherTaxAmount.*n*	Amount of other taxes applied to the line item.
itemTotalCost.*n*	The total cost of the line item.

Payment object XML example

This XML example shows a Cassette for SET Payment and its cassette extension object:

```
Payment
        <PSPayment ID="P:123456789:36:1" amountExp10="-2" approveAmount="7000"
              batchNumber="1" currency="840" depositAmount="7000"
              merchantAccount="111111111" merchantNumber="123456789"
              orderNumber="36" paymentNumber="1" paymentType="SET"
              referenceNumber="203108572AA77D5D37EB65EF767382D8D5BE06F4"
              state="payment_closed" timeStampCreated="938116597000"
              timeStampModified="938117029000">
          <CassetteExtensionObject>
            <CassetteProperty propertyId="authCode" value="0">
            </CassetteProperty>
            <CassetteProperty propertyId="capCode" value="0">
            </CassetteProperty>
            <CassetteProperty propertyId="authTime" value="938116606000">
            </CassetteProperty>
            <CassetteProperty propertyId="cardCurrency" value="840">
            </CassetteProperty>
            <CassetteProperty propertyId="currConvRate" value="1.0">
            </CassetteProperty>
            <CassetteProperty propertyId="capTime" value="938116630000">
            </CassetteProperty>
```

```
    <CassetteProperty propertyId="SETPaymentState" value="payment_closed">
    </CassetteProperty>
  </CassetteExtensionObject>
</PSPayment>
```

Payment states

Payments are in one of these states:

State	Description
Authorized	The generic payment will be in Approved state.
Captured	The generic payment will be in Deposited state.
Closed	The generic payment will be in Closed state.
Declined	The generic payment will be in Declined state.
Reset	The generic payment will be in Void state.
AuthPending, AuthRevPending, CapPending, CapRevPending	Indicates an outstanding SET request message (AuthReq, AuthRevReq, CapReq, or CapRevReq) to the acquirer. The generic payment will be in Pending state.

Credit object

The WebSphere Commerce Payments command that creates the Credit object is called Refund. The Credit object identifies one credit made against the amount of money identified in one Order (that is, the payment agreement) object. There can be zero or more Credits per Order. The attributes for the Credit object are:

Table 8. WebSphere Commerce Payments Credit Object Attributes Cassette for SET extensions

Field Name	Description
credCode	Enumerated code indicating status of credit. See SET Book 2 for valid values.
credTime	When the credit was issued.
SETCreditState	The state of the Credit: • Reset • Credited • Closed • Declined • CredPending • CredRevPending For more information on Credit states, see “Credit states”.

Credit object XML example

This XML example shows a Cassette for SET Credit and its cassette extension object:

```
<PSCredit ID="C:123456789:36:1" amount="4000" amountExp10="-2" batchNumber="2"
      creditNumber="1" currency="840" merchantAccount="111111111"
      merchantNumber="123456789" orderNumber="36" paymentType="SET"
      referenceNumber="203108572AA77D5D37EB65EF767382D8D5BE06F4"
      state="credit_refunded" timeStampCreated="938117017000"
      timeStampModified="938117029000">
  <CassetteExtensionObject>
    <CassetteProperty propertyId="credCode" value="0">
    </CassetteProperty>
    <CassetteProperty propertyId="SETCreditState"
        value="credit_credited">
    </CassetteProperty>
    <CassetteProperty propertyId="credTime" value="938117027000">
    </CassetteProperty>
  </CassetteExtensionObject>
</PSCredit>
```

Credit states

Credits are in one of these states:

State	Description
Credited	The generic credit will be in Refunded state.
Closed	The generic credit will be in Closed state.
Declined	The generic credit will be in Declined state.
Reset	The generic credit will be in Void state.
CredPending, CredRevPending	Indicates an outstanding SET request message (CredReq or CredRevReq) to the acquirer. The generic credit will be in Pending state.

Batch object

A Batch is a collection of financial transactions (Payments and Credits) that are processed as a unit by a financial institution. A Batch is associated with an Account and a merchant. An Account can have zero or more Batches. The attributes for the Batch object are:

Table 9. WebSphere Commerce Payments Batch Object Attributes Cassette for SET extensions

Field Name	Description
batchID	The number that uniquely identifies the Batch.

Batch object XML example

This XML example shows a Cassette for SET Batch and its cassette extension object:

```
<PSBatch ID="B:123456789:2" batchNumber="2" batchStatus="batch_not_yet_balanced"
       merchantAccount="111111111" merchantControl="1" merchantNumber="123456789"
       paymentType="SET" purgeAllowed="1" state="batch_open" timeStampClosed="0"
       timeStampModified="938117030000" timeStampOpened="938117010000">
  <CassetteExtensionObject>
    <CassetteProperty propertyId="batchID" value="2">
    </CassetteProperty>
  </CassetteExtensionObject>
</PSBatch>
```

Batch states

The Cassette for SET does not define any cassette-specific states for Batches.

Administration objects used by Cassette for SET

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

- Cassette
- Account
- Brand (extension of Account)

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

Cassette object

The Cassette object describes the state of a cassette that is installed in WebSphere Commerce Payments. In addition to the generic Cassette object attributes described in the *WebSphere Commerce Payments Administrator's Guide*, Cassette for SET extensions to the Cassette object include:

Table 10. WebSphere Commerce Payments Cassette Object Attributes Cassette for SET extensions

Field Name	Description
paymentPort	TCP port used for wallet requests (PInitReq, PReq, InqReq).
wakeupMIMEType	MIME-type used for SET initiation messages.
hardwareCryptographyID	User ID for 4758 hardware cryptographic card (AIX only).
hwCryptoID	Password for 4758 hardware cryptographic card (AIX only).
certFlatFilePath	Path for flatfile (if ODBC not used).
certPassword	Password protecting the SET certificate database.
setFlatFilePath	Path used for SET encryption.
certDBType	Identifies database type (ODBC or flatfile).
odbcDriverName	Name of ODBC device driver.

Table 10. WebSphere Commerce Payments Cassette Object Attributes Cassette for SET extensions (continued)

Field Name	Description
odbcVersion	Identifies ODBC version.
readTimeout	Amount of time to wait for incoming requests from SET wallets.

Cassette object XML example

This XML example shows a Cassette for SET Cassette and its cassette extension objects:

```
<CassetteCollection>
        <PSCassette active="1" cassette="SET" changesPending="0" companyPkgName="ibm"
          enabled="1" traceSetting="-1" valid="1">
      <CassetteExtensionObject>
        <CassetteProperty propertyId="readTimeout" value="30">
        </CassetteProperty>
        <CassetteProperty displayType="hidden" propertyId="certFlatFilePath"
            value="c:\Program Files\IBM\Payments\data\">
        </CassetteProperty>
        <CassetteProperty propertyId="paymentPort" value="8620">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="certDBType" value="odbc">
        </CassetteProperty>
        <CassetteProperty propertyId="setFlatFilePath"
            value="c:\Program Files\IBM\Payments\data\">
        </CassetteProperty>
        <CassetteProperty propertyId="wakeupMIMEType" value="application/set-payment">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="odbcVersion" value="3">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="odbcDriverName"
              value="db2cli">
        </CassetteProperty>
      </CassetteExtensionObject>
    </PSCassette>
  </CassetteCollection>
```

Account object

The WebSphere Commerce Payments merchant Account object describes the state of an account that a merchant holds with a financial institution. Accounts are defined within a merchant/cassette relationship. SET account attribute are defined per-account (that is, there is one entry per account).

In addition to the generic Account object attributes described in the *WebSphere Commerce Payments Administrator's Guide*, Cassette for SET extensions to the Account object include:

Table 11. WebSphere Commerce Payments Account Object Attributes Cassette for SET extensions

Field Name	Visible in query result	Description
signingBrandId	Yes	Brand used to sign messages to the acquirer.
gatewayHostname	Yes	DNS name of payment gateway, obtained from acquirer.
gatewayPort	Yes	TCP port at payment gateway, obtained from acquirer.

Table 11. WebSphere Commerce Payments Account Object Attributes Cassette for SET extensions (continued)

Field Name	Visible in query result	Description
gatewayURI	Yes	Gateway uniform resource identifier.
maxConnections	Yes	Maximum number of simultaneous connections to acquirer. Not recommended. If used, obtained from acquirer. This field is not dynamic (that is, the Account must be restarted for a new value to take effect).
readTimeout	Yes	Either: • Read timeout for acquirer TCP connection • Interval between ″immediate″ attempts This field is not dynamic.
maxImmediateRetries	Yes	Number of ″immediate″ retry attempts. Not recommended. This field is not dynamic.
delayRetryInterval	Yes	Interval between ″delayed″ retry attempts. Not recommended. This field is not dynamic.
maxDelayedRetries	Yes	Number of ″delayed″ retry attempts. Not recommended. This field is not dynamic.
setProfile	Yes	Modifies the behavior of the SET cassette to communicating with a particular acquirer. Must be obtained from that acquirer.

Account object XML example

This XML example shows a Cassette for SET Account object:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult; objectCount="1" primaryRC="0" secondaryRC="0">
  <MerchantAccountCollection;>
    <PSMerchantAccount; active="1" cassette="SET" changesPending="0" enabled="1"
          financialInstName="My FI" merchantAccount="1"
          merchantAccountName="Test Account" merchantNumber="1" valid="1">
      <CassetteExtensionObject;>
        <CassetteProperty; propertyId="readTimeout" value="60">
        </CassetteProperty>
        <CassetteProperty; propertyId="maxConnections" value="10">
        </CassetteProperty>
        <CassetteProperty; propertyId="maxDelayedRetries" value="0">
        </CassetteProperty>
        <CassetteProperty; propertyId="delayRetryInterval" value="1">
        </CassetteProperty>
        <CassetteProperty; propertyId="maxImmediateRetries" value="0">
        </CassetteProperty>
        <CassetteProperty; propertyId="signingBrandId" value="VISA">
        </CassetteProperty>
        <CassetteProperty; propertyId="gatewayPort" value="10011">
        </CassetteProperty>
        <CassetteProperty; propertyId="gatewayHostname" value="etill03">
        </CassetteProperty>
        <CassetteProperty; propertyId="setProfile" value="2">
        </CassetteProperty>
        <CassetteProperty; propertyId="gatewayURI" value="http://etill03:10011">
        </CassetteProperty>
      </CassetteExtensionObject>
    </PSMerchantAccount>
                .
```

```
                .
                .
  </MerchantAccountCollection>
</PSApiResult>
```

Brand object

As described in “Administration objects” on page 9, the Cassette for SET defines Brand objects that encapsulate a SET certificate information. All Brand objects are associated with an Account objects; an Account has zero or more related Brand objects. Brand objects are MerchantCassetteObjects, as defined in the *WebSphere Commerce Payments Administrator's Guide*.

Brand objects can be created and manipulated using the CreateMerchantCassetteObject, ModifyMerchantCassetteObject, and DeleteMerchantCassetteObject API commands, where the OBJECTNAME parameter is set to the value ″Brand″.

Table 12. Brand Object Attributes defined by Cassette for SET

Field Name	Visible in query result	Description
brand	Yes	The object ID of the Merchant Cassette Object.
brandID	Yes	Brand identifier of the certificate Required when brand is created, thereafter unchangeable.
certificateBrandID	Yes	The certificate authority brand identifier. Required when brand is created, thereafter unchangeable.
bin	Yes	Bank identification number of the merchant. Required when brand is created, thereafter unchangeable.
acquirerBusinessID	Yes	Business identification number of the acquirer. Read-only, set during PCert flows.
merchantID	Yes	Merchant ID for this account and brand, assigned by Acquirer. Required when brand is created, thereafter unchangeable. **Note:** A brand certificate's merchant ID is unique for each merchant. Do not create two merchants configured with the same brand and merchant ID. WebSphere Commerce Payments chooses a merchant certificate based on the brand and merchant ID.
brandURL	Yes	URL where the shopper gets the logo GIF file for the brand. Returned in the Certificate Response message. Read-only, set during certificate flow.
caURL	Yes	The uniform resource locator of the Certificate Authority. Required when brand is created.
caReadtimeout	Yes	Read Timeout for Certificate Authority Configurable, but optional (default value is 60 seconds).
caMaxretries	Yes	Number of times to retry a request flow. Default is 0.
terminalID	Yes	Optional for SET but may be required by your Acquirer. Enter this value if required by your Acquirer.

Table 12. Brand Object Attributes defined by Cassette for SET (continued)

Field Name	Visible in query result	Description
chainnum	Yes	Optional for SET but may be required by your Acquirer. Enter this value if required by your Acquirer.
agentnum	Yes	Optional for SET but may be required by your Acquirer. Enter this value if required by your Acquirer.
storenum	Yes	Optional for SET but may be required by your Acquirer. Enter this value if required by your Acquirer.
haveGatewayCert	Yes	Indicates if this brand object has payment gateway certificates (obtained with the PCertReq).
haveMerchCerts	Yes	Indicates if this brand object has merchant certificates for signing messages and encrypting data.
language	Yes	The desired language to be used for the Certificate flow. Specified in the MeAqCInitReq message to the CA
roothash	Yes	Required in this situation: SDK receives MeAqCInitRes from CA. SDK tries to validate the signature certificate as follows: if signature certificate is a trusted root, then all is good; if signature certificate is not a trusted root, then look in other certificates sent in the message and, if not found there, in the CMS certificate database for the certificate's root issuer. If root issuer is found, check that the root is trusted. If not trusted, or root issuer not found, then RootHashRequired is returned. User will be responsible for providing trusted root hash. This usually occurs when doing private label (not using or paying for SETCo trusted root) or SETCo root becomes compromised.
accountNumber	Yes	Part of the Brand identifier - identifies the account for this brand. Required at creation time and then unchangeable.
accountData	Yes	For a merchant, this field is unique to that merchant as defined by the payment card brand and acquirer. This data is collected with the registration form.

Table 12. Brand Object Attributes defined by Cassette for SET (continued)

Field Name	Visible in query result	Description
certReqState	Yes	Current state in the Certificate Finite State Machine (FSM). The certificate States are: • Stable - There are no outstanding certificate request information for this Brand object. • Init - The CertInit flows have been successfully passed for a certificate request. • RootHashRequired - A MeAqCInitRes has been successfully received, but the Root Hash is needed to process the MeAqCInitRes. • CertsPending - The certificate request has been processed by the CA, but the merchant must try back later to get the certificates. • InitFailure - The CertInit flow has failed.

certStatusCode	Yes	Enumerated code in the CertRes indicates the status of the Certificate Request. Values are defined in ibmset.h and include:
		`requestComplete = 1`
		`invalidLanguage = 2`
		`invalidBIN = 3`
		`sigValidationFail = 4`
		`decryptionError = 5`
		`requestInProgress = 6`
		`rejectedByIssuer = 7`
		`requestPended = 8`
		`rejectedByAquirer = 9`
		`regFormAnswerMalformed = 10`
		`rejectedByCA = 11`
		`unableToEncryptResponse = 12`

policyText	Yes	Statement to be displayed (along with the Registration Form) to the user. The text is in the language specified in the MeAqCInitReq.
cardLogoURL	Yes	URL where the shopper gets the logo GIF file for the credit card. Returned in the Certificate Response message. Information-only property - set during Certificate flows.
rejectReason	Yes	Optional reason for registration form rejection (returned in the MeAqCInitRes).
eeMessage	Yes	Optional message from the CA in natural language to be displayed to the user (sent by the CA in an unsuccessful CertRes).
brandEnabled	Yes	Configurable property that indicates whether or not the system should attempt to activate the object. On startup, the Payment Server will try to initialize all objects that have their enabled property set to true. When the value of enabled is modified through the API, Payment Server will attempt to start or stop the object for true and false values, respectively.

Table 12. Brand Object Attributes defined by Cassette for SET (continued)

Field Name	Visible in query result	Description
brandActive	Yes	Information property that indicates whether the object is currently active. Note that setting the enabled property to false will stop an object and set the active property to false. Setting enabled to true will result in an attempt to start the object; if the startup is successful, then the active property will be set to true. If an object ″fails″ for whatever reason, the active property of that object should be set to false.
brandValid	Yes	Indicates whether or not the object has problems which may prevent it from starting. Set to true when there is either a known problem with configuration or there is a problem bringing up the object.
brandPending	Yes	Indicates whether or not the configuration of an object as viewed through the Query API precisely matches the current configuration of that object. This is used in conjunction with properties whose values can only be changed when the object is not active. If a user attempts to change such a property while the object is active, the value will be changed and pending will be set to true. The next time the object is brought up, the new value will take effect and pending will be reset.
brandMessages	Yes	Message information associated with this object. This information can be displayed by the Administration UI.
expiry	Yes	Expiration date of the certificate.
requestable	Yes	Allows user to continue certificate request.
updatable	Yes	If true, brand attributes (for example, StoreNum) can be updated.
abortable	Yes	Passed to the cassette in Protocol Data to indicate if the user would like to end a request to obtain certificates.
renewable	Yes	Passed to the cassette in Protocol Data to indicate if the user would like to renew certificates for a brand that already has certificates, which may be about to expire or have already expired.
inquirable	Yes	Indicates certificate is pending and a CA inquiry can be issued.
referralURL	Yes	Optional collection of URLs pointing to referral information, listed in the order of relevance (returned in the MeAqCInitRes). Sent only when registration form cannot be returned.
regform	Yes	The Registration Form returned from the certificate authority in the MeAqCInitRes.
regField	Yes	The answers to the Registration Form as supplied by the user.

Table 12. Brand Object Attributes defined by Cassette for SET (continued)

Field Name	Visible in query result	Description
certState	Yes	Because the enc and sig certificates are obtained in pairs, this represents the earliest expiration date of the 2 certificates (that is, if the sig cert expires before the enc cert, this will be the expiration date of the sig cert.)
presentToWallets	Yes	If true, present this brand to wallets in the SET initiation message. If false, this brand is not presented.

Brand states

The most important property of the Brand object is the state of the certificate request (CertReqState). During Brand creation, there are several steps that are performed to obtain SET certificates. The CertReqState property indicates the progress of the certificate request process for this brand.

Certificate requests are in one of the states shown in Table 13. Note that the state of a certificate request is unrelated to whether or not valid certificates have been obtained for a Brand. In other words, this is the state of the certificate request, rather than an indicator of valid certificates.

Table 13. Brand states

State	Description
Stable	No outstanding certificate request exist for this brand object. Brand objects are initially created in Stable state and return to Stable state at the completion of all certificate message flows. In this state, the ModifyMerchantCassetteObject can be used to request or renew certificates.
Root Hash Required	Indicates a MeAqCInitRes has been successfully received from the CA but it cannot be processed because it is not known within the current trust hierarchy. In this state, the merchant must issue a ModifyMerchantCassetteObject command that includes the valid root hash for the certificate.
Initialized	Indicates the certificate initiation exchange has been successful and the registration form is now visible to the merchant through Query API commands. The merchant must submit a ModifyMerchantCassetteObject request that includes answers to all relevant registration form questions.
Initialization failure	The certificate initialization exchange has failed due to connectivity problems with the CA. The merchant can submit a ModifyMerchantCassetteObject command with a corrected CA URL.
Certificates pending	The CA has received a valid certificate request but will not deliver certificates to the merchant immediately. The merchant must issue a ModifyMerchantCassetteObject command to request the certificates later.

When obtaining certificates, successful scenarios include:

- If the certificate is known within the trust hierarchy, Brand creation (CreateMerchantCassetteObject) places the Brand in the Init state. The

subsequent ModifyMerchantCassetteObject command results in certificate delivery and returns the Brand object to the Stable state.

- If the certificate is outside the current trust hierachy, an additional step is required to supply the root hash. The state of the request moves from RootHashRequired to Init to Stable in the successful case.

Brand object XML examples

Let's look at the HTTP requests and XML responses exchanged when creating a Brand and obtaining certificates for that Brand. The example here shows the API commands that you can use to create a Brand and obtain certificates. This example shows these commands:

- CreateMerchantCassetteObject to create the brand in Init state
- QueryAccounts to view the Brand and registration form questions
- ModifyMerchantCassetteObject to complete the registration form questions, which results in obtaining certificates and moving the Brand object to Stable state
- QueryAccounts to view the Brand with its certificates

Initial creation: This shows creation of a Brand. The HTTP post request has been generated by a CreateMerchantCassetteObject request for the SET Brand object:

```
POST /webapp/PaymentManager/PaymentServlet HTTP/1.1
Connection: Keep-Alive
Accept: application/xml
Authorization: Basic YWRtaW46YWRtaW4=
Host: localhost
User-Agent: Java PaymentServerClient
Content-Encoding: 8859_1
Content-Length: 306
Content-Type: application/x-www-form-urlencoded

OPERATION=CREATEMERCHANTCASSETTEOBJECT&ETAPIVERSION=3&
MERCHANTNUMBER=123456789&ACCOUNTNUMBER=111111111&%24LANGUAGE=en&
%24MERCHANTID=112233&ENABLED=1&%24BIN=120002&OBJECTNAME=BRAND&
%24BRANDID=MASTERCARD&%24CAURL=http%3A%2F%2Fecomsetca%3A5065%2FMASTERCARDMCA&
CASSETTENAME=SET&%24ACQUIRERBUSINESSID=1122334455
```

This response shown indicates that the registration form is available (that is, the request is successful):

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="0" primaryRC="0" secondaryRC="1357">
</PSApiResult>
```

View of Brand object using QueryAccount command (Response only): The Brand object shown in this XML response (with some data omitted for brevity) is the result of a QueryAccount command:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
  <MerchantAccountCollection>
    <PSMerchantAccount active="1" cassette="SET" ... deleted ... >
      <CassetteExtensionObject>
        ... deleted ..
      </CassetteExtensionObject>
      <CassetteConfigObject active="0" changesPending="0" enabled="1"
            key="MASTERCARD" objectId="brand" valid="0">
        <CassetteProperty displayType="readOnly" propertyId="bin" value="120002">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="acquirerBusinessID"
              value="1122334455">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="caURL"
              value="http://ecomsetca:5065/MASTERCARDMCA">
```

