IBM WebSphere Commerce

Payments CustomOffline Cassette Supplement

Version 5.5

IBM WebSphere Commerce

Payments CustomOffline Cassette Supplement

Version 5.5

Note

Before using this information and the product it supports, be sure to read the general information under "Notices", on page 49.

First Edition (June 2003)

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

Contents

About this book

Who should read this book? This supplement is for users and Administrators of the CustomOffline Cassette who are responsible for deploying the cassette in a WebSphere Commerce Payments environment. This book explains the concepts behind the cassette and what you need to implement the cassette. Programmers who are responsible for developing applications to manage WebSphere Commerce Payments may find the cassette-specific information provided here useful. Reference information about API commands, parameter requirements, objects and XML examples of object use are included.

This book is a supplement to the *WebSphere Commerce Administration Guide* and *WebSphere Commerce Installation Guide*.

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

Conventions used in this book

This book uses the following highlighting conventions:

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

When you see the following names, substitute your system value as described.

▶ Windows indicates information specific to the Windows® operating environment.

▶ AIX indicates information specific to AIX®.

▶ Solaris indicates information specific to the Solaris Operating Environment.

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

▶ Linux indicates information specific to Linux.

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

WC_installdir represents the following default installation paths for WebSphere Commerce:

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

Linux Solaris /opt/WebSphere/CommerceServer*nn*

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

400 /QIBM/ProdData/CommerceServer*nn*

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

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

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

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

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

Terminology used in this book

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

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

Store* and *Merchant

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

Site Administrator* and *Payments Administrator

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

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

Merchant Administrator

In the Payments component user interface, the term *Merchant Administrator* is used to refer to the Site Administrator.

Additional information

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

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

Using the online help

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

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

Locating the printable documentation

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

http://www.adobe.com

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

Viewing the WebSphere Commerce Web site for product information

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

- Business Edition:

  http://www.ibm.com/software/webservers/commerce/wc_be/
- Professional Edition:

  http://www.ibm.com/software/webservers/commerce/wc_pe/
- Professional Entry Edition:

  http://www.ibm.com/software/webservers/commerce/wc_pentry/

A copy of this book, and any updated versions of this book, are available as PDF files from the Library section of the WebSphere Commerce Web site at:

www.ibm.com/software/commerce/wscom/library/lit-tech.html

Other WebSphere Commerce Payments documents and Web sites

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

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

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

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

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

Chapter 1. WebSphere Commerce Payments overview

This chapter describes WebSphere Commerce Payments in its various deployments, user roles, multiple payment methods, the user interface and internal components.

What is WebSphere Commerce Payments?

WebSphere Commerce Payments is a component of WebSphere Commerce which provides secure, electronic payment processing to Internet merchants. Based on open standards technology, WebSphere Commerce Payments works with payment *cassettes* to support multiple payment protocols, including:

- VisaNet, a provider of worldwide telecommunications data and payment processing that has the ability to authorize and settle payments.
- BankServACH, a payment gateway that interfaces with the Automated Clearing House Network (ACH) to support online electronic check payments.
- Paymentech, a merchant processor of online credit card payments with direct links to Visa and MasterCard.
- Third-party payment cassettes written for WebSphere Commerce Payments.

Simply put, WebSphere Commerce Payments integrates with WebSphere Commerce and provides cash register-like functionality to manage payment processing. The buyer never interacts with WebSphere Commerce Payments directly because WebSphere Commerce Payments sits behind the Internet merchant's storefront, receiving payments and processing those payments with banks and other financial institutions.

Accommodating multiple payment methods

WebSphere Commerce Payments implements a multipayment framework architecture that provides a flexible and extensible way to accommodate merchants who need to accept multiple payment methods. The multipayment framework separates payment management, the *framework*, from specific payment *cassettes*, so that each can evolve and operate independently.

WebSphere Commerce Payments provides a plug-in architecture whereby software cassettes for each payment method attach to the payment framework. The framework provides the generic infrastructure functions required for making and receiving payments using any payment method.

Payment cassettes are software applications that conform to the data flow and control conventions of the WebSphere Commerce Payments framework. Each payment cassette contains the implementation of specific payment methods and protocols.