```
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="brandURL"
      value="http://www.kkpower21/success.jpg">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="merchantID"
      value="112233">
</CassetteProperty>
<CassetteProperty displayLength="3" displayMode="basic"
      displayName="Average Employee Age" displayType="readWrite"
      propertyId="regField06" required="0" value="">
</CassetteProperty>
<CassetteProperty displayLength="12" displayMode="basic"
      displayName="Merchant Auth Flag (y/n)" displayType="readWrite"
      propertyId="regField05" required="0" value="">
</CassetteProperty>

<CassetteProperty displayLength="25" displayMode="basic"
      displayName="Merchant Country" displayType="readWrite"
      propertyId="regField04" required="1" value="">
</CassetteProperty>
<CassetteProperty displayLength="10" displayMode="basic"
      displayName="Merchant PostalCode" displayType="readWrite"
      propertyId="regField03" required="1" value="">
</CassetteProperty>
<CassetteProperty displayLength="25" displayMode="basic"
      displayName="Merchant State" displayType="readWrite"
      propertyId="regField02" required="1" value="">
</CassetteProperty>
<CassetteProperty displayLength="25" displayMode="basic"
      displayName="Merchant City" displayType="readWrite"
      propertyId="regField01" required="1" value="">
</CassetteProperty>
<CassetteProperty displayLength="25" displayMode="basic"
      displayName="Merchant Name" displayType="readWrite"
      propertyId="regField00" required="1" value="">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="caMaxretries"
      value="0">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="certStatusCode"
      value="-1">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="haveMerchCerts"
      value="0">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="haveGatewayCert"
      value="0">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="brandID"
      value="MASTERCARD">
</CassetteProperty>
<CassetteProperty displayMode="basic" displayType="readOnly"
      propertyId="abortable" value="1">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="certReqState"
      value="certs_init">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="caReadtimeout"
      value="60">
</CassetteProperty>
<CassetteProperty displayMode="basic" displayType="readWrite"
      propertyId="regform" value="">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="policyText"
      value="No policy text">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="cardLogoURL"
```

```
            value="http://www.kkpower21/success.jpg">
          </CassetteProperty>
          <CassetteProperty displayMode="basic" displayType="readOnly"
            propertyId="requestable" value="1">
          </CassetteProperty>
          <CassetteProperty displayLength="128" displayMode="basic"
            displayName="Financial Instituion 125" displayType="readWrite"
            propertyId="accountData" required="1" value="">
          </CassetteProperty>
          <CassetteProperty displayType="readOnly" propertyId="language" value="en">
          </CassetteProperty>
          <CassetteProperty displayType="readOnly" propertyId="presentToWallets"
            value="1">
          </CassetteProperty>
          <CassetteProperty displayType="readOnly" propertyId="certBrandID"
            value="MASTERCARD">
          </CassetteProperty>
        </CassetteConfigObject>
      </PSMerchantAccount>
    </MerchantAccountCollection>
</PSApiResult>
```

Submit answers to registration form questions using ModifyMerchantCassetteObject: This ModifyMerchantCassetteObject command includes answers to all registration form questions:

```
POST /webapp/PaymentManager/PaymentServlet HTTP/1.1
Connection: Keep-Alive
Accept: application/xml
Authorization: Basic YWRtaW46YWRtaW4=
Host: localhost
User-Agent: Java PaymentServerClient
Content-Encoding: 8859_1
Content-Length: 350
Content-Type: application/x-www-form-urlencoded


OPERATION=MODIFYMERCHANTCASSETTEOBJECT&ETAPIVERSION=3&
MERCHANTNUMBER=123456789&ACCOUNTNUMBER=111111111&
%24ACCOUNTDATA=Test+Bank+Account&OBJECTNAME=BRAND&
%24REGITEM.FIELDVALUE.5=USA&%24REGITEM.FIELDVALUE.4=27709&
%24REGITEM.FIELDVALUE.3=NC&%24REGITEM.FIELDVALUE.2=Raleigh&
%24REGITEM.FIELDVALUE.1=SET+Test+Merchant&%24BRANDID=MASTERCARD&
CASSETTENAME=SET
```

This XML response indicates that certificates have been successfully delivered and the Brand has moved to Stable state:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="0" primaryRC="0" secondaryRC="0">
</PSApiResult>
```

View of Brand object using QueryAccount command (Response only): This XML document (the result of a QueryAccount command) displays the Brand with its certificates:

```
<?xml version="1.0" encoding="UTF-8"?>
<PSApiResult objectCount="1" primaryRC="0" secondaryRC="0">
  <MerchantAccountCollection>
    <PSMerchantAccount active="1" cassette="SET" ... deleted ... >
      <CassetteExtensionObject>
        ... deleted ...
      </CassetteExtensionObject>
      <CassetteConfigObject active="1" changesPending="0" enabled="1"
            key="MASTERCARD" objectId="brand" valid="1">
        <CassetteProperty displayType="readOnly" propertyId="bin" value="120002">
        </CassetteProperty>
```

```
<CassetteProperty displayType="readOnly" propertyId="acquirerBusinessID"
      value="1122334455">
</CassetteProperty>
<CassetteProperty displayMode="advanced" displayType="readOnly"
      propertyId="updatable" value="1">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="expiry"
      value="1988164800000">
</CassetteProperty>
<CassetteProperty displayType="readWrite" propertyId="caURL"
      value="http://ecomsetca:5065/MASTERCARDMCA">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="brandURL"
      value="http://www.kkpower21/success.jpg">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="merchantID"
      value="112233">
</CassetteProperty>
<CassetteProperty displayLength="3" displayMode="advanced"
      displayName="Average Employee Age"
      displayType="readOnly" propertyId="regField06" value="">
</CassetteProperty>
<CassetteProperty displayLength="12" displayMode="advanced"
      displayName="Merchant Auth Flag (y/n)" displayType="readOnly"
      propertyId="regField05" value="">
</CassetteProperty>
<CassetteProperty displayLength="25" displayMode="advanced"
      displayName="Merchant Country" displayType="readOnly"
      propertyId="regField04" value="USA">
</CassetteProperty>
<CassetteProperty displayMode="basic" displayType="readOnly"
      propertyId="renewable" value="1">
</CassetteProperty>
<CassetteProperty displayLength="10" displayMode="advanced"
      displayName="Merchant PostalCode" displayType="readOnly"
      propertyId="regField03" value="27709">
</CassetteProperty>
<CassetteProperty displayLength="25" displayMode="advanced"
      displayName="Merchant State" displayType="readOnly"
      propertyId="regField02" value="NC">
</CassetteProperty>
<CassetteProperty displayLength="25" displayMode="advanced"
      displayName="Merchant City" displayType="readOnly"
      propertyId="regField01" value="Raleigh">
</CassetteProperty>
<CassetteProperty displayLength="25" displayMode="advanced"
      displayName="Merchant Name" displayType="readOnly"
      propertyId="regField00" value="SET Test Merchant">
</CassetteProperty>
<CassetteProperty propertyId="caMaxretries" value="0">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="certStatusCode"
      value="1">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="haveMerchCerts"
      value="1">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="haveGatewayCert"
      value="1">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="brandID"
      value="MASTERCARD">
</CassetteProperty>
<CassetteProperty displayType="readOnly" propertyId="certReqState"
      value="certs_stable">
</CassetteProperty>
<CassetteProperty propertyId="caReadtimeout" value="60">
```

```
        </CassetteProperty>
        <CassetteProperty displayMode="advanced" displayType="readOnly"
              propertyId="regform" value="">
        </CassetteProperty>
        <CassetteProperty displayMode="advanced" displayType="readOnly"
              propertyId="walletGroup" value="1">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="policyText"
              value="No policy text">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="cardLogoURL"
              value="http://www.kkpower21/success.jpg">
        </CassetteProperty>
        <CassetteProperty displayMode="basic" displayType="readOnly"
              propertyId="certState" value="">
        </CassetteProperty>
        <CassetteProperty displayLength="128" displayMode="advanced"
              displayName="Financial Instituion 125" displayType="readOnly"
              propertyId="accountData" value="Test Bank Account">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="language" value="en">
        </CassetteProperty>
        <CassetteProperty displayType="readWrite" propertyId="presentToWallets"
              value="1">
        </CassetteProperty>
        <CassetteProperty displayType="readOnly" propertyId="certBrandID"
              value="MASTERCARD">
        </CassetteProperty>
      </CassetteConfigObject>
    </PSMerchantAccount>
  </MerchantAccountCollection>
</PSApiResult>
```

Address Verification Service (AVS) result codes

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

The cardholder's billing address, including street address and zip code, are sent in the electronic authorization request message to the issuer. The issuer compares the street address and zip code to those it has on file and returns an AVS response code to advise you of the comparison status. This information enables decision making that limits risks when shipping merchandise, and risk reduction for the financial institution can result in reduced transaction fees for the merchant. Therefore, the SET protocol allows the use of this tool by its merchants. It is up to the merchant to decide what risks are allowable if AVS data does not compare favorably.

When Address Verification Services (AVS) are requested on an AcceptPayment or a ReceivePayment command, subsequent approvals will reflect the results of the AVS check by storing the associated AVS result code in the Payment object. Since other credit card-oriented cassettes also support AVS, but may use different result codes, the WebSphere Commerce Payments Framework provides a set of common AVS result codes that can be used by any cassette that supports AVS. These common codes relieve merchant software from having to be aware of which cassette is being used. For more information on Address Verification Services, see "SaleDetail and AVSData" on page 18.

Note: If you are interested in additional information regarding AVS and merchant chargeback liabilities, contact your acquiring financial institution.

The following table illustrates the way the Cassette for SET maps the SET-specific AVS result codes to the WebSphere Commerce Payments Framework's common AVS codes.

Table 14. WebSphere Commerce Payments common AVS result codes mapped to Cassette for SET AVS result codes

Cassette for SET AVS result code (returned from acquirer)	**WebSphere Commerce Payments Common AVS result code**	**Explanations**
4	0	Both the postal code (that is, the AVS 5–digit and 9–digit) and the street address match.
2	1	The street address matches, but the postal code does not match.
3	2	The 5–digit or 9–digit postal codes matches, but the street address does not match.
1	3	Neither the street address nor the postal code matches.
0	4	This constant maps: • Address information unavailable • System unavailable • Card type not supported • Transaction ineligible for Address Verification Services

Appendix A. Certificate utility

A certificate utility program, **CertUtil** has been created to allow the user to view SET certificates that are stored in the SET certificate database (which can either be a flatfile stored on disk, or an ODBC database). It replaces the **eTillCertReq** program that was shipped with Payment Server 1.2, with one exception: it does not support the certificate request functions. Obtaining certificates from a Certificate Authority must now be done from within WebSphere Commerce Payments, as described in the “Cassette for SET tutorial” on page 33.

CertUtil properties

The **CertUtil** program has an associated properties file, called `CertUtil.properties`. Use of this file allows the user to specify information about the SET certificate database (flatfile or ODBC, certificate database password, location of database, etc...). Use of this file is optional. If you do not update `CertUtil.properties`, then you will be prompted to enter the values needed to view the SET certificate database.

To customize the `CertUtil.properties` file, edit the file and uncomment the appropriate lines and enter appropriate values:

- If SET certificate database is a flatfile, uncomment and provide values for these lines:
  - keyDatabaseType=flatfile
  - keyDatabasePath=directory where your SET certificate database exists
  - tracePath (optional...only if you want to turn tracing on)
  - keyDatabasePassword=your SET certificate database cryptographic password
- If SET certificate database is ODBC, uncomment these lines:
  - keyDatabaseType=odbc
  - tracePath (optional...only if you want to turn tracing on)
  - keyDatabaseUserID=userID of the WebSphere Commerce Payments database
  - keyDatabaseDataSource=WebSphere Commerce Payments database name
  - keyLogicalDBName=WebSphere Commerce Payments database name (should always be the same as keyDatabaseDataSource)
  - keyDatabaseDataSourcePW=password of the user of the WebSphere Commerce Payments database
  - keyDatabaseTableOwner=userID of the WebSphere Commerce Payments database owner
  - odbcDrvrMgrName=name of the ODBC driver manager (e.g. db2, intersolv)
  - odbcVersion=version number of the ODBC driver manager
  - keyDatabasePassword=your SET certificate database cryptographic password

Starting the CertUtil program

To start the CertUtil program, enter **CertUtil** at the command prompt in the directory in which WebSphere Commerce Payments was installed.

Running the CertUtil program

After the CertUtil program has obtained all necessary data (either from the CertUtil.properties file or from prompting the user), the main menu displays:

```
Select option
   1  - List all Merchant Certificates
   2  - List all Gateway Certificates
   3  - List all Certificates
   4  - List all Pending Certificate Requests
   5  - Show comprehensive information on a certificate request by record id
   6  - Show comprehensive information on a certificate by record id
   7  - Show Certificate Expiration Dates
   8  - Dump Certificate Database To File
   9  - Delete Pending Certificate Requests
  10  - Delete Certificate
  11  - Delete All Certificates
   q  - Quit

Where:

List all Merchant Certificates
     Lists all of the certificates in the database that have an entity type of
     Merchant.

List all Gateway Certificates
     Lists all of the certificates in the database that have an entity type of
     Payment Gateway.

List all Certificates
     Lists all of the certificates in the database.

List all Pending Certificate Requests
     Lists all of the pending certificate requests in the database. Pending
     Certificate Requests are created when a Certificate Authority is unable to
     issue certificates right away to the end entity. As a result, the
     certificate request is saved in the database so that the end entity can
     try to get the certificates at a later date.

Show comprehensive information on a certificate request by record ID
     Shows detailed information about a certificate request. The user will be
     prompted to enter a record ID to indicate which certificate request they want
     to look at.

Show comprehensive information on a certificate by record ID
     Shows detailed information about a certificate. The user will be prompted to
     enter a record ID to indicate which certificate they want to look at.

Show Certificate Expiration Dates
     Lists all of the certificates in the database, and shows the expiration date
     of each. The expiration date that is listed is the public key expiration date
     and is in the format YY.MM.DD.

Dump Certificate Database To File
     Dumps the entire certificate database (which includes: certificates,
     certificate requests, BCI, and CRL) to a file on disk.

Delete Pending Certificate Requests
     Deletes the specified certificate request from the SET Certificate database.
     The user is prompted to enter the record ID of the certificate request to be
     deleted. NOTE: Do not delete certificate requests while
     WebSphere Commerce Payments is running.
Delete Certificate
     Deletes the specified certificate from the SET Certificate database. The
     user is prompted to enter the record ID of the certificate to be deleted.
     NOTE: Do not delete certificates while WebSphere Commerce Payments
     is running.

Delete All Certificates
     Deletes all certificates from the SET Certificate database. This is a very
     dangerous function, so use it cautiously.
     NOTE: Do not delete certificates while WebSphere Commerce Payments is running.
```

Appendix B. SaleDetail parameters

The following table lists the SaleDetail parameters:

SaleDetail	Description
$SD.PAYRECURIND	Enumerated transaction type
$SD.AUTHCHARIND	Enumerated value that indicates the conditions present when the authorization was performed
$SD.ORDERSUMMARY	Description of the order
$SD.CUSTOMERREFERENCENUMBER	Reference number assigned to the order by the Cardholder
$SD.CUSTOMERSERVICEPHONE	Merchant's customer service telephone number
$SD.OKTOPRINTPHONEIND	Boolean value indicating if the Issuer may print the customer service telephone number on the Cardholder's statement
CommercialCardData	
$SD.CCARD.MERCHANTLOCATION.COUNTRYCODE	ISO 3166 country code for the location
$SD.CCARD.MERCHANTLOCATION.CITY	City name of the location
$SD.CCARD.MERCHANTLOCATION.STATEPROVINCE	Name of abbreviation of the state or province
$SD.CCARD.MERCHANTLOCATION.POSTALCODE	Postal code of the location
$SD.CCARD.MERCHANTLOCATION.LOCATIONID	Identifier used by the Merchant to identify one of its locations
$SD.CCARD.SHIPFROM.COUNTRYCODE	ISO 3166 country code for the location
$SD,CCARD.SHIPFROM.CITY	City name of the location
$SD.CCARD.SHIPFROM.STATEPROVINCE	Name or abbreviation of the state or province
$SD.CCARD.SHIPFROM.POSTALCODE	Postal code of the location
$SD.CCARD.SHIPFROM.LOCATIONID	Identifier used by the Merchant to identify one of its locations
$SD.CCARD.SHIPTO.COUNTRYCODE	ISO 3166 country code for the location
$SD.CCARD.SHIPTO.CITY	City name of the location

$SD.CCARD.SHIPTO.STATEPROVINCE	Name or abbreviation of the state or province
$SD.CCARD.SHIPTO.POSTALCODE	Postal code of the location
$SD.CCARD.SHIPTO.LOCATIONID	Identifier used by the Merchant to identify one of its locations
$SD.CCARD.CHARGEINFO.TOTALFREIGHTSHIPPINGAMOUNT	Total amount added to the order for shipping and handling
$SD.CCARD.CHARGEINFO.TOTALDUTYTARIFFAMOUNT	Total amount of duties or tariff for the order
$SD.CCARD.CHARGEINFO.DUTYTARIFFREFERENCE	Reference number assigned to the duties or tariff for the order
$SD.CCARD.CHARGEINFO.TOTALNATIONALTAXAMOUNT	Total amount of national tax (sales or VAT) applied to the order
$SD.CCARD.CHARGEINFO.TOTALLOCALTAXAMOUNT	Total amount of national tax (sales or VAT) applied to the order
$SD.CCARD.CHARGEINFO.TOTALOTHERTAXAMOUNT	Total amount of other taxes applied to the order
$SD.CCARD.CHARGEINFO.TOTALTAXAMOUNT	Total amount of taxes applied to the order
$SD.CCARD.CHARGEINFO.MERCHANTTAXID	Tax identification number of the Merchant
$SD.CCARD.CHARGEINFO.MERCHANTDUTYTARIFFREF	Duty or tariff reference number assigned to the Merchant
$SD.CCARD.CHARGEINFO.CUSTOMERDUTYTARIFFREF	Duty or tariff reference number assigned to the Cardholder
$SD.CCARD.CHARGEINFO.SUMMARYCOMMODITYCODE	Commodity code that applies to the entire order
$SD.CCARD.CHARGEINFO.MERCHANTTYPE	Type of merchant
$SD.CCARD.ITEM.QUANTITY	Number of items
$SD.CCARD.ITEM.UNITOFMEASURECODE	Unit of measure for the item
$SD.CCARD.ITEM.DESCRIPTOR	Item description
$SD.CCARD.ITEM.COMMODITYCODE	Item commodity code
$SD.CCARD.ITEM.PRODUCTCODE	Item product code
$SD.CCARD.ITEM.UNITCOST	Item cost
$SD.CCARD.ITEM.NETCOST	Net cost of item
$SD.CCARD.ITEM.DISCOUNTIND	Indicates if discount applies
$SD.CCARD.ITEM.DISCOUNTAMOUNT	Amount of discount applied to item

$SD.CCARD.ITEM.NATIONALTAXAMOUNT	Amount of national tax (sales or VAT) applied to the item
$SD.CCARD.ITEM.NATIONALTAXRATE	National tax (sales or VAT) rate applied to the item
$SD.CCARD.ITEM.NATIONALTAXTYPE	Type of national tax applied to the item
$SD.CCARD.ITEM.LOCALTAXAMOUNT	How much local tax was applied to the item
$SD.CCARD.ITEM.OTHERTAXAMOUNT	How much other tax was applied to the item
$SD.CCARD.ITEM.ITEMTOTALCOST	Total cost of item
MarketSpec Data	Code that identifies the type of market-specific data supplied on the authorization (as determined by the financial network)
$SD.MARKETSPECDATAID	Code that identifies the type of market-specific data supplied on the authorization (determined by the financial network)
$SD.AUTO.RENTERNAME	Name of person renting the vehicle
$SD.AUTO.RENTALLOCATION.COUNTRYCODE	ISO 3166 country code for the location
$SD.AUTO.RENTALLOCATION.CITY	City name of the location
$SD.AUTO.RENTALLOCATION.STATEPROVINCE	Name or abbreviation of the state or province
$SD.AUTO.RENTALLOCATION.POSTALCODE	Postal code of the location
$SD.AUTO.RENTALLOCATION.LOCATIONID	Identifier used by the Merchant to identify one of its locations
$SD.AUTO.RENTALDATETIME	Date (and optionally time) the vehicle was rented
$SD.AUTO.AUTONOSHOW	Code indicating that the customer failed to show up to rent the vehicle as scheduled
$SD.AUTO.RENTALAGREEMENTNUMBER	Rental agreement number
$SD.AUTO.REFERENCENUMBER	Rental reference number
$SD.AUTO.INSURANCETYPE	Type of insurance selected by the renter
$SD.AUTO.RATEINFO.DAILYRENTALRATE	Daily rental rate
$SD.AUTO.RATEINFO.WEEKLYRENTALRATE	Weekly rental rate
$SD.AUTO.RATEINFO.LATERETURNHOURLYRATE	Hourly charge for late returns

$SD.AUTO.RATEINFO.DISTANCERATE	Rate charged per mile in excess of any free distance allowance
$SD.AUTO.RATEINFO.FREEDISTANCE.SCALE	Unit of measure (kilometers, miles) for distance
$SD.AUTO.RATEINFO.FREEDISTANCE.DIST	Distance the vehicle can travel per day without incurring an additional charge
$SD.AUTO.RATEINFO.VEHICLECLASSCODE	Class of vehicle rented
$SD.AUTO.RATEINFO.CORPORATEID	Corporate identification number that applies to the rental rate
$SD.AUTO.RETURNLOCATION.COUNTRYCODE	ISO 3166 country code for the location
$SD.AUTO.RETURNLOCATION.CITY	City name of the location
$SD.AUTO.RETURNLOCATION.STATEPROVINCE	Name or abbreviation of the state or province
$SD.AUTO.RETURNLOCATION.POSTALCODE	Postal code of the location
$SD.AUTO.RETURNLOCATION.LOCATIONID	Identifier used by the Merchant to identify one of its locations
$SD.AUTO.RETURNDATETIME	Date (and optionally time) the vehicle was returned
$SD.AUTO.CHARGES.REGULARDISTANCE	Amount of charges for the rental (excluding extras classified below)
$SD.AUTO.CHARGES.LATERETURN	Amount of charges for returning the vehicle after the date and time due back
$SD.AUTO.CHARGES.TOTALDISTANCE.SCALE	Unit of measure (kilometers, miles) for distance
$SD.AUTO.CHARGES.TOTALDISTANCE.DIST	Total distance the vehicle was driven
$SD.AUTO.CHARGES.EXTRADISTANCE	Amount of the charges resulting from exceeding the free distance allowance
$SD.AUTO.CHARGES.INSURANCE	Amount of charges resulting from insurance
$SD.AUTO.CHARGES.FUEL	Amount of refueling charges
$SD.AUTO.CHARGES.AUTOTOWING	Amount of charges resulting from towing

$SD.AUTO.CHARGES.ONEWAYDROPOFF	Amount of the drop-off charges resulting from a one-way rental
$SD.AUTO.CHARGES.TELEPHONE	Amount of charges resulting from the use of the rental vehicle telephone
$SD.AUTO.CHARGES.VIOLATIONS	Amount of charges resulting from violations assessed during the rental period
$SD.AUTO.CHARGES.DELIVERY	Amount of charges resulting from the delivery of the rental vehicle
$SD.AUTO.CHARGES.PARKING	Amount of charges resulting from parking the rental vehicle
$SD.AUTO.CHARGES.OTHER	Amount of other charges not classified elsewhere
$SD.AUTO.CHARGES.TOTALTAXAMOUNT	Total amount of taxes applied to the rental
$SD.AUTO.CHARGES.AUDITADJUSTMENT	Amount the transaction was adjusted as a result of auditing by the rental company
$SD.HOTEL.ARRIVALDATE	When the Cardholder checked in (or was scheduled to check in) to the hotel
$SD.HOTEL.HOTELNOSHOW	Enumerated code indicating that the customer failed to check in to the hotel as scheduled
$SD.HOTEL.DEPARTUREDATE	Date the Cardholder checked out of the hotel
$SD.HOTEL.DURATIONOFSTAY	Number of days the Cardholder stayed in the hotel
$SD.HOTEL.FOLIONUMBER	Folio number
$SD.HOTEL.PROPERTYPHONE	Telephone number of the hotel
$SD.HOTEL.CUSTOMERSERVICEPHONE	Customer service telephone number (of the hotel or the hotel chain)
$SD.HOTEL.PROGRAMCODE	Indicates type of special program that applies to the stay
$SD.HOTEL.RATEINFO.DAILYROOMRATE	Daily room rate, including applicable taxes unless the DailyTaxRate is specified

$SD.HOTEL.RATEINFO.DAILYTAXRATE	Amount of taxes applied to the daily room rate
$SD.HOTEL.CHARGES.ROOM	Total amount charged for the room (excluding extras classified below)
$SD.HOTEL.CHARGES.ROOMTAX	Amount of tax applied to the RoomCharges. charges
$SD.HOTEL.CHARGES.PREPAIDEXPENSES	Total amount of pre-paid expenses
$SD.HOTEL.CHARGES.FOODBEVERAGE	Total amount of food and beverage charges
$SD.HOTEL.CHARGES.ROOMSERVICE	Total amount of room service charges
$SD.HOTEL.CHARGES.MINIBAR	Total amount of mini bar charges
$SD.HOTEL.CHARGES.LAUNDRY	Total amount of laundry
$SD.HOTEL.CHARGES.TELEPHONE	Total amount of telephone charges
$SD.HOTEL.CHARGES.BUSINESSCENTER	Total amount of business center charges
$SD.HOTEL.CHARGES.PARKING	Total amount of parking charges
$SD.HOTEL.CHARGES.MOVIE	Total amount of in-room movie charges
$SD.HOTEL.CHARGES.HEALTHCLUB	Total amount of health club charges
$SD.HOTEL.CHARGES.GIFTSHOPPURCHASES	Total amount of gift shop purchase charges
$SD.HOTEL.CHARGES.FOLIOCASHADVANCES	Total amount of cash advances applied to the room
$SD.HOTEL.CHARGES.OTHER	Total amount of other charges (not classified above)
$SD.HOTEL.CHARGES.TOTALTAXAMOUNT	Total amount of taxes applied to the bill
$SD.HOTEL.CHARGES.AUDITADJUSTMENT	Amount the transaction was adjusted as a result of auditing by the hotel
$SD.TRANSPORT.PASSENGERNAME	Name of the passenger to whom the tickets were issued
$SD.TRANSPORT.DEPARTUREDATE	Departure date
$SD.TRANSPORT.ORIGCITYAIRPORT	City of origin for the trip

$SD.TRANSPORT.TRIPLEG.DATEOFTRAVEL	Date of travel for this trip leg
$SD.TRANSPORT.TRIPLEG.CARRIERCODE	Carrier code for this trip leg
$SD.TRANSPORT.TRIPLEG.SERVICECLASS	Class of service for this trip leg
$SD.TRANSPORT.TRIPLEG.STOPOVERCODE	Indicates whether stopovers are permitted for this trip leg
$SD.TRANSPORT.TRIPLEG.DESTCITYAIRPORT	Destination city for this trip leg
$SD.TRANSPORT.TRIPLEG.FAREBASISCODE	Fare basis code for this trip leg
$SD.TRANSPORT.TRIPLEG.DEPARTURETAX	Departure tax for this trip leg
$SD.TRANSPORT.TICKETNUMBER	Ticket number
$SD.TRANSPORT.TRAVELAGENCYCODE	Travel agency code
$SD.TRANSPORT.TRAVELAGENCYNAME	Travel agency name
$SD.TRANSPORT.RESTRICTIONS	Code Indicating restrictions on refunds or changes

Appendix C. Purchasing card parameters

The following table lists the parameters used for purchasing cards. For a description of purchasing cards, see “Purchasing cards” on page 18.

Table 15. Purchasing card parameters

Purchasing Card parameter	Required (R) or Optional (O)	Value	Description
$PCARD.SHIPPINGAMOUNT	O	0 - MAX INT (ASCII)	Total shipping/freight amount for the order.
$PCARD.DUTYAMOUNT	O	0 - MAX INT (ASCII)	Total amount of duties or tariff for the order.
$PCARD.DUTYREFERENCE	O	1 to 28 characters (UTF8)	Reference number assigned to the duties or tariff for the order.
$PCARD.NATIONALTAXAMOUNT	O	0 - MAX INT (ASCII)	Total amount of national tax (sales or VAT) applied to the order.
$PCARD.LOCALTAXAMOUNT	O	0 - MAX INT (ASCII)	Total amount of local tax applied to the order.
$PCARD.OTHERTAXAMOUNT	O	0 - MAX INT (ASCII)	Total amount of other taxes applied to the order.
$PCARD.TOTALTAXAMOUNT	O	0 - MAX INT (ASCII)	Total amount of all taxes applied to the order.
$PCARD.MERCHANTTAXID	O	1 to 10 characters (UTF8)	Tax identification number of the merchant.
$PCARD.MERCHANTDUTYTARIFFREFERENCE	O	1 to 28 characters (UTF8)	Duty or tariff reference number assigned to the merchant.
$PCARD.CUSTOMERDUTYTARIFFREFERENCE	O	1 to 28 characters (UTF8)	Duty or tariff reference number assigned to the cardholder.
$PCARD.SUMMARYCOMMODITYCODE	O	1 to 15 characters (UTF8)	Commodity code that applies to the entire order.
$PCARD.MERCHANTTYPE	O	1 to 4 characters (UTF8)	Type of merchant.
$PCARD.MERCHANTCOUNTRYCODE	R*	1 to 999 integer (ASCII)	*Required only if any other merchant location information is present. The ISO country code portion of the merchant’s location.
$PCARD.MERCHANTCITYCODE	O	1 to 50 characters (UTF8)	City name portion of the merchant’s location.
$PCARD.MERCHANTSTATEPROVINCE	O	1 to 50 characters (UTF8)	Name or abbreviation of the state or province of the merchant’s location.
$PCARD.MERCHANTPOSTALCODE	O	1 to 14 characters (UTF8)	Postal code of the merchant’s location.
$PCARD.MERCHANTLOCATIONID	O	1 to 10 characters (UTF8)	Identifier that the merchant uses to specify one of its locations.
$PCARD.SHIPFROMCOUNTRYCODE	R*	1 to 999 integer (ASCII)	*Required only if any other ShipFrom information is present. The ISO country code portion of the location where the goods are shipped from.
$PCARD.SHIPFROMCITYCODE	O	1 to 50 characters (UTF8)	City name portion of the location where the goods are shipped from.
$PCARD.SHIPFROMSTATEPROVINCE	O	1 to 50 characters (UTF8)	Name or abbreviation of the state or province of the location where the goods are shipped from.
$PCARD.SHIPFROMPOSTALCODE	O	1 to 14 characters (UTF8)	Postal code of the location where the goods are shipped from.
$PCARD.SHIPFROMLOCATIONID	O	1 to 10 characters (UTF8)	An identifier that the merchant uses to specify one of its locations where the goods are shipped from.
$PCARD.SHIPTOCOUNTRYCODE	R*	1 to 999 integer (ASCII)	*Required only if any other ShipTo information is present. The ISO country code portion of the location where the goods are shipped to.

Table 15. Purchasing card parameters (continued)

Purchasing Card parameter	Required (R) or Optional (O)	Value	Description
$PCARD.SHIPTOCITYCODE	O	1 to 50 characters (UTF8)	City name portion of the location where the goods are shipped to.
$PCARD.SHIPTOSTATEPROVINCE	O	1 to 50 characters (UTF8)	Name or abbreviation of the state or province of the location where goods are shipped to.
$PCARD.SHIPTOPOSTALCODE	O	1 to 14 charactes (UTF8)	Postal code of the location where the goods are shipped to.
$PCARD.SHIPTOLOCATIONID	O	1 to 10 characters (UTF8)	An identifier that the merchant uses to specify the location where the goods are shipped to.
$PCARD.MERCHANTORDERNUMBER	O	1 to 25 characters (ASCII)	Merchant order number.
$PCARD.CUSTOMERREFERENCENUMBER	O	1 to 28 characters (UTF8)	Reference number assigned to the order by the cardholder.
$PCARD.ORDERSUMMARY	O	1 to 35 characters (UTF8)	Summary description of the order.
$PCARD.CUSTOMERSERVICEPHONE	O	1 to 20 characters (UTF8)	Merchant's customer service telephone number.
$PCARD.PRINTCUSTOMERSERVICEPHONENUMBER	O	Boolean value (default is False)	Specifies if the issuer may print the merchant's customer service phone number on the cardholder's statement.
Line item data			
$ITEM.COMMODITYCODE	O	1 to 15 characters (UTF8)	Commodity code for the line item.
$ITEM.PRODUCTCODE	O	1 to 12 characters (UTF8)	Product code for the line item.
$ITEM.DESCRIPTOR	R*	1 to 35 characters (UTF8)	*Required only if any other line item information is present. A description of the line item.
$ITEM.QUANTITY	R*	0 - MAX INT (ASCII)	*Required only if any other line item information is present. The quantity for the line item.
$ITEM.UNITCOST	O	0 - MAX INT (ASCII)	Unit cost of the line item.
$ITEM.UNITOFMEASURE	O	1 to 12 characters (UTF8)	Unit of measure for the line item.
$ITEM.NETCOST	O	0 - MAX INT (ASCII)	Net cost per unit of the line item.
$ITEM.DISCOUNTAMOUNT	O	0 - MAX INT (ASCII)	Amount of discount applied to the line item.
$ITEM.DISCOUNTINDICATOR	O	Boolean value (default is False unless DISCOUNTAMOUNT is specified, in which case default is True)	Required only if any other line item information is present. Indicates if a discount was applied.
$ITEM.NATIONALTAXAMOUNT	O	0 - MAX INT (ASCII)	Amount of national tax (sales or VAT) applied to the line item.
$ITEM.NATIONALTAXRATE	O	0 - MAX DOUBLE (ASCII)	National tax (sales or VAT) rate applied to the line item.
$ITEM.NATIONALTAXTYPE	O	1 to 4 characters (UTF8)	Type of national tax applied to the line item.
$ITEM.LOCALTAXAMOUNT	O	0 - MAX INT (ASCII)	Amount of local tax applied to the line item.
$ITEM.OTHERTAXAMOUNT	O	0 - MAX INT (ASCII)	Amount of other taxes applied to the line item.
$ITEM.TOTALCOST	R*	0 - MAX INT (ASCII)	*Required only if any other line item information is present. The total cost of the line item.
Note: MAX INT = 2 147 483 647			

Appendix D. Retry mechanism used in WebSphere Commerce Payments

This appendix describes the retry mechanism used by WebSphere Commerce Payments to improve the reliability of the payment transfer process with the Cassette for SET. It also provides some tips that may help you when troubleshooting connection failures in your network.

Purchase process overview

Generally speaking, requests are made of WebSphere Commerce Payments to perform electronic commerce (eCommerce). The underlying purpose of eCommerce is to provide a mechanism for money (funds) to change hands in exchange for services and products. Because money is involved, the participants are typically very concerned with the accuracy of these transactions. Usually, the participating software products (components) exist on geographically dispersed equipment connected through the Internet and communicate through TCP/IP. Since any of these components (including the Internet) can fail, the need for a retry mechanism exists. WebSphere Commerce Payments implements a two-phase retry mechanism, described in this section, to greatly improve the reliability of the payment transfer process.

The interested parties in this process can be represented as follows:

```
Consumer <---> Merchant <---> WebSphere <---> Payment <--->Financial
Software       Software       Commerce        Gateway      Institution
                              Payments
```

The purchasing process begins when a consumer uses software (generally a Web browser) to interact with the software on the merchant host to initiate a purchase request. At this point, the merchant software communicates with WebSphere Commerce Payments to begin the transaction. Requests are sent by WebSphere Commerce Payments to the Payment Gateway, and there are corresponding replies. These requests are used to perform a transaction, and the response is used to communicate the success or failure of the transaction request. This appendix describes how the WebSphere Commerce Payments retry mechanism is used to increase the reliability of the transfer of funds between the participants.

References to *idempotency* in this appendix refer to the property that enables the sender of a request to repeat the request with a guarantee that the outcome will be the same regardless of regardless of whether the request or response is lost (or delayed due to network problems). In effect, idempotency ensures that multiple identical requests that are sent are ignored as duplicates and produce no change in the final result (a + a = a).

WebSphere Commerce Payments retry protocol

To make requests of WebSphere Commerce Payments, the merchant software uses one of the WebSphere Commerce Payments application programming interfaces (APIs). These API routines make connections from the merchant to WebSphere Commerce Payments, check for connection failures, and indicate failures to the merchant software with appropriate return codes. The merchant software can be written to retry the request, or inform the consumer of the failure, based on the needs of the merchant. Once the request is communicated to

WebSphere Commerce Payments, WebSphere Commerce Payments *may* be able to respond to the requests without communicating with the Payment Gateway.

If WebSphere Commerce Payments *must* communicate with the Payment Gateway to satisfy the request, a routine is invoked which attempts to open a connection (TCP/IP socket) from WebSphere Commerce Payments to the Payment Gateway. However, this request may not succeed. What to do about the failure depends on the underlying reason for the failure.

The SET specification anticipates the occurrence of network failures by allowing the same request to be sent to the gateway multiple times. The gateway must process the request only once, regardless of how many times it is received. It must send the same response message each time an identical request is received. However, it is unrealistic to expect the gateway to recognize a request as a duplicate forever. It is allowed to define a period of time during which duplicates are recognized. The software should only send duplicates within this time period. A duplicate is defined as an identical message, bit for bit.

WebSphere Commerce Payments has *automatic retry logic*. When no response is received for a request, it will automatically resend the original message as many times as specified in the retry configuration. If, for some reason, the merchant software does not handle this process correctly, the consumer, or the merchant software itself may issue multiple requests of WebSphere Commerce Payments through the API, resulting in multiple (non-identical) transaction requests being made of WebSphere Commerce Payments, which will not recognize these as duplicates. Several internal identifiers will exist that are not the same as the original message, so multiple duplicate transactions may be initiated and processed.

If the reason for the communication failure between WebSphere Commerce Payments and the Payment Gateway is due to a sporadic failure, a subsequent attempt to communicate may succeed. This attempt represents the *first phase* of the retry logic, which is the *immediate retry phase*. It is best used under circumstances where failure is occasional and another attempt will be likely to succeed. The problem with having a significant number of immediate retries is that the source of the failure may be more severe. While WebSphere Commerce Payments is attempting these retries, it is unable to process other commands. Under these circumstances, you probably do not want to configure an excessively high number of retries because that can cause WebSphere Commerce Payments resources to be consumed while attempting to perform requests that will not complete.

An example of a more severe error is when a failure occurs in an important Internet component such as a bridge, switch, or router. It is unlikely that a failure of this type will be corrected in a trivial amount of time. At this point, *second phase* of the retry logic is initiated. The retry logic for this phase is that the request will be attempted, and if it fails, a specific amount of time elapses before another attempt is made. The configuration parameters used to define this retry phase are delayRetryInterval, which defines the delay retry interval (in minutes), and MAXDELAYEDRETRIES, which defines the maximum number of delayed retry attempts that will be made. (Parameters are described in Chapter 7, “Using WebSphere Commerce Payments commands with Cassette for SET” on page 47.)

The primary consideration associated with the retry logic is concerned with how long the Payment Gateway maintains information about requests. For example, consider the situation where the request made by WebSphere Commerce Payments is one in which funds are transferred from one party to another. If the

request does make its way to the Payment Gateway, and from there, to the financial institution, the funds transfer can succeed. The success indication should then return to the Payment Gateway, and then to WebSphere Commerce Payments, and eventually to the merchant software. Consider the scenario where the communication failure occurs between WebSphere Commerce Payments and the Payment Gateway after the request has been successfully received and forwarded to the financial institution. The subsequent response (indicating success or failure) finds its way to the Payment Gateway, which is unable to send the response to WebSphere Commerce Payments because the connection has been disrupted.

The failure should be identified and corrected, or a backup/alternate route is established between WebSphere Commerce Payments and the Payment Gateway. A subsequent request to establish a connection from WebSphere Commerce Payments to the Payment Gateway succeeds, and the request is retransmitted to the Payment Gateway. The Payment Gateway is configured to recognize these types of requests as identical (from some amount of time), and respond to the request using the information that was stored when it was received earlier. This way, the funds transfer is not duplicated, and the reliability of the system is greatly enhanced.

Configuring the retry characteristics

You may be asking yourself how long should a Payment Gateway hold information, such that subsequent requests are identified as identical? This is a question that can only be answered by the business needs and practices of the group responsible for managing and maintaining the Payment Gateway. Customers of the IBM Payment Gateway use time periods ranging from 5 to almost 100 hours.

Therefore, to configure WebSphere Commerce Payments for optimal retry characteristics, you should know the idempotency window configured for a Payment Gateway. For the following scenario, assume that the Payment Gateway is configured for a 5-hour idempotency window. This means that the maximum time WebSphere Commerce Payments should use must be less than 5 hours. It is reasonable to configure 1 or 2 immediate retries, and to configure the delayed retries to happen every 10 or 15 minutes for almost 5 hours. For instance, you could configure the delayed retry interval to be 10 minutes, and the number of delayed retries to be 28. By using 28 instead of 30, you are allowing for the time the immediate retries take plus a little extra to make sure that all the retry attempts complete within the 5-hour idempotency window.

Checking for failures

How does the administrator of WebSphere Commerce Payments know if and when these types of failures occur? By checking the PMError log file, you can find a message identified by CEPSET0617 whenever an attempt to establish a connection to the gateway fails. This message appears once for every immediate retry that fails. If all of the configured immediate retries fail and the delayed retry phase is initiated, the transaction is queued for a delayed retry. You can find a message identified with CEPSET0301. This message informs you that the transaction is being queued for a delayed retry. If all retries fail, message CEPSET0300 appears on the last retry.

No action is required unless all retries fail. For all 5 hours worth of retries to fail, one or more of the following would have to occur for the entire 5 hours:

- The network would have to be extremely congested.

- The Payment Gateway would have to be down.
- A significant Internet component (bridge, router, switch) would have to be down.

You should expect that most of your transactions will succeed at some point within that 5 hours without any intervention on your part.

However, since a functioning network is required for a transaction to complete, and since networks failures cannot be prevented all of the time, every business must have processes and procedures for dealing with transactions that fail completely. These procedures depend on your business and the policy of your acquirer or gateway.

Implementing a strategy

Generally, a strategy such as the following could be used:

1. Once a day, determine which transactions have failed completely by checking the database for messages with the ″IdempotencyFailure″ field set.
2. Communicate with your acquirer. There are two possibilities:
   - The gateway received and processed the message but you did not receive the response. In this case, the batch totals in WebSphere Commerce Payments will not match the batch totals at the Payment Gateway. Make note of this so that when you receive your statement you can take these transactions into account and make the appropriate adjustments (for example, human intervention may be required to forcibly close a batch).
   - The Payment Gateway did not receive the message. In this case, you should reissue the transaction (for example, deposit, authorize, and so on). WebSphere Commerce Payments will have put the transaction into the correct state to allow the transaction after the final retry failed.

Other types of errors

The previous example focused on the CEPSET0617 messages, which indicate the failure to establish a connection. This is a common type of network error, but there are other possible network errors. For example, the connection might be established, but a failure occurs when either writing the request or reading the response. In this case, you may find occurrences of the CEPSET0626 message. Or, the request may reach the gateway, but the gateway may timeout the HTTP connection to the merchant before a response was built. In this case, message CEPSET0307 is written, indicating that a null response was received from the gateway. The trace will show that the gateway returned an HTTP header with an error code of 400, but no message body containing a SET response. All of these conditions are to be expected occasionally. All of these network errors will result in the transaction being automatically retried for the configured amount of time. These errors indicate network trouble, not software failure. Almost all occurrences of these failures will be transient and handled by appropriately configured retry values.

If an excess of any one of these errors is witnessed, some type of network tuning may be needed. For instance, too many CEPSET0626 messages may mean that you have configured the read timeout too short for your network conditions. Too many CEPSET0307 messages may mean that the gateway has set its HTTP connection timeout too short. Too many CEPSET0617 messages may mean that the gateway is down, or some other network condition is preventing you from even reaching the gateway. (If CEPSET0617 is received on every single attempt and you know the gateway is running, the configuration of the gateway host or port may be incorrect.)