Cassettes can be written by IBM or by third-party payment system implementors. IBM supports cassette development and offers detailed instruction to developers interested in writing their own payment cassettes. For more information on cassette development, see the *WebSphere Commerce Payments Cassette Kit Programming Guide* and downloads at the following Web address:

http://www.software.ibm.com/commerce/payments/download.html

WebSphere Commerce Payments roles

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

There are four roles within the Payments framework which have relative mappings to corresponding roles in WebSphere Commerce. This is useful to know if you are an administrator who creates users and assigns roles. When you create users within the WebSphere Commerce Organization Administration Console, you must first assign those users the WebSphere Commerce roles listed below. Then the users will display in the Payments UI where you can assign them their corresponding Payments roles. The following Payments user roles table provides recommended WebSphere Commerce role assignments:

Table 1. Roles

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

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

Table 2. Role responsibilities

Role	Responsibilities
Payments Administrator	• Define Merchant Administrators, Supervisors and Clerks • Configure merchants and their cassettes • Identify the Payments host name and status • Configure any installed cassettes • Add, delete and update event listeners • Settle payments • Approve or sale orders • Issue credits and reverse credits • Deposit orders • Search for orders and batches • View daily batch totals
Merchant Administrator	• Define Merchant Administrators, Supervisors and Clerks • Configure merchants and their cassettes • Add, delete and update event listeners

Table 2. Role responsibilities (continued)

Role	Responsibilities
Supervisor	• Settle payments • Approve or sale orders • Issue credits and reverse credits • Deposit orders • Search for orders and batches • View daily batch totals
Clerk	• Settle payments • Approve or sale orders • Deposit orders • Search for orders and batches • View daily batch totals

WebSphere Commerce Payments user interface

The WebSphere Commerce Payments user interface can be used by Payments Administrators and individual merchants using hosted WebSphere Commerce Payments to:

- Configure WebSphere Commerce Payments
- Perform the following routine payment-processing tasks in a payment-neutral way:
  - Approve payments
  - Deposit payments
  - Settle batches
  - Issue credits
  - View daily batch totals

The browser-based WebSphere Commerce Payments user interface can be accessed remotely and securely using the Secure Sockets Layer (SSL) capabilities of your Web browser.

WebSphere Commerce Payments components

To understand WebSphere Commerce Payments, it is useful to look at the following WebSphere Commerce Payments components:

- Web server configured with WebSphere Application Server
- WebSphere Commerce Payments Servlet
- User Interface servlet
- Database

The WebSphere Commerce Payments Servlet is the major component for WebSphere Commerce Payments. The Servlet works with WebSphere Application Server – a product that provides a common servlet environment for all platforms.

By using a Web server-based architecture, WebSphere Commerce Payments utilizes the Web server's ability to handle HTTP requests while at the same time leveraging its many security features.

Chapter 2. What is the CustomOffline Cassette?

The CustomOffline Cassette is an instance of a *passive* cassette which is automatically installed with Payments to manage manual payment transactions. Passive cassettes record events that have already happened outside of WebSphere Commerce Payments within the WebSphere Commerce Payments object model. Transactions are recorded and maintained only in the WebSphere Commerce Payments database— there is no back-end financial system with which a passive cassette communicates. An Approve command issued through the CustomOffline Cassette records the results of an approval decision which the merchant makes manually, possibly with the help of some third party to evaluate the overall risk associated with the purchase request.

This supplement describes the WebSphere Commerce Payments CustomOffline Cassette, which allows merchants to use WebSphere Commerce Payments for managing information surrounding *manual* payment transactions, such as Collect On Delivery (COD), Bill Me Later, or any other merchant-defined method for manually collecting payments. A tutorial is included to describe how to configure both COD and BillMe. This supplement suggests a method for mapping typical manual operations to WebSphere Commerce Payments actions. However, regardless of the manual payment method, the merchant ultimately decides how to map real-world actions into WebSphere Commerce Payments. For example, the merchant will decide when to do the following:

- Mark payments as being approved or deposited
- Create a new credit
- Close a batch

Using the CustomOffline Cassette, the merchant can do the following

- Collect general information about the buyer in WebSphere Commerce Payments through their online store. The AcceptPayment command enters this information into WebSphere Commerce Payments, which stores the data in a new Order object. Along with the cassette-defined payment information, such as the billing address, the cassette allows the merchant to store up to two pieces of their own data with the order.
- Retrieve buyer information through the WebSphere Commerce Payments user interface for use in approving and collecting payments.
- Record the state of the manual transactions in WebSphere Commerce Payments. For example, once the merchant has determined that the risk associated with a given purchase is acceptable, the merchant would use the Approve button on the Order page to record the fact that they approve the purchase. merchants can record all payment, credit and batch transactions.
- Manage daily batches using the WebSphere Commerce Payments user interface and reports.

Chapter 3. Before you start

Before you can configure the CustomOffline Cassette, do the following:

- Ensure that WebSphere Commerce Payments is part of your WebSphere Commerce installation.
- Create a WebSphere Commerce Payments instance. (Alternatively, you can add a cassette to an existing Payments instance.)
- Use the Configuration Manager to add a cassette to the WebSphere Commerce Payments instance.
- Start the WebSphere Commerce Payments instance.
- Create a merchant and Merchant Administrator for that merchant.

To configure a cassette, you must log on to WebSphere Commerce Payments as a Merchant or Payments Administrator. is installed with the CustomOffline Cassette. The minimum framework that this cassette supports is WebSphere Commerce Payments Version 5.5. For more information, see the *IBM WebSphere Commerce Installation Guide*.