If you configure *both* phases of the retry logic to take advantage of the full 5-hour window, it is expected that you should see almost all transactions succeeding within 5 hours without any intervention on the part of the WebSphere Commerce Payments administrator. However, it is unrealistic to expect that network trouble will never cause the complete failure of a transaction. Appropriate business practices and procedures must be in place to deal with an occasional failures (through discussion with the appropriate acquirer).

Appendix E. Configuring the 4758

This appendix provides you with information and considerations regarding the configuration of the 4758 Cryptographic Coprocessor.

Prerequisite product levels

The Cassette for SET supports the following models of the IBM 4758 Cryptographic Coprocessor:

- 4758–1
- 4758–13
- 4758–2
- 4758–23

If you have model 4758–1 or 4758–13 and are interested in upgrading to 4758–2 or 4758–23, go to **http://www.ibm.com/security/cryptocards**.

By default, WebSphere Commerce Payments installs the CCA library for the latest 4758 coprocessor (that is, Model 2 or 23). If you are running WebSphere Commerce Payments with the older coprocessor, a different CCA library must be used. It is recommended that you upgrade to the latest support program level for the CCA supported for your card. Following are the minimum requirements for the CCA:

- For Models 1 and 13, CCA support program level 1.3.1, or later. It is, however, recommended that you use support program level 1.3.2.
- For Models 2 and 23, CCA support program level 2.2, or later.

For the latest information on the current levels, see **http://www.ibm.com/security/cryptocards**. For information on migrating from a previous version of WebSphere Commerce Payments, see the support Web site at **www.ibm.com/software/webservers/paymgr/support/**.

Note: You cannot have both Model 2 and Model 23 cards functioning in the same machine at the same time.

Configuring the 4758 PCI cryptographic coprocessor and CCA support program

If you are using the 4758 PCI Cryptographic Coprocessor, do not configure WebSphere Commerce Payments to start using the coprocessor until after you have successfully exchanged data with an acquirer. Installing and configuring WebSphere Commerce Payments before the 4758 card is installed makes it easier to identify and eliminate problems that may have occurred when you installed WebSphere Commerce Payments.

The general steps to take are:

1. Ensure that any required AIX PTFs are applied.
2. Install hardware and device driver.
3. Install the latest CCA software appropriate for your card.
4. Load firmware using *csufclu*.
5. Apply any FixPaks as necessary.
6. Configure crypto node using *csufcnm.*

a. Define administrative roles and profiles (may be one or more, depending on your installation security requirements):
   - To create, modify and delete other roles and profiles
   - To manage key storage

   Note: Perform the remaining items for this step under the appropriate administrative profile.

b. Define the role (and if necessary, the profile) under which WebSphere Commerce Payments will run.
c. Synchronize the coprocessor clock with the AIX system.
d. Create master key.
e. Initialize key storage for DES and PKA keys.
f. Load the function control vector.

7. Set up AIX permissions for access to executables using *csufappl* and `csufadmin`.
8. If you are using model 1 or 13 rename the library *ibmcokgskidcc1.a* as *ibmcokgskidcc.a*.
9. Start WebSphere Commerce Payments.
10. Using the WebSphere Commerce Payments user interface, go to Cassettes –> Cassette for SET –> Advanced Settings, to set the Hardware CryptoID and the Hardware Crypto Password.
11. Shut down and restart WebSphere Commerce Payments.
12. Use the WebSphere Commerce Payments user interface to set up your Merchant, Account, and Brand.

Coexistence considerations

If other applications on your system are using the 4758 Cryptographic Coprocessor, make sure the steps you perform for WebSphere Commerce Payments installation and configuration do not hinder the ability of the other applications to use the coprocessor.

Note that it is highly recommended that you do not share the 4758 Coprocessor with other applications on your system. For example, when sharing roles between multiple applications, ensure that you are granting the appropriate level of authority to users.

IBM Payment Registry and the IBM Payment Gateway for AIX products can use the 4758 coprocessor; therefore, if either of these products is installed on the WebSphere Commerce Payments AIX system, the 4758 may be shared by more than one application.

4758 role and profile requirements

One of the steps in configuring the 4758 coprocessor involves the definition of various roles and profiles under which the coprocessor may be used. This section provides the information you will need to plan for WebSphere Commerce Payments during this phase of configuration.

Refer to the *4758 PCI Cryptographic Coprocessor CCA Support Program* manual (SC31–8610) for a complete discussion on roles and profiles.

It is critical to ensure that you always have an active role and profile with the authority to create, modify and delete other roles and profiles. If you lose this

capability, you could lose the ability to administer the 4758 coprocessor. Before you modify the characteristics of the default role (as it is shipped from IBM), you should:

1. Define another role that is allowed to manage the 4758 roles and profiles
2. Define a profile on the new role
3. Log in under the new profile to ensure that it is active

One of the steps in configuring the 4758 coprocessor involves defining the roles under which the coprocessor may be used. WebSphere Commerce Payments must run under a role which allows access to the following functions:

- Compose SET Block
- Compute Verification Pattern
- Decipher
- Decompose SET Block
- Encipher
- Generate Key
- PKA96 Digital Signature Generate
- PKA96 Digital Signature Verify
- PKA96 One Way Hash
- PKA96 PKA Key Generate

You may either allow this access under the DEFAULT role, or you may choose to create a separate role specifically for WebSphere Commerce Payments to use. If you choose the latter approach, then you will also need to create a user profile whose capabilities are determined by the role you defined. If you are planning to use the default role, then it is not necessary to create a user profile.

If you have the CCA support program version 1.3.0 or earlier installed, the default role must also enable the Compute Verification Pattern function. If this function is not allowed, the WebSphere Commerce Payments user interface will fail with an error code.

Expiration of 4758 user profile passwords

Be aware that 4758 user profile passwords (or *passphrases*, as they are called in the 4758 documentation) have a default life of 90 days from the time they are defined. This section describes how to reset the 4758 passphrase for your WebSphere Commerce Payments user profile. If WebSphere Commerce Payments runs under the 4758 default role, then you may skip this section. See ″SET Configuration Profile″ section of the *WebSphere Commerce Payments Administrator's Guide* for details.

If your WebSphere Commerce Payments runs under a specific 4758 user profile, then you need to ensure that the profile passphrase does not expire. If the passphrase is allowed to expire, WebSphere Commerce Payments will fail on any SET transaction that requires access to the 4758. Use this procedure to update 4758 user profile:

Note: These steps must be performed before the password expires under a 4758 user profile that has the authority to create, modify, and delete other user profiles. Depending on your local 4758 administration policies, you may have to ask your local 4758 administrator to perform this operation for you. Note that if the passphrase does expire, you need only delete the profile and create a new one.

1. Start the *csufcnm* utility (provided with the CCA Support Program that came with the 4758 adapter) and log in under a user profile that has the authority to add, delete, and modify other user profiles. This will usually be a different profile than the one under which WebSphere Commerce Payments runs.
2. Edit the WebSphere Commerce Payments 4758 user profile (Access Control->Profiles->Edit). Under the edit dialog, enter a new passphrase in the ″Passphrase″ and the ″Confirm passphrase″ entry boxes. To put the new passphrase into effect, press the **Replace** button. The new passphrase will expire 90 days from the date it is set.
3. Exit the *csufcnm* utility.
4. Use the WebSphere Commerce Payments user interface to change the HardwareCryptoPassword value to the new passphrase value.
5. Shut down WebSphere Commerce Payments.
6. Restart WebSphere Commerce Payments.

Accessing libraries and files

To ensure that WebSphere Commerce Payments has the necessary permissions to access the CCA libraries, use the *csufappl* or the `csufadmin` utility. For more details on the *csufappl* and the `csufadmin` utility, refer to the *csufappl* and the `csufadmin` main pages after the 4758 hardware and software have been installed.

Since WebSphere Commerce Payments causes keys to be generated, the user ID under which these will run, or a group to which that user ID belongs, must have write access to the files in the CCA keys database directory. The default is */usr/lpp/csufkeys*. The easiest way to verify this is to ensure that the group name used with the csufappl and csuadmin commands is the same as the group name of the user which starts WebSphere Commerce Payments. This directory can be modifed with `csufkeys`.

Deleting CDSA ODM objects

If you are migrating from a previous version of WebSphere Commerce Payments (such as from Payment Manager) and you previously configured the 4758 hardware, you no longer need CDSA ODM objects for use with WebSphere Commerce Payments. Since WebSphere Commerce Payments no longer uses CDSA, you can uninstall support and delete CDSA ODM objects. Note that you should not delete CDSA support unless you are sure that none of the remaining applications use the 4758 coprocessor. Once you delete a CDSA ODM object, you delete that object for all users, not just for WebSphere Commerce Payments.

To delete CDS ODM objects:

1. Log on as **root.**
2. Make sure that no other applications are using the 4758 coprocessor. Deleting objects means these objects are deleted for all users.
3. Change directory to `/usr/lpp/PaymentManager/cdsa`
4. Execute the following commands:

   ```
   CDSAforPS uninstall
   ```

   or

   ```
   install_ibmcca libccacsp.a /usr/lpp/IBM_Payment_Server/lib uninstall
   install_cssm libcssm32.a /usr/lpp/IBM_Payment_Server/lib uninstall
   install_ibmswcsp libibmswcsp.a /usr/lpp/WebSpherePaymentManager/lib uninstall
   ```

4758 error codes

When 4758–related errors occur, they display as errors in the WebSphere Commerce Payments eTillSETMsgTrace and eTillSETMsgDump files (use the *fmttrace.exe* utility to format thse files to readable form). For more information regarding error codes, check the *CCA Basic Services Reference and Guide*.

Appendix F. Cassette for SET Return Codes

The following table lists secondary return codes specific to the SET protocol.

RC_SET_UNSPECIFIED_ERROR	1001	Unspecified error.
RC_SET_BATCH_ID	1002	Batch ID was either (1) specified when prohibited or (2) not specified when required.
RC_SET_REFUND_AMOUNT_NOT_ZERO	1003	The SET Cassette allows only complete refund reversals (that is, the amount must be zero).
RC_SET_OPERATION_FAILED	1004	The operation experienced financial failure.
RC_SET_PENDING_ACQUIRER	1005	The operation did not complete because it is pending on an acquirer entity.
RC_SET_PENDING_BATCH	1006	The operation did not complete because it is pending on a batch entity.
RC_SET_PENDING_ORDER	1007	The operation did not complete because it is pending on an order entity.
RC_SET_ENCRYPTION_ERROR	1008	A SET encryption error occurred while composing or processing a message.
RC_SET_UNDEFINED_ACQUIRER	1009	The acquirer object associated with the operation is undefined.
RC_SET_IMPLICIT_BATCHES_ONLY	1010	A etBatchOpen() or etBatchClose() command was issued for an acquirer that controls its own batches.
RC_SET_BATCH_CURRENCY	1011	The currency for all payments/credits in a batch must be the same.
RC_SET_BATCH_AMOUNTEXP10	1012	The amount exponent for all payments/credits in a batch must be the same.

RC_SET_NO_CERTIFIED_BRANDS	1013	No certified brands exist for a given merchant, and so an initialization message cannot be generated.
RC_SET_BRAND	1014	Response refers to the brand parameter (specified in protocol data).
RC_SET_PAN	1015	Response refers to the PAN parameter (specified in protocol data).
RC_SET_EXPIRY	1016	Response refers to the expiry parameter (specified in protocol data).
RC_SET_DEPOSIT_AMOUNT_NOT_ZERO	1017	This acquirer only allows complete deposit reversals (that is, the amount must be zero).
RC_SET_ACQUIRER_COMMUNICATION_ERROR	1018	A communication error between WebSphere Commerce Payments and the acquirer occurred.
RC_SET_ACQUIRER_RESPONSE_NULL	1019	The acquirer communicated a NULL response, which was not expected.
RC_SET_ACQUIRER_RESPONSE_UNEXPECTED	1020	The response from the acquirer had an unexpected type.
RC_SET_BATCH_ERROR	1021	A batch-related error occurred.
RC_SET_BATCH_BALANCE_ERROR	1022	The totals for this batch calculated by the merchant and the acquirer did not match.
RC_SET_CERTLESS_ERROR	1023	The certificate for this brand cannot be used with the etAcceptPayment() API function to make a certless purchase.
RC_SET_DOMAIN_NAME_ERROR	1024	The SET domain name is not valid.
RC_SET_MERORDERNUM	1025	Response refers to the SET merchant order number parameter.

RC_SET_MERCHCATCODE	1026	Response refers to the SET MerchCatCode parameter.
RC_SET_MERCHGROUP	1027	Response refers to the SET MerchGroup parameter.
RC_SET_REQUIRECARDCERT	1028	Response refers to the SET RequireCardCert parameter.
RC_SET_TRANSMETHOD	1029	Response refers to the SET TransMethod parameter.
RC_SET_MSG_PROCESSING_FAILED	1030	Processing of SET Message from acquirer failed and a SET Error Message was sent.
RC_SET_BATCHID	1040	Response refers to the SET BatchId parameter.
RC_SET_AVSDATA	1051	Response refers to the SET AVSdata parameter.
RC_SET_AVS_COUNTRYCODE	1052	Response refers to the SET AVS CountryCode parameter.
RC_SET_AVS_STREETADDRESS	1053	Response refers to the SET AVS StreetAddress parameter.
RC_SET_AVS_CITY	1054	Response refers to the SET AVS City parameter.
RC_SET_AVS_STATEPROVINCE	1055	Response refers to the SET AVS StateProvince parameter.
RC_SET_AVS_POSTALCODE	1056	Response refers to the SET AVS PostalCode parameter.
RC_SET_AVS_LOCATIONID	1057	Response refers to the SET AVS LocationId parameter.
RC_SET_ACQUIRER_VIOLATED_PROFILE	1058	The acquirer responded with a batchid that does not match a currently open batch.

RC_SET_SALEDETAIL	1100	Response refers to the SET SaleDetail parameter.
RC_SET_SD_PAYRECURIND	1101	Response refers to the SET SaleDetail PayRecurInd parameter.
RC_SET_SD_AUTHCHARIND	1102	Response refers to the SET SaleDetail AuthCharInd parameter.
RC_SET_SD_MARKETSPECSALEDATA	1103	Response refers to the SET SaleDetail MarketSpecSaleData parameter.
RC_SET_SD_COMMERCIALCARDDATA	1104	Response refers to the SET SaleDetail CommercialCardData parameter.
RC_SET_SD_ORDERSUMMARY	1105	Response refers to the SET SaleDetail OrderSummary parameter.
RC_SET_SD_CUSTOMERREFERENCENUMBER	1106	Response refers to the SET SaleDetail CustomerReferenceNumber parameter.
RC_SET_SD_CUSTOMERSERVICEPHONE	1107	Response refers to the SET SaleDetail CustomerServicePhone parameter.
RC_SET_SD_OKTOPRINTPHONEIND	1108	Response refers to the SET SaleDetail OkToPrintPhoneInd parameter.
RC_SET_SD_CCARD_MERCHANTLOCATION	1109	Response refers to the SET SaleDetail CommercialCardData MerchantLocation parameter.
RC_SET_SD_CCARD_MERCHANTLOCATION_COUNTRYCODE	1110	Response refers to the SET SaleDetail CommercialCardData MerchantLocation CountryCode parameter.
RC_SET_SD_CCARD_MERCHANTLOCATION_CITY	1111	Response refers to the SET SaleDetail CommercialCardData MerchantLocation City parameter.

RC_SET_SD_CCARD_MERCHANTLOCATION_STATEPROVINCE	1112	Response refers to the SET SaleDetail CommercialCardData MerchantLocation StateProvince parameter.
RC_SET_SD_CCARD_MERCHANTLOCATION_POSTALCODE	1113	Response refers to the SET SaleDetail CommercialCardData MerchantLocation PostalCode parameter.
RC_SET_SD_CCARD_MERCHANTLOCATION_LOCATIONID	1114	Response refers to the SET SaleDetail CommercialCardData MerchantLocation LocationId parameter.
RC_SET_SD_CCARD_SHIPFROM	1115	Response refers to the SET SaleDetail CommercialCardData ShipForm parameter.
RC_SET_SD_CCARD_SHIPFROM_COUNTRYCODE	1116	Response refers to the SET SaleDetail CommercialCardData ShipForm CountryCode parameter.
RC_SET_SD_CCARD_SHIPFROM_CITY	1117	Response refers to the SET SaleDetail CommercialCardData ShipForm City parameter.
RC_SET_SD_CCARD_SHIPFROM_STATEPROVINCE	1118	Response refers to the SET SaleDetail CommercialCardData ShipForm StateProvince parameter.
RC_SET_SD_CCARD_SHIPFROM_POSTALCODE	1119	Response refers to the SET SaleDetail CommercialCardData ShipForm PostalCode parameter.
RC_SET_SD_CCARD_SHIPFROM_LOCATIONID	1120	Response refers to the SET SaleDetail CommercialCardData ShipForm LocationId parameter.
RC_SET_SD_CCARD_SHIPTO	1121	Response refers to the SET SaleDetail CommercialCardData ShipTo parameter.

RC_SET_SD_CCARD_SHIPTO_COUNTRYCODE	1122	Response refers to the SET SaleDetail CommercialCardData ShipTo CountryCode parameter.
RC_SET_SD_CCARD_SHIPTO_CITY	1123	Response refers to the SET SaleDetail CommercialCardData ShipTo City parameter.
RC_SET_SD_CCARD_SHIPTO_STATEPROVINCE	1124	Response refers to the SET SaleDetail CommercialCardData ShipTo StateProvince parameter.
RC_SET_SD_CCARD_SHIPTO_POSTALCODE	1125	Response refers to the SET SaleDetail CommercialCardData ShipTo PostalCode parameter.
RC_SET_SD_CCARD_SHIPTO_LOCATIONID	1126	Response refers to the SET SaleDetail CommercialCardData ShipTo LocationId parameter.
RC_SET_SD_CCARD_CHARGEINFO	1127	Response refers to the SET SaleDetail CommercialCardData ChargeInfo parameter.
RC_SET_SD_CCARD_CHARGEINFO_TOTALFREIGHTSHIPPINGAMOUNT	1128	Response refers to the SET SaleDetail CommercialCardData ChargeInfo TotalFreightShippingAmount parameter.
RC_SET_SD_CCARD_CHARGEINFO_TOTALDUTYTARIFFAMOUNT	1129	Response refers to the SET SaleDetail CommercialCardData ChargeInfo TotalDutyTariffAmount parameter.
RC_SET_SD_CCARD_CHARGEINFO_DUTYTARIFFREFERENCE	1130	Response refers to the SET SaleDetail CommercialCardData ChargeInfo DutyTariffReference parameter.
RC_SET_SD_CCARD_CHARGEINFO_TOTALNATIONALTAXAMOUNT	1131	Response refers to the SET SaleDetail CommercialCardData ChargeInfo TotalNationalTaxAmount parameter.

RC_SET_SD_CCARD_CHARGEINFO_TOTALLOCALTAXAMOUNT	1132	Response refers to the SET SaleDetail CommercialCardData ChargeInfo TotalLocalTaxAmount parameter.
RC_SET_SD_CCARD_CHARGEINFO_TOTALOTHERTAXAMOUNT	1133	Response refers to the SET SaleDetail CommercialCardData ChargeInfo TotalOtherTaxAmount parameter.
RC_SET_SD_CCARD_CHARGEINFO_TOTALTAXAMOUNT	1134	Response refers to the SET SaleDetail CommercialCardData ChargeInfo TotalTaxAmount parameter.
RC_SET_SD_CCARD_CHARGEINFO_MERCHANTTAXID	1135	Response refers to the SET SaleDetail CommercialCardData ChargeInfo MerchantTaxId parameter.
RC_SET_SD_CCARD_CHARGEINFO_MERCHANTDUTYTARIFFREF	1136	Response refers to the SET SaleDetail CommercialCardData ChargeInfo MerchantDutyTariffRef parameter.
RC_SET_SD_CCARD_CHARGEINFO_CUSTOMERDUTYTARIFFREF	1137	Response refers to the SET SaleDetail CommercialCardData ChargeInfo CustomerDutyTariffRef parameter.
RC_SET_SD_CCARD_CHARGEINFO_SUMMARYCOMMODITYCODE	1138	Response refers to the SET SaleDetail CommercialCardData ChargeInfo SummaryCommodityCode parameter.
RC_SET_SD_CCARD_CHARGEINFO_MERCHANTTYPE	1139	Response refers to the SET SaleDetail CommercialCardData ChargeInfo MerchantType parameter.
RC_SET_SD_CCARD_ITEMSEQ	1140	Response refers to the SET SaleDetail CommercialCardData ItemSeq parameter.
RC_SET_SD_CCARD_ITEM	1141	Response refers to the SET SaleDetail CommercialCardData Item parameter.

RC_SET_SD_CCARD_ITEM_QUANTITY	1142	Response refers to the SET SaleDetail CommercialCardData Item Quantity parameter.
RC_SET_SD_CCARD_ITEM_UNITOFMEASURECODE	1143	Response refers to the SET SaleDetail CommercialCardData Item UnitOfMeasureCode parameter.
RC_SET_SD_CCARD_ITEM_DESCRIPTOR	1144	Response refers to the SET SaleDetail CommercialCardData Item Descriptor parameter.
RC_SET_SD_CCARD_ITEM_COMMODITYCODE	1145	Response refers to the SET SaleDetail CommercialCardData Item CommodityCode parameter.
RC_SET_SD_CCARD_ITEM_PRODUCTCODE	1146	Response refers to the SET SaleDetail CommercialCardData Item ProductCode parameter.
RC_SET_SD_CCARD_ITEM_UNITCOST	1147	Response refers to the SET SaleDetail CommercialCardData Item UnitCost parameter.
RC_SET_SD_CCARD_ITEM_NETCOST	1148	Response refers to the SET SaleDetail CommercialCardData Item NetCost parameter.
RC_SET_SD_CCARD_ITEM_DISCOUNTIND	1149	Response refers to the SET SaleDetail CommercialCardData Item DiscountInd parameter.
RC_SET_SD_CCARD_ITEM_DISCOUNTAMOUNT	1150	Response refers to the SET SaleDetail CommercialCardData Item DiscountAmount parameter.
RC_SET_SD_CCARD_ITEM_NATIONALTAXAMOUNT	1151	Response refers to the SET SaleDetail CommercialCardData Item NationalTaxAmount parameter.

RC_SET_SD_CCARD_ITEM_NATIONALTAXRATE	1152	Response refers to the SET SaleDetail CommercialCardData Item NationalTaxRate parameter.
RC_SET_SD_CCARD_ITEM_NATIONALTAXTYPE	1153	Response refers to the SET SaleDetail CommercialCardData Item NationalTaxType parameter.
RC_SET_SD_CCARD_ITEM_LOCALTAXAMOUNT	1154	Response refers to the SET SaleDetail CommercialCardData Item LocalTaxAmount parameter.
RC_SET_SD_CCARD_ITEM_OTHERTAXAMOUNT	1155	Response refers to the SET SaleDetail CommercialCardData Item OtherTaxAmount parameter.
RC_SET_SD_CCARD_ITEM_ITEMTOTALCOST	1156	Response refers to the SET SaleDetail CommercialCardData Item ItemTotalCost parameter.
RC_SET_SD_MARKETSPECDATAID	1157	Response refers to the SET SaleDetail MarketSpecDataId parameter.
RC_SET_SD_MARKETSPECCAPDATA	1158	Response refers to the SET SaleDetail MarketSpecCapData parameter.
RC_SET_SD_MARKETAUTOCAP	1159	Response refers to the SET SaleDetail MarketAutoCap parameter.
RC_SET_SD_AUTO_RENTERNAME	1160	Response refers to the SET SaleDetail MarketAutoCap RenterName parameter.
RC_SET_SD_AUTO_RENTALLOCATION	1161	Response refers to the SET SaleDetail MarketAutoCap RentalLocation parameter.
RC_SET_SD_AUTO_RENTALLOCATION_COUNTRYCODE	1162	Response refers to the SET SaleDetail MarketAutoCap RentalLocation CountryCode parameter.

RC_SET_SD_AUTO_RENTALLOCATION_CITY	1163	Response refers to the SET SaleDetail MarketAutoCap RentalLocation City parameter.
RC_SET_SD_AUTO_RENTALLOCATION_STATEPROVINCE	1164	Response refers to the SET SaleDetail MarketAutoCap RentalLocation StateProvince parameter.
RC_SET_SD_AUTO_RENTALLOCATION_POSTALCODE	1165	Response refers to the SET SaleDetail MarketAutoCap RentalLocation PostalCode parameter.
RC_SET_SD_AUTO_RENTALLOCATION_LOCATIONID	1166	Response refers to the SET SaleDetail MarketAutoCap RentalLocation LocationId parameter.
RC_SET_SD_AUTO_RENTALDATETIME	1167	Response refers to the SET SaleDetail MarketAutoCap RentalDateTime parameter.
RC_SET_SD_AUTO_AUTONOSHOW	1168	Response refers to the SET SaleDetail MarketAutoCap AutoNoShow parameter.
RC_SET_SD_AUTO_RENTALAGREEMENTNUMBER	1169	Response refers to the SET SaleDetail MarketAutoCap RentalAgreementNumber parameter.
RC_SET_SD_AUTO_REFERENCENUMBER	1170	Response refers to the SET SaleDetail MarketAutoCap ReferenceNumber parameter.
RC_SET_SD_AUTO_INSURANCETYPE	1171	Response refers to the SET SaleDetail MarketAutoCap InsuranceType parameter.
RC_SET_SD_AUTO_RATEINFO	1172	Response refers to the SET SaleDetail MarketAutoCap RateInfo parameter.

RC_SET_SD_AUTO_RATEINFO_AUTOAPPLICABLERATE	1173	Response refers to the SET SaleDetail MarketAutoCap RateInfo AutoApplicableRate parameter.
RC_SET_SD_AUTO_RATEINFO_DAILYRENTALRATE	1174	Response refers to the SET SaleDetail MarketAutoCap RateInfo DailyRentalRate parameter.
RC_SET_SD_AUTO_RATEINFO_WEEKLYRENTALRATE	1175	Response refers to the SET SaleDetail MarketAutoCap RateInfo WeeklyRentalRate parameter.
RC_SET_SD_AUTO_RATEINFO_LATERETURNHOURLYRATE	1176	Response refers to the SET SaleDetail MarketAutoCap RateInfo LateReturnHourlyRate parameter.
RC_SET_SD_AUTO_RATEINFO_DISTANCERATE	1177	Response refers to the SET SaleDetail MarketAutoCap RateInfo DistanceRate parameter.
RC_SET_SD_AUTO_RATEINFO_FREEDISTANCE_SCALE	1178	Response refers to the SET SaleDetail MarketAutoCap RateInfo FreeDistance Scale parameter.
RC_SET_SD_AUTO_RATEINFO_FREEDISTANCE_DIST	1179	Response refers to the SET SaleDetail MarketAutoCap RateInfo FreeDistance Dist parameter.
RC_SET_SD_AUTO_RATEINFO_VEHICLECLASSCODE	1180	Response refers to the SET SaleDetail MarketAutoCap RateInfo VehicleClassCode parameter.
RC_SET_SD_AUTO_RATEINFO_CORPORATEID	1181	Response refers to the SET SaleDetail MarketAutoCap RateInfo CorporateId parameter.

RC_SET_SD_AUTO_RETURNLOCATION	1182	Response refers to the SET SaleDetail MarketAutoCap ReturnLocation parameter.
RC_SET_SD_AUTO_RETURNLOCATION_COUNTRYCODE	1183	Response refers to the SET SaleDetail MarketAutoCap ReturnLocation CountryCode parameter.
RC_SET_SD_AUTO_RETURNLOCATION_CITY	1184	Response refers to the SET SaleDetail MarketAutoCap ReturnLocation City parameter.
RC_SET_SD_AUTO_RETURNLOCATION_STATEPROVINCE	1185	Response refers to the SET SaleDetail MarketAutoCap ReturnLocation StateProvince parameter.
RC_SET_SD_AUTO_RETURNLOCATION_POSTALCODE	1186	Response refers to the SET SaleDetail MarketAutoCap ReturnLocation PostalCode parameter.
RC_SET_SD_AUTO_RETURNLOCATION_LOCATIONID	1187	Response refers to the SET SaleDetail MarketAutoCap ReturnLocation LocationId parameter.
RC_SET_SD_AUTO_RETURNDATETIME	1188	Response refers to the SET SaleDetail MarketAutoCap ReturnDateTime parameter.
RC_SET_SD_AUTO_CHARGES	1189	Response refers to the SET SaleDetail MarketAutoCap Charges parameter.
RC_SET_SD_AUTO_CHARGES_REGULARDISTANCE	1190	Response refers to the SET SaleDetail MarketAutoCap Charges RegularDistance parameter.
RC_SET_SD_AUTO_CHARGES_LATERETURN	1191	Response refers to the SET SaleDetail MarketAutoCap Charges LateReturn parameter.

RC_SET_SD_AUTO_CHARGES_TOTALDISTANCE_SCALE	1192	Response refers to the SET SaleDetail MarketAutoCap Charges TotalDistance Scale parameter.
RC_SET_SD_AUTO_CHARGES_TOTALDISTANCE_DIST	1193	Response refers to the SET SaleDetail MarketAutoCap Charges TotalDistance Dist parameter.
RC_SET_SD_AUTO_CHARGES_EXTRADISTANCE	1194	Response refers to the SET SaleDetail MarketAutoCap Charges ExtraDistance parameter.
RC_SET_SD_AUTO_CHARGES_INSURANCE	1195	Response refers to the SET SaleDetail MarketAutoCap Charges Insurance parameter.
RC_SET_SD_AUTO_CHARGES_FUEL	1196	Response refers to the SET SaleDetail MarketAutoCap Charges Fuel parameter.
RC_SET_SD_AUTO_CHARGES_AUTOTOWING	1197	Response refers to the SET SaleDetail MarketAutoCap Charges AutoTowing parameter.
RC_SET_SD_AUTO_CHARGES_ONEWAYDROPOFF	1198	Response refers to the SET SaleDetail MarketAutoCap Charges OnewayDropOff parameter.
RC_SET_SD_AUTO_CHARGES_TELEPHONE	1199	Response refers to the SET SaleDetail MarketAutoCap Charges Telephone parameter.
RC_SET_SD_AUTO_CHARGES_VIOLATIONS	1200	Response refers to the SET SaleDetail MarketAutoCap Charges Violations parameter.
RC_SET_SD_AUTO_CHARGES_DELIVERY	1201	Response refers to the SET SaleDetail MarketAutoCap Charges Delivery parameter.

RC_SET_SD_AUTO_CHARGES_PARKING	1202	Response refers to the SET SaleDetail MarketAutoCap Charges Parking parameter.
RC_SET_SD_AUTO_CHARGES_OTHER	1203	Response refers to the SET SaleDetail MarketAutoCap Charges Other parameter.
RC_SET_SD_AUTO_CHARGES_TOTALTAXAMOUNT	1204	Response refers to the SET SaleDetail MarketAutoCap Charges TotalTaxAmount parameter.
RC_SET_SD_AUTO_CHARGES_AUDITADJUSTMENT	1205	Response refers to the SET SaleDetail MarketAutoCap Charges AuditAdjustment parameter.
RC_SET_SD_MARKETHOTELCAP	1206	Response refers to the SET SaleDetail MarketHotelCap parameter.
RC_SET_SD_HOTEL_ARRIVALDATE	1207	Response refers to the SET SaleDetail MarketHotelCap ArrivalDate parameter.
RC_SET_SD_HOTEL_HOTELNOSHOW	1208	Response refers to the SET SaleDetail MarketHotelCap HotelNoShow parameter.
RC_SET_SD_HOTEL_DEPARTUREDATE	1209	Response refers to the SET SaleDetail MarketHotelCap DepartureDate parameter.
RC_SET_SD_HOTEL_DURATIONOFSTAY	1210	Response refers to the SET SaleDetail MarketHotelCap DurationOfStay parameter.
RC_SET_SD_HOTEL_FOLIONUMBER	1211	Response refers to the SET SaleDetail MarketHotelCap FolioNumber parameter.
RC_SET_SD_HOTEL_PROPERTYPHONE	1212	Response refers to the SET SaleDetail MarketHotelCap PropertyPhone parameter.

RC_SET_SD_HOTEL_CUSTOMERSERVICEPHONE	1213	Response refers to the SET SaleDetail MarketHotelCap CustomerServicePhone parameter.
RC_SET_SD_HOTEL_PROGRAMCODE	1214	Response refers to the SET SaleDetail MarketHotelCap ProgramCode parameter.
RC_SET_SD_HOTEL_RATEINFO	1215	Response refers to the SET SaleDetail MarketHotelCap RateInfo parameter.
RC_SET_SD_HOTEL_RATEINFO_DAILYROOMRATE	1216	Response refers to the SET SaleDetail MarketHotelCap RateInfo DailyRoomRate parameter.
RC_SET_SD_HOTEL_RATEINFO_DAILYTAXRATE	1217	Response refers to the SET SaleDetail MarketHotelCap RateInof DailyTaxRate parameter.
RC_SET_SD_HOTEL_CHARGES	1218	Response refers to the SET SaleDetail MarketHotelCap Charges parameter.
RC_SET_SD_HOTEL_CHARGES_ROOM	1219	Response refers to the SET SaleDetail MarketHotelCap Charges Room parameter.
RC_SET_SD_HOTEL_CHARGES_ROOMTAX	1220	Response refers to the SET SaleDetail MarketHotelCap Charges RoomTax parameter.
RC_SET_SD_HOTEL_CHARGES_PREPAIDEXPENSES	1221	Response refers to the SET SaleDetail MarketHotelCap Charges PrepaidExpenses parameter.
RC_SET_SD_HOTEL_CHARGES_FOODBEVERAGE	1222	Response refers to the SET SaleDetail MarketHotelCap Charges FoodBeverage parameter.

RC_SET_SD_HOTEL_CHARGES_ROOMSERVICE	1223	Response refers to the SET SaleDetail MarketHotelCap Charges RoomService parameter.
RC_SET_SD_HOTEL_CHARGES_MINIBAR	1224	Response refers to the SET SaleDetail MarketHotelCap Charges MiniBar parameter.
RC_SET_SD_HOTEL_CHARGES_LAUNDRY	1225	Response refers to the SET SaleDetail MarketHotelCap Charges Laundry parameter.
RC_SET_SD_HOTEL_CHARGES_TELEPHONE	1226	Response refers to the SET SaleDetail MarketHotelCap Charges TelePhone parameter.
RC_SET_SD_HOTEL_CHARGES_BUSINESSCENTER	1227	Response refers to the SET SaleDetail MarketHotelCap Charges BusinessCenter parameter.
RC_SET_SD_HOTEL_CHARGES_PARKING	1228	Response refers to the SET SaleDetail MarketHotelCap Charges Parking parameter.
RC_SET_SD_HOTEL_CHARGES_MOVIE	1229	Response refers to the SET SaleDetail MarketHotelCap Charges Movie parameter.
RC_SET_SD_HOTEL_CHARGES_HEALTHCLUB	1230	Response refers to the SET SaleDetail MarketHotelCap Charges HealthClub parameter.
RC_SET_SD_HOTEL_CHARGES_GIFTSHOPPURCHASES	1231	Response refers to the SET SaleDetail MarketHotelCap Charges GiftShopPurchases parameter.
RC_SET_SD_HOTEL_CHARGES_FOLIOCASHADVANCES	1232	Response refers to the SET SaleDetail MarketHotelCap Charges FolioCashAdvances parameter.

RC_SET_SD_HOTEL_CHARGES_OTHER	1233	Response refers to the SET SaleDetail MarketHotelCap Charges Other parameter.
RC_SET_SD_HOTEL_CHARGES_TOTALTAXAMOUNT	1234	Response refers to the SET SaleDetail MarketHotelCap Charges TotalTaxAmount parameter.
RC_SET_SD_HOTEL_CHARGES_AUDITADJUSTMENT	1235	Response refers to the SET SaleDetail MarketHotelCap Charges AuditAdjustment parameter.
RC_SET_SD_MARKETTRANSPORTCAP	1236	Response refers to the SET SaleDetail MarketTransportCap parameter.
RC_SET_SD_TRANSPORT_PASSENGERNAME	1237	Response refers to the SET SaleDetail MarketTransportCap PassengerName parameter.
RC_SET_SD_TRANSPORT_DEPARTUREDATE	1238	Response refers to the SET SaleDetail MarketTransportCap DepartureDate parameter.
RC_SET_SD_TRANSPORT_ORIGCITYAIRPORT	1239	Response refers to the SET SaleDetail MarketTransportCap OrigCityAirport parameter.
RC_SET_SD_TRANSPORT_TRIPLEGSEQ	1240	Response refers to the SET SaleDetail MarketTransportCap TripLegSeq parameter.
RC_SET_SD_TRANSPORT_TRIPLEG	1241	Response refers to the SET SaleDetail MarketTransportCap TirpLeg parameter.
RC_SET_SD_TRANSPORT_TRIPLEG_DATEOFTRAVEL	1242	Response refers to the SET SaleDetail MarketTransportCap TripLeg DateOfTravel parameter.
RC_SET_SD_TRANSPORT_TRIPLEG_CARRIERCODE	1243	Response refers to the SET SaleDetail MarketTransportCap TripLeg CarrierCode parameter.

RC_SET_SD_TRANSPORT_TRIPLEG_SERVICECLASS	1244	Response refers to the SET SaleDetail MarketTransportCap TripLeg ServiceClass parameter.
RC_SET_SD_TRANSPORT_TRIPLEG_STOPOVERCODE	1245	Response refers to the SET SaleDetail MarketTransportCap TripLeg StopOverCode parameter.
RC_SET_SD_TRANSPORT_TRIPLEG_DESTCITYAIRPORT	1246	Response refers to the SET SaleDetail MarketTransportCap TripLeg DestCityAirport parameter.
RC_SET_SD_TRANSPORT_TRIPLEG_FAREBASISCODE	1247	Response refers to the SET SaleDetail MarketTransportCap TripLeg FareBasisCode parameter.
RC_SET_SD_TRANSPORT_TRIPLEG_DEPARTURETAX	1248	Response refers to the SET SaleDetail MarketTransportCap TripLeg DepartureTax parameter.
RC_SET_SD_TRANSPORT_TICKETNUMBER	1249	Response refers to the SET SaleDetail MarketTransportCap TicketNumber parameter.
RC_SET_SD_TRANSPORT_TRAVELAGENCYCODE	1250	Response refers to the SET SaleDetail MarketTransportCap TravelAgencyCode parameter.
RC_SET_SD_TRANSPORT_TRAVELAGENCYNAME	1251	Response refers to the SET SaleDetail MarketTransportCap TravelAgencyName parameter.
RC_SET_SD_TRANSPORT_RESTRICTIONS	1252	Response refers to the SET SaleDetail MarketTransportCap Restrictions parameter.
RC_SET_STORENUM	1300	Response refers to the SET StoreNum parameter.
RC_SET_CHAINNUM	1301	Response refers to the SET ChainNum parameter.

RC_SET_AGENTNUM	1302	Response refers to the SET AgentNum parameter.
RC_SET_TERMINALID	1303	Response refers to the SET TerminalId parameter.
RC_SET_WAKEUPMIMETYPE	1304	Response refers to the SET WakeupMimeType parameter.
RC_SET_HWCRYPTOID	1305	Response refers to the SET HwCryptoId parameter.
RC_SET_HWCRYPTOPWD	1306	Response refers to the SET HwCryptoPwd parameter.
RC_SET_CERTFLATFILEPATH	1307	Response refers to the SET CertFlatFilePath parameter.
RC_SET_SETFLATFILEPATH	1308	Response refers to the SET SETFlatFilePath parameter.
RC_SET_CERTPASSWORD	1309	Response refers to the SET CertPassword parameter.
RC_SET_GATEWAYHOSTNAME	1310	Response refers to the SET GatewayHostName parameter.
RC_SET_GATEWAYPORT	1311	Response refers to the SET GatewayPort parameter.
RC_SET_GATEWAYURI	1312	Response refers to the SET GatewayURI parameter.
RC_SET_SIGNINGBRANDID	1313	Response refers to the SET SigningBrandId parameter.
RC_SET_MAXCONNECTIONS	1314	Response refers to the SET MaxConnections parameter.
RC_SET_READTIMEOUT	1315	Response refers to the SET ReadTimeOut parameter.

RC_SET_MAXIMMEDIATERETRIES	1316	Response refers to the SET MaxImmediateRetries parameter.
RC_SET_DELAYEDRETRYINTERVAL	1317	Response refers to the SET DelayedRetryInterval parameter.
RC_SET_MAXDELAYEDRETRIES	1318	Response refers to the SET MaxDelayedRetries parameter.
RC_SET_BRANDID	1319	Response refers to the SET BrandId parameter.
RC_SET_BRANDURL	1320	Response refers to the SET BrandURL parameter.
RC_SET_BIN	1321	Response refers to the SET Bin parameter.
RC_SET_ACQUIRERBUSINESSID	1322	Response refers to the SET AcquirerBusinessId parameter.
RC_SET_MERCHANTID	1323	Response refers to the SET MerchantId parameter.
RC_SET_ACQUIRERSETPROFILE	1324	Response refers to the SET AcquirerSETProfile parameter.
RC_SET_PRESENTTOWALLETS	1325	Response refers to the $PRESENTTOWALLETS parameter.
RC_SET_BRAND_ADMIN	1330	Response refers to the SET Brand Admin parameter.
RC_SET_LANGUAGE	1350	Response refers to the SET Language parameter.
RC_SET_REG_FIELD_VALUE	1351	Response refers to the SET Registration Field Value parameter.
RC_SET_ACCOUNT_DATA	1352	Response refers to the SET Account Data parameter.
RC_SET_RECORD_ID	1353	Response refers to the SET Record Id parameter.

RC_SET_ROOT_HASH	1354	Response refers to the SET Root Hash parameter.
RC_SET_CAURL	1355	Response refers to the SET CAURL parameter.
RC_SET_ROOT_HASH_REQUIRED	1356	Root hash parameter is required.
RC_SET_REG_ANSWERS_REQUIRED	1357	Registration answer(s) parameter is required.
RC_SET_CERT_REQUEST_PENDED	1358	Certificate request is pending. Certificate may be obtained later.
RC_SET_EXISTING_CERTS_WILL_BE_USED	1359	There are existing certificates in the certificate database with the same credentials (brand, bin, merchantID) that are specified in the create brand request. The newly added brand will use the existing certificates.
RC_SET_RENEW_CERTS	1360	Response refers to the SET Renew Certs parameter.
RC_SET_ABORT	1361	Response refers to the SET Abort parameter.
RC_SET_CERT_RES_FAILURE	1362	SET certificate response message failed.
RC_SET_CERT_INIT_FAILURE	1363	SET certificate initialization failed.
RC_SET_CERT_REG_FORM_ERROR	1364	There is an error in SET certificate registration form.
RC_SET_RETRIEVE_GATEWAY_CERTS	1365	Response refers to the SET Retrieve Gateway Certs parameter.
RC_SET_OD	1400	Response refers to the SET OD parameter.
RC_SET_CHARSET	1401	Response refers to the SET CharSET parameter.
RC_SET_SUCCESSURL	1402	Response refers to the SET SuccessURL parameter.

RC_SET_FAILUREURL	1403	Response refers to the SET FailureURL parameter.
RC_SET_CANCELURL	1404	Response refers to the SET CancelURL parameter.
RC_SET_SERVICEURL	1405	Response refers to the SET ServiceURL parameter.
RC_SET_SPLITALLOWED	1406	Response refers to the SET SplitAllowed parameter.
RC_SET_CA_COMMUNICATION_ERROR	1407	An error occurred while communicating with SET CA.
RC_SET_CARDVERIFYCODE	1408	Response refers to the SET CardVerifyCode (CVV2, CVC2) parameter.
RC_SET_SAVECARDVERIFYCODE	1409	Response refers to the SET Save CardVerifyCode (CVV2.NOPERM) parameter.
RC_SET_PCERT_UNSPECIFIED_FAILURE	1410	Gateway certificate request (PCERT) failed due to an unspecified error.
RC_SET_PCERT_BRAND_FAILURE	1411	Gateway certificate request (PCERT) failed because of bad brand.
RC_SET_PCERT_BIN_FAILURE	1412	Gateway certificate request (PCERT) failed because of unknown BIN.
RC_SET_PCARD_PAYRECURINDICATOR	1413	Response refers to the purchase card data member PayRecurIndicator.
RC_SET_PCARD_SPLITALLOWED	1414	Response refers to the inability to specify both purchase card data and SplitAllowed equal to true at order creation time.
RC_SET_PCARD_WITH_SALEDETAIL	1415	Response refers to the inability to specify both purchase card data and sale detail at the same time.

Appendix G. Cassette for SET Messages

CEPSET0140 No certified brands are associated with the merchant number *merchant_number***.**

Severity: Error

Explanation: No certified brand has been associated with a Merchant. A RECEIVEPAYMENT cannot be completed for this Merchant.

User Response: Stop WebSphere Commerce Payments and configure at least one brand in the database for the specified Merchant. Note that the brand you configure should be included as an option for wallets in SET initiation messages.

CEPSET0141 No brands are associated with the merchant number *merchant_number***.**

Severity: Error

Explanation: No brand has been associated with a Merchant. WebSphere Commerce Payments has been terminated.

User Response: Either 1) configure at least one brand in the database for the specified Merchant, or 2) remove the Merchant from configuration.

CEPSET0155 An attempt was made to add to an unopened batch with BATCHID *batch_number* **for Merchant** *merchant number***, Acquirer** *acquirer_name***.**

Severity: Error

Explanation: The batch corresponding to the transaction item was not found.

User Response: The corresponding batch must be created before items can be added.

CEPSET0156 An attempt was made to set AcquirerSETPROFILE with an invalid profile number for Acquirer *acquirer_name***.**

Severity: Error

Explanation: There was an attempt to set an Acquirer's AcquirerSETPROFILE field with an invalid profile number.

User Response: Refer to the *WebSphere Commerce Payments Administrator's Guide* for a description of the Acquirer Profile number format.

CEPSET0157 An SQL exception was thrown while attempting to update the HAVECERT field in the Brand Configuration table. The SQL exception text describes the exception and provides state information that can be looked up in the XOPEN SQL specification. This could be the result of a disruption in communication between WebSphere Commerce Payments and the database server, or a discrepancy between the Brand Configuration table definition and the definition expected by WebSphere Commerce Payments.

Severity: Error

Explanation: An SQL exception occurred while attempting to log an order to the Order database. This could be due to an error connecting or writing to the database or due to an error in the data being stored.

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

CEPSET0158 An attempt was made to compose a PRES for an order that does not exist: *text***.**

Severity: Error

Explanation: An attempt was made to create a PRES message for an order that does not exist in the WebSphere Commerce Payments database.

User Response: An order must be created before a PRES message can be sent.

CEPSET0159 An item in batch *batch_number* **for Merchant** *merchant_number***, account** *account_number* **has no entry in the SET Batch Item table** *table_name***.**

Severity: Error

Explanation: An attempt was made to retrieve a SET batch item whose BATCHCONTENT record had no corresponding SETBATCHCONTENT record.

User Response: A new record must be created for the batch item in the SET Batch Item table.

CEPSET0160 SET batch *batch_number* **has no entry in the Generic Batch table** *table_name***.**

Severity: Error

Explanation: A SET batch is missing its corresponding record in the Generic Batch table.

User Response: A new record must be created for the batch in the Generic Batch table.

CEPSET0161 Failed to find an Acquirer for order *order_number*, **Merchant** *merchant_number*, **brand** *brand_name*.

Severity: Error

Explanation: No Acquirer is associated with the brand specified in the PINITREQ message for this Merchant.

User Response: Configure an Acquirer for this brand in the Brand Configuration table, ETBRANDCFG.

CEPSET0162 Received an unsuccessful response from the Payment Gateway for Merchant *merchant_number*, **order number** *order_number*: *text*.

Severity: Information

Explanation: The Payment Gateway failed to perform the requested action for the reason specified by the result code.

User Response: No response required.

CEPSET0163 Failed to find a brand profile for order *order_number*, **Merchant** *merchant_number*, **brand** *brand_name*.

Severity: Error

Explanation: No Acquirer is associated with the brand specified in the PINITREQ message for this Merchant.

User Response: Configure an Acquirer for this brand in the Brand Configuration table, ETBRANDCFG.

CEPSET0164 Received an unsuccessful response from the Payment Gateway for Merchant *merchant_number*, **batch** *batch_ID*: *text*.

Severity: Information

Explanation: The Payment Gateway failed to perform the requested action for the reason specified by the result code.

User Response: No response required.

CEPSET0165 Batch totals at Merchant do not agree with those at Acquirer for Merchant *merchant_number*, **batch** *batch_number*.

Severity: Error

Explanation: The batch totals at Merchant and Acquirer do not agree. The batch cannot be closed.

User Response: Contact financial institution to reconcile batch.

CEPSET0166 The acquirer responded with batchID *new_batchid* **for a transaction already bound to a batch with batchID** *old_batchid*.

Severity: Error

Explanation: The acquirer responded with a new batchID for a transaction already bound to a batch.

User Response: Contact financial institution

CEPSET0202 SQL exception logging Order *order_number* **for Merchant** *merchant_number* **:** *text*

Severity: Error

Explanation: An SQL exception occurred while attempting to log an order to the Order database. This could be due to an error connecting or writing to the database or due to an error in the data being stored.

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

CEPSET0203 An SQL exception occurred while updating order *order_number* **for Merchant** *merchant_number* **:** *text*

Severity: Error

Explanation: An SQL exception occurred while updating the state and transaction key of the order in the WebSphere Commerce Payments database. This could be due to an error connecting to or writing to the database or due to an error in the content of the data being stored.

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

CEPSET0204 An SQL exception occurred while reading Order database record for order number *order_number* **and Merchant** *merchant number*: *text*

Severity: Error

Explanation: An SQL Exception occurred while retrieving a record from the WebSphere Commerce Payments order database. This could be due to an error connecting to or reading from the database, or due to an error in the content of the data that was read from the database.

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

CEPSET0300 Write of *message* from Merchant *merchant_number* to Acquirer *acquirer_name* failed.

Severity: Error

Explanation: An I/O exception occurred writing the specified SET message to the specified Acquirer's Payment Gateway. This could be due to a network problem, a problem at the Payment Gateway, or a problem at WebSphere Commerce Payments. The WebSphere Commerce Payments won't retry this transaction again.

User Response: See corresponding message 0671 or 0625 for details.

CEPSET0301 Acquirer *acquirer_name* is not responding to Merchant *merchant_number*. Queuing the operation.

Severity: Information

Explanation: A read timeout occurred waiting to communicate with the Payment Gateway on the specified acquirer. This could be due to a network problem, a problem at the Payment Gateway, or a problem at WebSphere Commerce Payments. WebSphere Commerce Payments queued the operation to attempt at a later time.

User Response: Check that the network is functioning. Verify that the Acquirer software is functioning. If this happens frequently, consider increasing the read timeout interval. Specified in the Account Acquirer advanced settings UI panel, or the $READTIMEOUT parameter of the ModifyAccount API.

CEPSET0302 Unable to close connection on port *number* between Acquirer *acquirer_name* and Merchant *merchant_number*.

Severity: Error

Explanation: The socket connection that performs the communication between WebSphere Commerce Payments and the specified Acquirer failed to close successfully. This may be due to the connection already being closed, which can happen if there is more than one definition for this Acquirer/Payment Gateway in the ETACQCFG table in the configuration database.

User Response: Make sure there is only one entry for the Acquirer in ETACQCFG.

CEPSET0303 Unknown ComPoint type *type* for gateway *gateway_name* and gateway port *port_number*.

Severity: Warning

Explanation: In the ETACQCFG table, the specified Payment Gateway is configured to use a protocol that is not supported. The Payment Gateway for this Acquirer failed to be created. The supported Payment Gateway profiles are HTTP and TEST.

User Response: Change the GATEWAYPROTOCOL entry in the ETACQCFG to HTTP (note that this entry is case-sensitive).

CEPSET0304 PCERTREQ Response not received on port *port_number* by Merchant *merchant_name*, from Acquirer *acquirer_name*.

Severity: Warning

Explanation: A PCERTREQ has been sent to the Acquirer, via a Payment Gateway, but a PCERTRESPONSE was not received back from the Acquirer as expected.

User Response: Verify that the Acquirer is up and running. Verify that the Acquirer's and WebSphere Commerce Payments certificate and key file data match and that the WebSphere Commerce Payments configuration is correct.

CEPSET0305 COMPOSEPCERTREQ call failed from Merchant *merchant_name*, to Acquirer *acquirer_name* with return code *rc*

Severity: Error

Explanation: The SET Toolkit failed to compose a PCERTREQ for the Acquirer. This may occur:

- when the Certificate BCI expires.
- if the merchant certificate is not valid.
- if there is a mismatch between the SET Toolkit level, the key.db(s), and the WebSphere Commerce Payments facade library.

User Response:

- Verify that you have a merchant certificate by looking at the Brand details. At the bottom of the screen, there will be a checkmark for valid merchant and valid gateway certificates.
- Use CertUtil to dump the certificates to a file. The BCIs are listed at the end of the file. Check the validity dates. If they have expired, notify the Acquirer. To update the BCI, select Renew Certificate, and when the registration form appears, select Abort Certificate Request. Check the FAQs for more details instructions.
- If there is no mismatch between the SET Toolkit, the key.db(s), and the facade library, contact your service representative.

CEPSET0308 WebSphere Commerce Payments failed composing a message with return code *rc* when composing message type *type*

Severity: Error

Explanation: A SET message could not be composed to send to the Payment Gateway of the type specified.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET0314 A SET error message was received with error code *rc*.

Severity: Error

Explanation: The Cassette on the Cardholder or Payment Gateway detected an error in the content of a SET message. It replied by sending a SET error message indicating the condition that was detected. This problem occurred because the content of a SET message is in error. One possible cause for this is mismatching roots for key.db files.

User Response: This may be caused by incompatibility between the Cardholder's certificate and the Merchant's certificate. If this is not the case, contact your service representative.

CEPSET0318 An SQL exception occurred while updating ETSETMESSAGES with Merchant *merchant_name*, order number*number*, and split number*number*: *text*.

Severity: Error

Explanation: An SQL exception occurred while writing the current SETMessageRecord to the ETSETMESSAGES table. The SETMessageRecord keeps track of the state of a transaction by storing the SET state and message information. This exception may be due to a mismatch in the definition of the database table and the definition required by WebSphere Commerce Payments. It could also be due to a loss in the connection to the database due to a network problem or a failure at the database server.

User Response: Use the SQL exception information to determine the nature of the problem; follow the diagnostic information from the database documentation to resolve this problem.

CEPSET0319 An SQL exception occurred while retrieving next row of data from ETSETMESSAGES table:*text*.

Severity: Error

Explanation: An attempt to read a SETMessageRecord for a specific order failed because such a record does not exist in the ETSETMESSAGES table. The transaction key of the order is used to look up the SETMessageRecord. This means that a failure occurred processing the order or the processing of the order has not yet progressed to the point of logging a SETMessageRecord.

User Response: Retry the action that caused this message to occur; if this fails again, then the entire payment will have to be reprocessed starting with the RECEIVEPAYMENT.

CEPSET0320 WebSphere Commerce Payments SET transaction processing failed with return code *rc* when processing a message with input type *type*.

Severity: Error

Explanation: An error occurred while processing an incoming SET message from either the Payment Gateway or the Cardholder. The message type specified could not be decoded successfully by the SET Toolkit on the WebSphere Commerce Payments machine.

The input represents the type of message that was being processed—either a PaymentAPI command or a SET message signal.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET0321 Unknown action*action* received during SET transaction processing for Merchant *merchant_name*, order number *order_number*.

Severity: Error

Explanation: An action was generated by the Finite State Machine for transaction processing that is not a valid action.

User Response: This is an internal WebSphere Commerce Payments error; contact your service representative.

CEPSET0323 There was an error while composing a PINITRES message (return code = *rc*), for Merchant *merchant_number*, order number *order_number* .

Severity: Error

Explanation: A PINITRES SET message could not be composed to send to the Cardholder. An error response has been written to the Cardholder to indicate that an error has occurred. The ETSETMESSAGES database has been updated to indicate an internal SET failure occurred.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET0324 There was an error while composing a PRES SET message (return code = *rc*) for Merchant *merchant_number*, order number *order_number*.

Severity: Error

Explanation: A PRES SET message could not be composed to send to the Cardholder. An error response has been written to the Cardholder to indicate that an error has occurred. The ETSETMESSAGES table has been updated to indicate an internal SET failure occurred.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET0325 There was an error while composing an AUTHREQ SET message (return code = *rc*) for Merchant *merchant_number*, order number *order_number*

Severity: Error

Explanation: An AUTHREQ SET message could not be composed to send to the Payment Gateway. The ETSETMESSAGES table has been updated to indicate an internal SET failure occurred.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET0327 There was an error while composing an AUTHREVREQ SET message (return code = *rc*) for Merchant *merchant_number*, order number *order_number*.

Severity: Error

Explanation: An AUTHREVREQ SET message could not be composed to send to the Payment Gateway. The ETSETMESSAGES table has been updated to indicate an internal SET failure occurred.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET0328 There was an error while composing a SET error message (return code = *rc*) for Merchant *merchant_number*, order number *order_number*.

Severity: Error

Explanation: A SET error message could not be composed to send to the Payment Gateway. The ETSETMESSAGES table has been updated to indicate that an internal SET failure occurred.

User Response: This is an internal error returned by the SET Toolkit. Contact your service representative.

CEPSET0329 WebSphere Commerce Payments SET transaction processing failed with return code *rc* when processing a message for Merchant *merchant_number*.

Severity: Error

Explanation: An error occurred while processing an incoming SET message from the Cardholder. The message could not be processed successfully by the SET Toolkit on the WebSphere Commerce Payments machine.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET0331 The retrieve of a SETMessageRecord for an APPROVE API with Merchant *merchant_number*, order number *order_number* failed.

Severity: Error

Explanation: An APPROVE API command was issued, but there is no corresponding transaction in the appropriate state.

User Response: Check that you have entered the correct merchant number and order number for the APPROVE API. If these values are correct, verify that this transaction is in the correct state for an APPROVE.

CEPSET0332 SET transaction processing entered an incorrect state for Merchant*merchant_number*, order number *order_number*.

Severity: Error

Explanation: An incorrect state was generated by the Finite State Machine for SET transaction processing.

User Response: This is an internal WebSphere Commerce Payments error; contact your service representative.

CEPSET0341 An IO exception occurred while building the SET Payment Initiation message:*message*. Message truncated at *number* bytes.

Severity: Error

Explanation: An SQL exception occurred while building the SET Payment Initiation (Wake up) message to be sent to the Merchant Server. The wake up message length at the time the error occurred is indicated in the message. This is a failure of the Java Virtual Machine (JVM).

User Response: Stop and restart WebSphere Commerce Payments.

CEPSET0351 WebSphere Commerce Payments received a null authorization response from the Acquirer for Merchant *merchant_number***, order number** *order_number***.**

Severity: Error

Explanation: WebSphere Commerce Payments attempted to send an authorization or authorization reversal request to an Acquirer and the response received back was null, meaning the MIME header contained no message.

User Response: Check to see if there was trace back information previous to this error. It may display a return code from the Acquirer. In this case, the Acquirer is running and received the authorization request, but returned a null response. You should also check any error data that may have been produced by the Acquirer.

CEPSET0352 WebSphere Commerce Payments SET database pruning of the SETFacade *record_number***failed with return code** *rc***.**

Severity: Error

Explanation: WebSphere Commerce Payments attempted to prune a SET database record and failed.

User Response: Make sure the database environment is configured properly.

CEPSET0356 The PCERT message did not complete successfully.

Severity: Error

Explanation: The data in the Merchant's certificate does not agree with the data in the Payment Gateway certificate.

User Response: Check the Acquirer Settings. Also check if your acquirer has off days or has a downtime. If none of the prior mentioned reasons are applicable then it may be internal error from SET ToolKit. Contact your service representative.

CEPSET0357 The Cassette failed to create and open a ServerSocket to listen for payment or inquiry messages.

Severity: Error

Explanation: At startup, the Cassette will attempt to open two ServerSockets, one to listen for payment messages from the Cardholder, the other to listen for inquiry messages from Cardholder. One or both of these has failed to open.

User Response: Check the SET Profile configuration values for PAYMENTCGIPORT and INQUIRYPORT.

CEPSET0358 Order *order_number***, for Merchant** *merchant_number* **has been rejected.**

Severity: Error

Explanation: An order was rejected by WebSphere Commerce Payments. Most likely the order description was corrupted.

User Response: The user must ensure that the order description received on the PINITREQ is the same that has been returned on in the wake up message.

CEPSET0359 The Acquirer sent an unexpected message to WebSphere Commerce Payments for Merchant *merchant_number***, order number** *order_number***.**

Severity: Error

Explanation: WebSphere Commerce Payments attempted to send a request to an Acquirer, and the response received back from the Acquirer was unexpected.

User Response: Check to see if there was trace-back information previous to this error; it may display a return code from the Acquirer. You also should check any error data that may have been produced by the Acquirer.

CEPSET0360 Failed to process a response message from acquirer to WebSphere Commerce Payments for Merchant *merchant_number***, order number** *order_number***. A SET Error Message was sent.**

Severity: Error

Explanation: WebSphere Commerce Payments attempted to process a response from an Acquirer and failed.

User Response: Check to see if there was trace-back information previous to this error; it may display a return code generated during processing. If possible check SET dump and trace files for details of the failure.

CEPSET0400 An SQL exception occurred while reading the Acquirer Profile table *table_name***:** *text***.**

Severity: Error

Explanation: An SQL exception has occurred writing an Acquirer Profile to the Acquirer database table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the Acquirer Profile table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0401 An SQL exception occurred while reading an Acquirer Profile from *table_name*:*text*.

Severity: Error

Explanation: An SQL exception occurred while retrieving the Acquirer Configuration table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0402 Either the MERCHANTNAME *merchant_number* **or the ACQUIRERBIN** *BIN* **is null in an Acquirer Profile configuration record.**

Severity: Error

Explanation: An attempt was made to update, add, or delete an ETACQCFG table record when the MERCHANTNAME field was null. This field must be set since it is part of the SQL database key for the Acquirer Configuration table.

User Response: Make sure the MERCHANTNAME is set in the ETACQCFG table. If it is initialized, contact your service representative.

CEPSET0403 An SQL exception occurred while writing an Acquirer Profile record to the Acquirer Configuration table: *text*.

Severity: Error

Explanation: An SQL exception occurred while writing ETACQCFG fields to an SQL prepared statement. The SQL Exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a database field versus the data type of the Acquirer Profile.

User Response: Make sure the database field was initialized to the correct data type. If it was done correctly, contact your service representative.

CEPSET0404 A non-numeric ACQUIRERBIN value *value_number* **was found in the ETACQCFG table.**

Severity: Error

Explanation: A value was specified for the ACQUIRERBIN field of an Acquirer Profile that is not a numeric string. The ACQUIRERBIN field is a 1 to 6 digit numeric string.

User Response: Correct the value in the Acquirer configuration panel.

CEPSET0405 The MERCHANTNAME value, *merchant_number*, **should be a numeric string of one to nine digits.**

Severity: Error

Explanation: A value was specified for the MERCHANTNAME field of an AcquirerProfile that is not a numeric string. The MERCHANTNAME field is a 1 to 9 digit numeric string.

User Response: Correct the value of MERCHANTNAME on the Acquirer Configuration table (ETACQCFG).

CEPSET0406 SETProfile configuration error, payment port *port_number* **equals inquiry port** *port_number*.

Severity: Error

Explanation: The configured values for the PAYMENTPORT and INQUIRYPORT of the SET Configuration table are equal. The PAYMENTPORT and INQUIRYPORT numbers must be different. They also must not equal any of the ports used by WebSphere Commerce Payments, such as the Payment API Port.

User Response: Change the value of one or both of the ports defined in ETSETCFG so that they are unique

---

CEPSET0407 An SQL exception occurred while reading the ETSETCFG table: *text*

Severity: Error

Explanation: An SQL exception has occurred writing a record to the SET Configuration table (ETSETCFG). The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the SET database table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0408 An SQL Exception occurred while reading the SET Configuration table ETSETCFG: *text*

Severity: Error

Explanation: An SQL exception occurred while retrieving the SET Configuration table (ETSETCFG). The SQL exception text describes the exception and

provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0409 The PAYMENTSYSTEMNAME field of the SET Profile record is not set.

Severity: Error

Explanation: The PAYMENTSYSTEMNAME field of a SET Configuration record was not set. This field must have a value since it is the SQL table key.

User Response: Check to make sure the PAYMENTSYSTEMNAME field in the ETSETCFG table is initialized with the value 'SET'. If it is, contact your service representative.

CEPSET0410 An SQL exception occurred while attempting to update the SET Configuration table ETSETCFG: *text***.**

Severity: Error

Explanation: An SQL exception has occurred writing SETProfile fields to an SQL PreparedStatement. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a SET Configuration field versus the data type of the corresponding field in the SETProfile.

User Response: Make sure the fields are initialized with the correct data types. If they are, contact your service representative.

CEPSET0416 An SQL Exception occurred while reading the Brand Profile table *table_name***:** *text***.**

Severity: Error

Explanation: An SQL exception has occurred reading a BrandProfile to the Brand Configuration table (ETBRANDCFG). The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the Brand Configuration table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0417 An SQL exception occurred while reading from the Batch Configuration table *table_name***:** *text***.**

Severity: Error

Explanation: An SQL exception occurred while retrieving the Brand Configuration table. The SQL Exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0418 Either the MERCHANTNAME *merchant_number***, ACQUIRERBIN** *BIN***, or BRANDID** *ID* **is null.**

Severity: Error

Explanation: Either the MERCHANTNAME, ACQUIRERBIN, or BRANDID is null. All of them must be set, since these fields are the SQL key for a Brand Configuration table record.

User Response: Verify that each of these fields is filled in on the WebSphere Commerce Payments Brand Configuration table (ETBRANDCFG). If they are and this error occurs, contact your service representative.

CEPSET0419 An SQL exception occurred while attempting to update the Brand Configuration table ETBRANDCFG *text***.**

Severity: Error

Explanation: An SQL exception occurred writing a Brand Profile into an SQL PreparedStatement. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a Brand Configuration field versus the data type of the corresponding field in the Brand Profile. It can also be caused if a required key field is not initialized.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Contact your service representative.

CEPSET0420 An SQL exception occurred while reading from *table_name***:** *text***.**

Severity: Error

Explanation: An SQL exception occurred while reading the ACQUIREROFFDAYS table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a

mismatch in the AcquirerOffDays Configuration table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0421 An SQL exception occurred while reading from *table_name*: *text*.

Severity: Error

Explanation: An SQL exception occurred while reading the ACQUIREROFFDAYS table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0423 An SQL exception occurred while attempting to update the ACQUIREROFFDAYS table: *text*.

Severity: Error

Explanation: An SQL exception occurred writing the values of an AcquirerOffDays into an SQL PreparedStatement. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a AcquirerOffDays Configuration field versus the data type of the corresponding field in the AcquirerOffDays Profile (ETACQOFFDAYS). It can also be caused if a required key field is not initialized.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Contact your service representative.

CEPSET0425 Invalid month specified in the ACQUIREROFFDAYS table.*value* specified; must be 1-12 or 0 (if year is also 0).

Severity: Error

Explanation: The value specified for the month of an ACQUIREROFFDAYS Configuration record was not valid. The value specified for month may be 0 only if the value configured for the year parameter is also 0. If both the year and month are 0, then the value configured for day represents the day of the week. Otherwise, the value must be between 1 and 12, inclusive, corresponding to the months of the calendar.

User Response: Correct the value of the month configured for this record in the ETACQOFFDAYS table.

CEPSET0426 Invalid day *value* specified in ACQUIREROFFDAYS; must not exceed the number of days *value* in the month *value*.

Severity: Error

Explanation: The value specified for day exceeds the number of days in that month. The value of day specified must not be greater than the number of days in that month.

User Response: Correct the value of day configured for this record in the ETACQOFFDAYS table.

CEPSET0427 Invalid day of the week *value* specified in ACQUIREROFFDAYS; must not exceed the number of days in a week, *value*.

Severity: Error

Explanation: The value specified for day corresponds to a day in the week since the values of month and year are both 0. The value of day must be between 1 and 7, inclusive.

User Response: Correct the value of day configured for this record in the ETACQOFFDAYS table.

CEPSET0432 Signing brand *brand_name* specified in the Acquirer Profile Configuration record for Merchant *merchant_number*, AcquirerBIN *AcquirerBIN_number* was not found among Brand Profiles.

Severity: Error

Explanation: The signing brand must be one of the brands supported. The PCERT message was skipped.

User Response: Correct the value of SIGNINGBRANDID in the Acquirer Profile (ETACQCFG), or create a new brand profile for the signing brand.

CEPSET0433 The SET Database Type was not initialized properly. The value processed was *value*.

Severity: Error

Explanation: The SET database type was not initialized properly. The acceptable values are ODBC or flat file.

User Response: Change the value of the SETDATABASETYPE in ETSETCFG to one of the acceptable values.

CEPSET0434 The Certificate Database Type was not initialized properly. The value processed was *value*.

Severity: Error

Explanation: The Certificate Database Type was not initialized properly. The acceptable value is ODBC or flat file.

User Response: Change the value of the CERTDATABASETYPE to one of the acceptable values.

CEPSET0439 The SET Database Name was not initialized.

Severity: Error

Explanation: The SET database name was not initialized.

User Response: Provide the SET Database Owner via the SETDBNAME field in ETSETCFG.

CEPSET0454 Hardware Cryptography User ID was not initialized.

Severity: Error

Explanation: The Hardware Cryptography User ID was not initialized.

User Response: Provide the Hardware Cryptography User ID via the HWCRYPTOID field in ETSETCFG.

CEPSET0455 Hardware Cryptography Password was not initialized.

Severity: Error

Explanation: The Hardware Cryptography Password was not initialized.

User Response: Provide the Hardware Cryptography password via the HWCRYPTOPASSWORD input parameter.

CEPSET0456 Flatfile is the only supported value for the SETDBTYPE field in ETSETCFG.

Severity: Error

Explanation: Flatfile is the only supported value for the SETDBTYPE fields in the SET Configuration profile (ETSETCFG).

User Response: Stop WebSphere Commerce Payments. Change the values of these fields in the table to 'flatfile' and restart.

CEPSET0472 The required parameter *number* was not specified in an ACCEPTPAYMENT API command.

Severity: Error

Explanation: The protocol data structure passed through the ACCEPTPAYMENT command is missing a required field.

User Response: Add the required field and reissue the command. See the *WebSphere Commerce Payments Programmer's Guide and Reference* for a listing of required fields for the ACCEPTPAYMENT command.

CEPSET0473 The Protocol Data PAN parameter *number* does not have a valid length.

Severity: Error

Explanation: The parameter length is either too long or too short.

User Response: Correct the value specified for the parameter.

CEPSET0474 The Protocol Data EXPIRY parameter *number* does not have a valid length.

Severity: Error

Explanation: The parameter length is either too long or too short.

User Response: Correct the value specified for the parameter.

CEPSET0475 The Protocol Data EXPIRY parameter does not have the required format.

Severity: Error

Explanation: The Protocol Data EXPIRY parameter must be of the form YYYYMM.

User Response: Correct the value specified for the parameter.

CEPSET0476 The Protocol Data PAN parameter does not have the required format.

Severity: Error

Explanation: The Protocol Data PAN parameter must be a positive numeric string.

User Response: Correct the value specified for the parameter.

CEPSET0477 The Protocol Data BRANDID parameter does not have a valid length.

Severity: Error

Explanation: The parameter length is either too long or too short.

User Response: Correct the value specified for the parameter.

CEPSET0478 The Account Number value *value_number* should be a positive numeric string.

Severity: Error

Explanation: A value was specified for account number that was not a positive numeric string. The Account Number is a 1 to 9 digit positive numeric string.

User Response: Correct the value specified for an Account number.

CEPSET0479 The Acquirer Business ID value *value_number* should be a positive numeric string.

Severity: Error

Explanation: A value was specified for Acquirer Business ID that was not a positive numeric string. The Acquirer business ID is a 1 to 32 digit positive numeric string.

User Response: Correct the value specified for an Acquirer business ID.

CEPSET0612 Internal error occurred.

Severity: Error

Explanation: An internal error occurred in WebSphere Commerce Payments.

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

CEPSET0617 IOException opening connection to host name *hostname*, port *port_number* for ComPoint.

Severity: Error

Explanation: An I/O exception occurred when WebSphere Commerce Payments attempted to open a socket to the host name specified using the port number indicated. This could be because the port is already in use, because there was a network error, or because the host name indicated either is not valid or is not responding.

User Response: Test the connection to the host name specified. Verify that the specified host is still up and running.

CEPSET0622 HTTP response code *code_number* received was not a valid HTTP: error code is *code_number*.

Severity: Warning

Explanation:

- Code of 1 means there was no end-of-line character in the HTTP.
- Code of 2 means the entire Mime header was not received.
- Code of 3 means the HTTP command was not POST.
- Code of 4 means that it was not an HTTP header; HTTP version not in header. This could be due to an error or disruption in the communication between WebSphere Commerce Payments and the Cardholder or an error in the content of the data being sent by the Cardholder.

User Response: This is a problem with the data received from a Cardholder. If the problem persists, determine the provider of the Cardholder and contact the Service provider for this Cardholder, or instruct the user of this Cardholder to do so.

CEPSET0623 Merchant name *merchant_name* received in a SET message on *compoint* was not a valid merchant name.

Severity: Warning

Explanation: The HTTP sent by the Cardholder to WebSphere Commerce Payments did not contain a valid merchant number in the HTTP that it sent. The merchant number is supposed to be a numeric string of 1 to 9 digits. The merchant number that is supposed to appear in the SET message is sent in the Payment Initiation message sent by the Cassette of WebSphere Commerce Payments.

User Response: This is a problem with the data received from a Cardholder. If the problem persists, determine the provider of the Cardholder and contact the service provider for this Cardholder or instruct the user of this Cardholder to do so.

CEPSET0624 Content length *length* in Mime header received on *message* was not a valid length.

Severity: Error

Explanation: The content length in the SET protocol message received from the Cardholder was not a valid length.

User Response: This is a problem with the data received from a Cardholder. If the problem persists,

determine the provider of the Cardholder and contact the service provider for this Cardholder or instruct the Cardholder to do so.

CEPSET0625 Received HTTP Response code *code_name* from Payment Gateway.

Severity: Error

Explanation: Communication with the Payment Gateway via HTTP protocol failed. The message returns one of the response codes defined in RFC 1945. Possible errors include:

- (200) OK
- (201) Created
- (202) Accepted
- (204) No Content
- (301) Moved Permanently
- (302) Moved Temporarily
- (400) Bad Request
- (401) Unauthorized
- (403) Forbidden
- (500) Internal Server Error
- (501) Not Implemented
- (503) Service Unavailable

The actions of WebSphere Commerce Payments as a result of this message depend on the response code. A code of 200 indicates that the communication was successful. If the response code is greater than 200 and less than 400, the SET message will not be retried. For codes 400 or greater, WebSphere Commerce Payments will retry the communication, depending on how the retry values are defined in the Acquirer configuration profile (ETACQCFG).

User Response: Verify that the communication between WebSphere Commerce Payments and the Payment Gateway is okay. Verify that the version of WebSphere Commerce Payments is compatible with the version of the Payment Gateway, and that you have the correct key.db files. Contact your service representative.

CEPSET0626 An IOException occurred while attempting to write a SET message to the Payment Gateway on socket *socket*: *text*.

Severity: Error

Explanation: A Java I/O exception was generated during a write to the Payment Gateway.

User Response: If WebSphere Commerce Payments has been configured to retry failed operations to this Gateway, the operation will retry automatically, doing nothing. If all retries fail, or if no retries are configured, WebSphere Commerce Payments will put the value *true* in the IDEMPOTENCYFAILURE field of the message being written. The operation can be reissued.

CEPSET0628 Improperly formatted SET protocol message was received.

Severity: Error

Explanation: The received message was not a SET message, although the MIME header indicated that it was.

User Response: This is a problem with the format of the SET message. Either a problem exists on the originating system (likely if a small number of these messages appear), or this may be an attack on the security of your Merchant Server (likely if a very large number of these messages appear). If the originating system can be determined, contact the appropriate party to inform them of the problem. If some type of attack is suspected, invoke local procedures for handling such a situation.

CEPSET0706 The SET API did not initialize.

Severity: Severe

Explanation: The SET API failed to initialize.

User Response: Check previous error messages for clarification. The problem could be that the key.db, keypair.db, bci.db, or cri.db files are not in the specified CMS path, or don't have write access. Make sure that the field SETDBTYPE in the ETSETCFG configuration table is set to ″flatfile″. CMS database password may be incorrect, or CMS database may be corrupted. If the 4758 Hardware Cryptographic Coprocessor is enabled, the hardware may have failed, or the software installed for the hardware may be misconfigured.

CEPSET0707 Certificate password was incorrect, or CMS database was corrupted.

Severity: Fatal Error

Explanation: The CMS certificate encryption password may be incorrect, or the CMS database may be corrupted.

User Response: Check the correctness of the CMS certificate encryption password. Note that the CMS password is case-sensitive. If the password is correct, check the CMS database, which may be corrupted.

CEPSET0708 An error occurred, with return code *rc* while attempting to change the cryptography password.

Severity: Fatal Error

Explanation: The CMS certificate database password was not changed. may be corrupted.

User Response: If the certificate database was backed up before attempting this operation, then restore

the certificate database back to the saved copy. Contact your service representative.

CEPSET0716 Attempted an illegal partial refund reversal for Merchant *merchant_number*, **order** *order_number*, **and credit** *credit_number*.

Severity: Error

Explanation: The Merchant attempted to perform a refund reversal with a non-zero amount. The Cassette does not support partial refund reversals.

User Response: Perform only complete refund reversals.

CEPSET0717 Currency mismatch when attempting to add a payment/credit to a batch.

Severity: Error

Explanation: The Merchant attempted to add a payment or credit to a batch with a Currency or an AmountExp10 that does not match the Currency and AmountExp10 of that batch.

User Response: Use only one currency and AmountExp10 per batch.

CEPSET0718 Batch *batch_number* **unknown or not in proper state for depositing a payment.**

Severity: Error

Explanation: The Merchant attempted to automatically authorize and deposit a payment using a Batch ID that is either undefined or in a state other than Open. A batch must be in Open state for payments or credits to be added to it.

User Response: Ensure the batch ID refers to a batch in Open state.

CEPSET0719 AmountExp10 mismatch when attempting to add a payment/credit to a batch.

Severity: Error

Explanation: The Merchant attempted to add a payment/credit to a batch with a Currency or an AmountExp10 that does not match the Currency/AmountExp10 of that batch.

User Response: Use only one Currency and AmountExp10 in a single batch.

CEPSET0720 An implicit batch operation failed because *num_batches* **are open for merchant** *merchant_name*, **account** *account_number*.

Severity: Error

Explanation: The merchant issued a Deposit or Refund command without specifying a batch number against an account with several open batches. Unless the merchant explicitly specifies a batch number on each Deposit or Refund command, an account can have at most one open batch.

User Response: Ensure that only one batch is Open for this account or explicitly specify a batchNumber.

CEPSET0725 Brand *brand_name* **is not available for the Merchant** *merchant_number*.

Severity: Error

Explanation: This is not a certified brand associated with this Merchant. An ACCEPTPAYMENT cannot be completed for this Merchant.

User Response: Use a certified brand for this order.

CEPSET0726 WebSphere Commerce Payments SET transaction failed composing a MOPREQ message with return code *code_number* **for Merchant** *merchant_number*, **and order** *order_number*.

Severity: Error

Explanation: A MOPREQ SET message could not be composed.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET0727 Attempted an illegal partial Deposit Reversal for Merchant *merchant_number*, **order** *order_number*, **payment** *payment_number*, **and account number** *account_number*.

Severity: Error

Explanation: The Merchant attempted to perform a Deposit Reversal with a non-zero amount. The Acquirer specified does not support partial Deposit Reversals.

User Response: Perform only complete deposit reversals.

CEPSET0728 WebSphere Commerce Payments failed composing a MOPREQ message for Merchant *merchant_number* **and order number** *order_number* **since brand** *brand_name* **does not allow certless purchases.**

Severity: Error

Explanation: The Acquirer certificate for this brand does not permit purchases without a Cardholder certificate. The ETACCEPTPAYMENT() API command cannot be used with this brand.

User Response: Contact the acquiring institution to obtain certificates that support certless purchases or use a different brand that supports such purchases.

CEPSET0729 The Cassette has stopped.

Severity: Information

Explanation: The Cassette is no longer accepting requests.

User Response: If necessary, restart the cassette.

CEPSET0750 WebSphere Commerce Payments did not validate certificate integrity.

Severity: Information

Explanation: WebSphere Commerce Payments performs certificate validation to ensure that the values in the configuration tables are consistent with Merchant and Acquirer certificates. However, the format of at least one certificate in the database was unexpected and prevented validation.

User Response: No response required.

CEPSET0751 WebSphere Commerce Payments could not locate Merchant certificates for Merchant *merchant_number* **and brand** *brand_name***.**

Severity: Error

Explanation: WebSphere Commerce Payments could not locate either of the two required Merchant certificates (that is, signature and encryption) for the brand indicated. This could be due to a missing certificate or a mismatch of either MERCHANTID or BIN in the ETBRANDCFG table.

User Response: Either remove the brand from the ETBRANDCFG configuration table or ensure that certificates exist in the database that match the brand.

CEPSET0752 WebSphere Commerce Payments could not locate a Merchant signature certificate for Merchant*merchant_number* **and brand** *brand_name***.**

Severity: Error

Explanation: WebSphere Commerce Payments could not locate the required Merchant signature certificate for the brand indicated. This could be due to a missing certificate or a mismatch of either MERCHANTID or BIN in the ETBRANDCFG table.

User Response: Either remove the brand from the ETBRANDCFG configuration table or ensure that certificates exist in the database that match the brand.

CEPSET0753 WebSphere Commerce Payments could not locate a Merchant encryption certificate for Merchant *merchant_number* **and brand** *brand_name***.**

Severity: Error

Explanation: WebSphere Commerce Payments could not locate the required Merchant encryption certificate for the brand indicated. This could be due to a missing certificate or a mismatch of either merchantID or BIN in the ETBRANDCFG table.

User Response: Either remove the brand from the ETBRANDCFG configuration table or ensure that certificates exist in the database that match the brand.

CEPSET0754 IOException received while attempting to read a message from a Cardholder on socket *socket***: text**

Severity: Information

Explanation: A Java IOException was generated during a read from a Cardholder.

User Response: No response required.

CEPSET0755 The Cassette has started

Severity: Information

Explanation: The Cassette is now accepting requests

User Response: none

CEPSET0756 The value specified for protocol data *pd* **is too short.**

Severity: Error

Explanation: The protocol data received is too short.

User Response: Re-specify the protocol data, passing in a value with a valid length.

CEPSET0757 The value specified for protocol data *pd* is too long.

Severity: Error

Explanation: The protocol data received is too long.

User Response: Re-specify the protocol data, passing in a value with a valid length.

CEPSET0758 A value is required for protocol data *pd*.

Severity: Error

Explanation: No protocol data was received for a required parameter.

User Response: Specify the required protocol data.

CEPSET0759 The specified Acquirer SET Profile value: *value* is not valid.

Severity: Error

Explanation: The value specified for Acquirer SET Profile is not valid.

User Response: Enter a valid Acquirer SET Profile.

CEPSET0760 Restarted *number* transactions for WebSphere Commerce Payments *hostname*.

Severity: Information

Explanation: The specified number of pending transactions were sent to the Acquirer for WebSphere Commerce Payments *hostname*.

User Response: None.

CEPSET0800 Failed to load a SET extension *name*.

Severity: Error

Explanation: The Cassette failed to load a SET extension module.

User Response: follow the diagnostic information from the documentation for the extension module to resolve this problem.

CEPSET0801 Failed to load a negotiator for a SET extension *name*.

Severity: Error

Explanation: The Cassette failed to load a negotiator module for a SET extension.

User Response: follow the diagnostic information from the documentation for the extension module to resolve this problem.

CEPSET0805 An SQL exception occurred while reading the JPO Configuration Profile table *table_name*: *text*.

Severity: Error

Explanation: An SQL exception has occurred reading an JPO Configuration table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the JPO Configuration Profile table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0806 An SQL exception occurred while reading an JPO Configuration Profile from *table_name*:*text*.

Severity: Error

Explanation: An SQL exception occurred while retrieving the JPO Configuration table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0807 An SQL exception occurred while writing an JPO Configuration Profile record to the JPO Configuration table: *text*.

Severity: Error

Explanation: An SQL exception occurred while writing ETJPOCONFIG fields to an SQL prepared statement. The SQL Exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a database field versus the data type of the Acquirer Profile.

User Response: Make sure the database field was initialized to the correct data type. If it was done correctly, contact your service representative.

CEPSET0811 SET extension JPO error: *text*.

Severity: Error

Explanation: An exception occurred while processing JPO extension. The exception text describes the exception that can be looked up in the JPO documentation.

User Response: follow the diagnostic information from the JPO documentation to resolve this problem. Refer to JPO manual to get specific details about the problem.

CEPSET0815 An SQL exception occurred while reading the Extension Profile table *table_name*: *text*.

Severity: Error

Explanation: An SQL exception has occurred reading an Extension Configuration table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the Extension Profile table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0816 An SQL exception occurred while reading an Extension Profile from *table_name*:*text*.

Severity: Error

Explanation: An SQL exception occurred while retrieving the Extension Configuration table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0817 The EXTENSIONNAME *extension_name* **is null in an Extension configuration record.**

Severity: Error

Explanation: An attempt was made to update, add, or delete an ETEXTENSIONCFG table record when the EXTENSIONNAME field was null. This field must be set since it is part of the SQL database key for the Extension Configuration table.

User Response: Make sure the EXTENSIONNAME is set in the ETEXTENSIONCFG table. If it is initialized, contact your service representative.

CEPSET0818 An SQL exception occurred while writing an Extension Profile record to the Extension Configuration table: *text*.

Severity: Error

Explanation: An SQL exception occurred while writing ETEXTENSIONCFG fields to an SQL prepared statement. The SQL Exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a database field versus the data type of the Acquirer Profile.

User Response: Make sure the database field was initialized to the correct data type. If it was done correctly, contact your service representative.

CEPSET0820 An SQL exception occurred while reading the Multi-Acquirer Profile table *table_name*: *text*.

Severity: Error

Explanation: An SQL exception has occurred reading a Multi-Acquirer Configuration table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the Multi-Acquirer Profile table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0821 An SQL exception occurred while reading a Multi-Acquirer Profile from *table_name*:*text*.

Severity: Error

Explanation: An SQL exception occurred while retrieving the Multi-Acquirer Configuration table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0822 The MERCHANTNAME *merchant_number* **is null in a Multi-Acquirer Profile configuration record.**

Severity: Error

Explanation: An attempt was made to update, add, or delete an ETCHOOSEACQCFG table record when one of the fields was null. This field must be set since it is part of the SQL database key for the Multi-Acquirer Configuration table.

User Response: Make sure all fields are set in the

ETCHOOSEACQCFG table. If it is initialized, contact your service representative.

CEPSET0823 An SQL exception occurred while writing a Multi-Acquirer Profile record to the Multi-Acquirer Configuration table: *text*.

Severity: Error

Explanation: An SQL exception occurred while writing ETCHOOSEACQCFG fields to an SQL prepared statement. The SQL Exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a database field versus the data type of the Acquirer Profile.

User Response: Make sure the database field was initialized to the correct data type. If it was done correctly, contact your service representative.

CEPSET0825 An SQL exception occurred while reading the JPO data table *table_name*: *text*.

Severity: Error

Explanation: An SQL exception has occurred reading an JPO data table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the JPO data table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0826 An SQL exception occurred while reading an JPO data from *table_name*:*text*.

Severity: Error

Explanation: An SQL exception occurred while retrieving the JPO data table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0840 An SQL exception occurred while reading the JPO Condition table ETJPOCOND : *text*.

Severity: Error

Explanation: An SQL exception has occurred reading JPO Condition table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the JPO Condition table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0841 An SQL exception occurred while reading an JPO Condition table from ETJPOCOND:*text*.

Severity: Error

Explanation: An SQL exception occurred while retrieving the JPO Condition table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0842 An SQL exception occurred while writing SETJpoCondition record to the JPO Condition table ETJPOCOND: *text*.

Severity: Error

Explanation: An SQL exception occurred while writing SETJpoCondition fields to an SQL prepared statement. The SQL Exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a database field versus the data type of the SETJpoCondition.

User Response: Make sure the database field was initialized to the correct data type. If it was done correctly, contact your service representative.

CEPSET0843 An SQL exception occurred while reading the JPO Transaction Condition table ETJPOTRCOND: *text*.

Severity: Error

Explanation: An SQL exception has occurred reading JPO Transaction Condition table. The SQL exception text describes the exception and provides SQL state

information that can be looked up in the XOPEN SQL specification. This is likely due to a mismatch in the JPO Transaction Condition table definition and the one expected by WebSphere Commerce Payments; the data type of one of the fields that was retrieved from the database row did not match what was expected.

User Response: Refer to the SQL state information to get specific details about the nature of the problem.

CEPSET0844 An SQL exception occurred while reading an JPO Transaction Condition table from ETJPOTRCOND: *text*.

Severity: Error

Explanation: An SQL exception occurred while retrieving the JPO Transaction Condition table. The SQL exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is likely due to an error or disruption in the communication between WebSphere Commerce Payments and the database server.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Verify that the connection to the database server is OK.

CEPSET0845 An SQL exception occurred while writing SETJpoTRCondition record to the JPO Transaction Condition table ETJPOTRCOND: *text*.

Severity: Error

Explanation: An SQL exception occurred while writing SETJpoTRCondition fields to an SQL prepared statement. The SQL Exception text describes the exception and provides SQL state information that can be looked up in the XOPEN SQL specification. This is caused by a mismatch between the data type of a database field versus the data type of the SETJpoTRCondition.

User Response: Make sure the database field was initialized to the correct data type. If it was done correctly, contact your service representative.

CEPSET1000 The Cassette does not support the CASSETTECONTROL command *command_name*.

Severity: Error

Explanation: The CASSETTECONTROL command issued was invalid and ignored.

User Response: Consult the Cassette supplement for valid syntax of the CASSETTECONTROL command.

CEPSET1300 Unknown action*action* **received during SET transaction processing for Merchant** *merchant_name*, **brand** *brand_name*.

Severity: Error

Explanation: An action was generated by the Finite State Machine for transaction processing that is not a valid action.

User Response: This is an internal WebSphere Commerce Payments error; contact your service representative.

CEPSET1302 WebSphere Commerce Payments failed processing a MeAqCInit message with return code *rc*.

Severity: Error

Explanation: A MeAqCInit message could not be processed from the Certificate Authority due to an invalid root hash.

User Response: A valid root hash must be entered.

CEPSET1303 WebSphere Commerce Payments CertRes message processing failed with return code *rc*.

Severity: Error

Explanation: An error occurred while processing an incoming SET message from the Certificate Authority.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET1304 SET transaction processing entered an incorrect state for Merchant*merchant_number*, **brand** *brand_name*.

Severity: Error

Explanation: An incorrect state was generated by the Finite State Machine for SET transaction processing.

User Response: This is an internal WebSphere Commerce Payments error; contact your service representative.

CEPSET1305 There is already a brand defined for the specified brandID *brand_name*, **BIN** *bin*, **and merchantID***merch_ID*.

Severity: Error

Explanation: WebSphere Commerce Payments located a Brand in the ETBRANDCFG table that has the same brandID, BIN, and MerchantID that was specified in the create request.

User Response: Either remove the brand from the

ETBRANDCFG configuration table that is a duplicate and retry the request, or use the existing brand.

CEPSET1306 WebSphere Commerce Payments failed processing a MeAqCInitRes message with return code *rc*.

Severity: Error

Explanation: A MeAqCInit response message could not be processed from the Certificate Authority.

User Response: Contact your service representative.

CEPSET1307 WebSphere Commerce Payments CertRes message processing failed with return code *rc*.

Severity: Error

Explanation: An error occurred while processing the incoming CertRes SET message from the Certificate Authority due to problems with the answers supplied for the Registration Form.

User Response: Correct the registration form answers that are in error and resubmit the request to obtain certificates.

CEPSET1308 Unable to open the Certificate database. A new (and empty) Certificate database has been created.

Severity: Information

Explanation: An existing Certificate database could not be found at the specified path (if using flat file) or the database table could not be found (if using ODBC). A new, empty Certificate database has been created. Form.

User Response: If you expected WebSphere Commerce Payments to find an existing certificate database, then check to see that the path or database location that is specified in ETSETCFG is correct. Otherwise, no action is needed.

CEPSET1309 The specified Certificate Authority URL: *caURL* is not valid.

Severity: Error

Explanation: The Certificate Authority URL must be of the format: *"://[:]/[/]"*

User Response: Reenter a valid Certificate Authority URL.

CEPSET1310 The specified Merchant Cassette Object: *value* is not valid.

Severity: Error

Explanation: The Merchant Cassette Object that was specified on the Create, Modify, or Delete command is not valid. *"://[:]/[/]"*

User Response: Enter a valid Merchant Cassette Object

CEPSET1311 An error occurred while attempting to *en_decrypt* the data.

Severity: Error

Explanation: Unable to encrypt/decrypt data such as the Registration Form, the Registration Form Answers, the Account Data, or the Account Data Answer, etc.

User Response: Contact your service representative.

CEPSET1313 There are existing certificates in the certificate database with the same credentials (brand, bin, merchantID) that are specified in the create brand request for brand *brand*. The newly added brand will use the existing certificates.

Severity: Information

Explanation: The newly created brand will not obtain new certificates, but rather, will use the existing certificates in the certificate database.

User Response: No response required.

CEPSET1314 Merchant certificates for brand *brand* will expire on *expDate*.

Severity: Information

Explanation: The certificates for the specified brand are going to expire soon. If you don't renew the certificates before they expire, then you will have to obtain new certificates for the brand after they expire.

User Response: Renew certificates for specified brand.

CEPSET1315 WebSphere Commerce Payments An error occurred with return code *rc* while either composing a message for the Certificate Authority or while processing a message from the Certificate Authority.

Severity: Error

Explanation: An error occurred while composing or processing a SET message.

User Response: This is an internal error returned by the SET Toolkit; contact your service representative.

CEPSET1316 WebSphere Commerce Payments unable to communicate with the Certificate Authority with Host Name *hostName* **and Port Number** *portNum.*

Severity: Error

Explanation: Unable to send a message to the specified Certificate Authority.

User Response: Check the connection to the Certificate Authority and verify that there is not a problem with the communication between the WebSphere Commerce Payments machine and the Certificate Authority. If communication between WebSphere Commerce Payments and the Certificate Authority is OK, then contact your service representative.

CEPSET1317 Unable to delete batch *batch_number* **for merchant** *merchant_number* **because there are non-deleted transactions.**

Severity: Error

Explanation: The Cassette for SET was unable to delete a batch.

User Response: Delete all payments and credits in the batch by issuing the appropriate `CloseOrder` commands and then reissue the BatchDelete command.

CEPSET1318 The PCERT request was not processed successfully by the gateway due to an unspecified failure.

Severity: Error

Explanation: An unspecified failure was returned by the gateway for PCertCode.

User Response: Verify that the gateway is working properly and that all information given is valid.

CEPSET1319 The PCERT request was not processed successfully by the gateway because a BRAND specified in the message was not supported.

Severity: Error

Explanation: An unsupported BRAND was returned by the gateway for PCertCode.

User Response: Verify that the gateway is working properly and that the brand being specified is valid.

CEPSET1320 The PCERT request was not processed successfully by the gateway because a BIN specified in the message was not supported.

Severity: Error

Explanation: An unsupported BIN was returned by the gateway for PCertCode.

User Response: Verify that the gateway is working properly and that the BIN being specified is valid.

CEPSETSDK1026 The message could not be decoded.

Severity: Error

Explanation: This generally means a message was received which was not in the SET format expected. Some example situations include:

- A certificate response was missing required information or it contained unexpected information. Possibly a SET error message was returned, or some other kind of non-SET message was sent from the URL specified as the Certificate Authority in the Brand configuration.
- The machine at the Gateway address specified in the Acquirer Settings returned a non-SET error message.
- When communicating with a SET Wallet, the proxy returns a message which is not in SET format. Wallet expects a CERN proxy.

User Response: Look in the logs to determine the nature of the error. Then, check configuration settings related to the problem that occured. Verify that the original settings for your Acquirer and Certificate Authority are still valid. Often the URLs change when the software or hardware for those entities are upgraded. Check the Gateway URI in Acquirer Settings, Certificate Authority URL in Brand settings, or the proxy server.

CEPSETSDK3011 Certificate Issuer was not found in the database.

Severity: Error

Explanation: This indicates that the received SET message did not contain all the certificates and revocation information required to process the message. The needed information does not already exist in the certificate database.

User Response: Check the logs to see if there is any related information. Contact the Acquirer that sent the message. They may ask you to try the command again so they can capture the error logs from their software.

CEPSETSDK4001 Indicates that a brand new root certificate was received which requires a root hash for verification.

Severity: Error

Explanation: This error appears if the message received is a message signed with a certificate that is not yet trusted.

This could happen in a test environment if you are either requesting a certificate from a Certificate Authority which is using a non-standard root certificate, or communicating with an entity which uses a non-standard root certificate that you have not configured to trust.

User Response: Verify that the root hash entered is legitimate through secure means.

The message may be a SET error message identifying that the date and time settings of the server are not accurate. Verify the time zone, date, and time settings on WebSphere Commerce Payments. If these are correct, contact the owner of failing Gateway or Certificate Authority and ask them to verify the date, time and issuer certificates.

CEPSETSDK10014 Gateway certificate is required for message.

Severity: Error

Explanation: This error may occur when the Gateway certificate is not available for the Acquirer being used. The Wallet is initiating an exchange and the SET Cassette cannot compose the response because the specified Account brand does not have a valid Gateway certificate.

User Response: Look in the logs to see which Account and Brand are being referenced by the Wallet. Then look at the Brand details and select the Retrieve Gateway Certificate button.

CEPSETSDK15002 The certificate has expired.

Severity: Error

Explanation: Once a merchant certificate expires, it cannot be renewed.

User Response: Stop WebSphere Commerce Payments. Use the CertUtil program to delete the expired merchant encryption and signature certificates. Start WebSphere Commerce Payments. Using the WebSphere Commerce Payments user interface, view the Brand. On the Brand details screen, select the Continue Request button. Fill in the certificate request information. For more detailed information see the FAQs posted on the IBM Web site for WebSphere Commerce Payments.

CEPSETSDK17003 Currency exponent is not valid for given currency.

Severity: Error

Explanation: The Currency and AmountExp10 pair specified in the AcceptPayment or ReceivePayment API is not correct.

User Response: The currency codes are documented in an appendix in the WebSphere Commerce Payments Programmer's Guide.

CEPSETSDK18015 Failed to find a certificate.

Severity: Error

Explanation: The program may be looking for a merchant or a gateway certificate. Sometimes the program will continue and look elsewhere for the certificate. If this error is at the end of an eTillSETMsgDump file, then it is a true error. View the PMTrace and eTillSETMsgDump files to determine the nature of the failure.

User Response:

- If you were trying to send a PCertReq, verify in the Brand details screen that the merchant certificate is checkmarked. For closer inspection, use CertUtil to verify the merchant signature and encryption certificates.
- If you were trying to send a message to the Acquirer, verify in the Brand details screen that the gateway certificate is checkmarked. For closer inspection, use CertUtil to verify the gateway encryption certificate.
- Verify the following settings:
  1. Certificate Authority URL is the current and correct address. Check with the Certificate Authority to verify that this address hasn't changed.
  2. Merchant ID is correct in the Brand details. Check with your Acquirer to verify the MerchantID to specify.
  3. BCI is valid (not expired). See message CEPSET0305.
  4. If you have multiple merchants which have the same brand, make sure each definition of this brand has a unique MerchantID in the brand configuration.
  5. The current time is within the merchant and gateway certificates validity dates. If you recently received a merchant or gateway certificate, it might not be valid yet. Sometimes Certificate Authorities and Acquirers set the notbefore date in advance. Use CertUtil to dump the certificates information to disk.

Appendix H. Notices

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

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

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

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

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

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

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

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

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

IBM Corporation

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

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

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

Trademarks

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

- AIX
- iSeries
- DB2
- IBM
- IBM Consumer Wallet
- IBM Payment Gateway
- IBM Payment Registry
- WebSphere
- RS/6000

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

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

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

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.networking.ibm.com/nsg/nsgmain.htm.

The following cross-references are used in this dictionary:

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

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

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

A

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

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

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

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

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

ADSM. See ADSTAR Distributed Storage Manager.

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

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

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

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

approve all. Selects all orders displayed for approval.

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

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

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

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

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

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

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

B

balance. Within IBM e-commerce, an attribute of a WebSphere Commerce Payments Batch object. Indicates whether the merchant and financial institution agreed on the contents of the batch when it was closed. See 172 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 173) 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 172). 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.

I

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 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 he 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 164.

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

U

uniform resource locator (URL). (1) A sequence of characters that represent information resources on a computer or in a network such as the Internet. This sequence of characters includes (a) the abbreviated name of the protocol used to access the information resource and (b) the information used by the protocol to locate the information resource. For example, in the context of the Internet, these are abbreviated names of some protocols used to access various information resources: `http` , `ftp` , `gopher` , `telnet` , and `news` ; and this is the URL for the IBM home page: `http://www.ibm.com` . (2) The address of an item on the World Wide Web. It includes the protocol followed by the fully qualified domain name (sometimes called the host name) and the request. The Web server typically maps the request portion of the URL to a path and file name. For example, if the URL is `http://www.networking.ibm.com/nsg/nsgmain.htm` , the protocol is `http` ; the fully qualified domain name is `www.networking.ibm.com`; and the request is `/nsg/nsgmain.htm`.

URL. See uniform resource locator.

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

V

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

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

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

W

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

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

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

Web. See World Wide Web.

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

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

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

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

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

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

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

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

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

WWW. See World Wide Web.

X

XML. See Extensible Markup Language.

Numerics

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

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

Index

Special Characters

(AVS), Address Verification Service 88
$ITEM parameters 101
$PCARD parameters 101

Numerics

2KP/3KP, understanding 21
4758, configuring 109

A

AcceptPayment 47
access roles 23
Account
  creating 35
  object
    attributes 77
    overview 9
    XML example 78
acquirer behavior 11
Acquirer information 33
Address Verification Service (AVS) 88
address, cardholder's billing 88
Administration objects 9
API commands
  AcceptPayment 47
  Approve 50
  ApproveReversal 50
  BatchOpen 50
  BatchPurge 16
  CancelOrder 51
  CloseOrder 51
  CreateAccount 51
  CreateMerchantCassettObject 52
  DeleteBatch 55
  DeleteMerchantCassetteObject 56
  Deposit 56
  mapping to SET messages 8
  ModifyAccount 57
  ModifyCassette 58
  ModifyMerchantCassetteObject 58
  ReceivePayment 61
  Refund 65
API style
  explicit 14
  implicit 14
Approve 50
ApproveReversal 50
approving orders 38
automatic retry logic 104
AVS data 67
AVS response code 88
AVSData 88

B

batch
  account association 75
  Acquirer Controls 13
  Merchant Controls 13
  object 75
    attributes 75
    states 76
    XML example 76
  processing 13
  reconciliation 15
  settling 40
Batch totals
  Viewing 43
Batch Totals report, Daily 43
BatchOpen 50
BatchPurge 16
billing address, cardholder's 88
Brand
  creating 36
  object
    attributes 79
    overview 10
    states 83
    XML example 86

C

CancelOrder 51
cardholder's billing address 88
cash register ix
Cassette
  object
    attributes 76
    XML example 77
Cassette for SET
  commands 47
  configuring 33
  installing 25
  messages 137
  pre-installation 25
  return codes 115
  tutorial 33
    approving order 38
    creating account 35
    creating brand 36
    creating order 37
    depositing paymentr 39
    issuing credit 41
    settling batch 40
  uninstalling
    AIX 30
    iSeries 31
    Solaris 30
    Windows 29
cassette specific event 22

certificate authority
  definition of viii
Certificate information 34
Certificate request
  states 83
Certificate utility
  Certificates 91
chargeback liabilities, merchant 89
CloseOrder 51
Commands
  AcceptPayment 47
  Approve 50
  ApproveReversal 50
  BatchOpen 50
  BatchPurge 16
  CancelOrder 51
  CloseOrder 51
  CreateAccount 51
  CreateMerchantCassettObject 52
  DeleteBatch 55
  DeleteMerchantCassetteObject 56
  Deposit 56
  mapping to SET messages 8
  ModifyAccount 57
  ModifyCassette 58
  ModifyMerchantCassetteObject 58
  ReceivePayment 61
  Refund 65
configuring Cassette for SET 33
configuring socks server 28
CreateAccount 51
CreateMerchantCassetteObject 52
creating
  account 35
  brand 36
  orders 37
Credit
  issuing 41
  object
    attributes 74
    states 75
    XML example 75
cryptographic coprocessor, configuring 4758 109

D

Daily Batch Totals report 43
delayRetryInterval parameter 104
DeleteBatch 55
DeleteMerchantCassetteObject 56
Deposit 56
depositing payments 39
directory structure 27

E

EAR file 27
Electronic Transaction LLC Web site, SET Secure x
enterprise archive (EAR) file 27
errors, connection 105
event notification 22
expiration of 4758 user profile passwords 111

F

financial entities 1
Financial objects 11
Framework objects 67
fraud prevention tool 88

G

glossary 161

I

IBM CommercePOINT Wallet viii
IBM Payment Gateway for AIX ix
IBM Payment Registry for AIX viii
immediate retries 104
information
  Acquirer 33
  Certificate 34
installing
  Cassette for SET 25
Installing
  Pre-installation caveats 25
issuing credits 41

J

JPO extension 22
JVM settings 28

L

Level I, II, and III data for purchasing cards 19
liabilities, merchant chargeback 89
line item detail considerations 20
LLC Web site, SET Secure Electronic Transaction x

M

managing payment processing 37
MAXDELAYEDRETRIES parameter 104
merchant chargeback liabilities 89
Merchant Initiated Authorizationl 17
merchant server
  definition of viii
message flows
  cardholder/merchant 2
  merchant/financial institution 3
messages
  Cassette for SET 137
MIA
  See Merchant Initiated Authorizationl
ModifyAccount 57
ModifyCassette 58
ModifyMerchantCassetteObject 58
multiple payments 12

N

Notices 159

O

object
- Account
  - attributes 77
  - overview 9
  - XML example 78
- Batch
  - attributes 75
  - states 76
  - XML example 76
- Brand
  - attributes 79
  - overview 10
  - states 83
  - XML example 86
- Cassette
  - attributes 76
  - XML example 77
- Credit
  - attributes 74
  - states 75
  - XML example 75
- Order
  - attributes 67
  - states 70
  - XML example 70
- Payment
  - attributes 71
  - states 74
  - XML example 73
- state 70

objects, Framework 67
Order
- approving 38
  - sale function 39
- creating 37
- line item detail 20
- object
  - attributes 67
  - states 70
  - XML example 70
- order details 39

order creation
- with wallet 17
- without wallet 17

P

parameters, initialization 27
Payment
- depositing 39
- object
  - attributes 71
  - states 74
  - XML example 73

payment gateway
- definition of viii

payment processing
- managing 37
- split payments 12

purchasing cards
- description 18
- parameters 101

Q

query results, limiting 23

R

ReceivePayment 61
Refund 65
report, Daily Batch Totals 43
response code, AVS 88
retry mechanism for payment transfer 103
return codes
- Cassette for SET 115

S

sale function
- approving orders using 39

sale transactions 20
SaleDetail 88
- parameters 93
- purchasing cards 20

Sample Checkout 38
Secure Electronic Transaction LLC Web site, SET x
sensitive data, protecting 23
SET init message, extracting 21
SET profiles 11
SET Protocol
- introduction 1
- message flows
  - cardholder/merchant 2
  - certificate 2, 4
  - funds approval 6
  - merchant/financial institution 3
  - purchase sequence 5
- understanding 1

SET Secure Electronic Transaction LLC Web site x
settling batches 40
site, SET Secure Electronic Transaction LLC Web x
socks server, configuring 28
split payments 12

T

terms, glossary 161
Totals report, Daily Batch 43
trademarks 160
Transaction LLC Web site, SET Secure Electronic x
transactions, sale 20
travel and expense data, considerations 20
tutorial, Cassette for SET 33

tutorial, Cassette for SET *(continued)*
  approving order 38
  creating account 35
  creating brand 36
  creating order 37
  depositing payment 39
  issuing credit 41
  settling batch 40

U

uninstalling
  Cassette for SET
    AIX 30
    iSeries 31
    Solaris 30
    Windows 29
user interface
  logging in 35

V

Viewing batch totals 43

W

wakeup message
  host address, configuring 22
wallet
  definition viii
WAR file 27
Web archive (WAR) file 27
Web site, SET Secure Electronic Transaction LLC x
WebSphere Commerce Payments
  commands 47
  First-time set-up 33
WebSphere Commerce Payments Administration
 object 76

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

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

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

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

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

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

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 _____

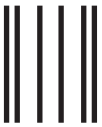
Company or Organization _____

Phone No. _____

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

Cut or Fold
Along Line

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 G7IA/ Bldg 503
P.O. Box 12195
Research Triangle Park, NC
27709-9990

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.