Chapter 4. Tutorial

This tutorial guides you through an initial WebSphere Commerce Payments setup. As part of this initial setup, and to demonstrate the most common administration and payment functions, WebSphere Commerce Payments provides tutorial support using the CustomOffline Cassette and a Sample Checkout. For detailed information on administration, configuration and payment functions, see the online help for the WebSphere Commerce Payments user interface.

Following are the five must-do tasks to set up an operational WebSphere Commerce Payments:

1. Access the WebSphere Commerce Payments user interface
2. Create a WebSphere Commerce Payments merchant and authorize a payment cassette
3. Define WebSphere Commerce Payments users
4. Assign user roles
5. Create an account

   Once the configuration tasks are complete, you are ready to create sample orders and begin the payment processing tasks that merchants typically perform on a daily basis:
6. Create an order
7. Approve orders
8. Deposit payments
9. Settle batches
10. Issue credits
11. View daily batch totals

Step 1: Accessing the WebSphere Commerce Payments user interface

The first step is to log on to Payments as the Payments Administrator.

To log on to the WebSphere Commerce Payments user interface, do the following:

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

   If you are using SSL with the Payments instance, use https:// instead.
2. Type your Payments User ID.
3. Type your Payments Password.
4. Click **Logon**.

Important: If the HTTP server that the WebSphere Commerce Payments instance is using is configured for a port number other than the default, include the port number following the host name in the WebSphere Commerce Payments. For example, the Web address http://*host_name*:80/webapp/PaymentManager represents port number 80.

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

- Click the multidirectional arrow to refresh the page.
- Click the left-pointing arrow to return to the last page visited.
- Click the question mark to access context sensitive online help for the page.

Step 2: Creating a WebSphere Commerce Payments merchant and authorizing a cassette

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

1. From the navigation frame click **Merchant Settings**.
2. From the Merchant Settings page click **Add a Merchant**.
3. On the Merchant Settings page, type the following information (note that the monospaced text must be entered in these fields for the tutorial):

Table 3. Create a merchant fields

Field name	**Description**
Merchant name	Type `Custom Store`. This is the name that you assign to the merchant. Its only function is to display in the user interface.
Merchant number	Type `987654321`. This is a number that you assign which uniquely identifies the merchant in all transaction data.
Authorized cassettes	Check the box for the **CustomOffline Cassette**. Checking this box authorizes the merchant to use this payment cassette.

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

If you have already created a merchant whom you want to authorize this cassette to use, do the following:

1. Click **Merchant Settings**.
2. Click the Merchant Name.
3. Select the box for **CustomOffline**.
4. Click **Update**.

The merchant is now authorized to use this cassette.

Step 3: Defining WebSphere Commerce Payments users

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

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

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

To configure Payments users, do the following:

1. In a Web browser point to https://*host_name*:*port*/orgadminconsole
2. Click **Access Management>Users**.
3. Click **New**.
4. Create the new user, Pat, using the New User wizard.
5. From the Role page, assign Pat the role of Store Adminstrator.

If you are installing WebSphere Commerce Payments on a system remotely from WebSphere Commerce products and would like to use the WebSphere Commerce stylesheet in your WebSphere Commerce Payments user interface, you must copy the `PMCustomUI.properties` file located in the *Payments_installdir*\samples\wcs\PMCustomUI.properties directory to the main WebSphere Commerce Payments installation directory.

Step 4: Assigning user roles

Users must be assigned to one of the following WebSphere Commerce Payments roles, which have relative mappings to the corresponding WebSphere Commerce roles:

Table 4. Role mapping

Payments role	**WebSphere Commerce role**
Payments Administrator	Site Administrator
Merchant Administrator	Site Adminstrator
Supervisor	Operations or Sales Manager
Clerk	Customer Service Supervisor

After creating the following users:

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

you are ready to assign Pat's role in the WebSphere Commerce Payments configuration.

Note: You can also assign the role of *No WebSphere Commerce Payments access* to deny users access to WebSphere Commerce Payments. This may be useful if you have a temporary need to deny a user access, for example, an employee on leave of absence. For more information on WebSphere Commerce Payments role permissions, see the Role Permissions Table in the *WebSphere Commerce Payments Programming Guide and Reference*.

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

1. In a Web browser point to http://*host_name:port*/webapp/PaymentManager to log on to Payments.

   If you are using SSL with the Payments instance, use https:// instead.
2. From the navigation frame click **Users**.

3. On the Users Search page, type the user name `Pat` and click **Search**.
4. From the Users page, click the user name **Pat**.
5. From the **Merchant** scroll box, select **Custom Store**.
6. Select the radio button for **Merchant Administrator**.
7. Click **Update** to save the user configuration.

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

Logging on as the Merchant Administrator

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

1. From the navigation frame, click **Logoff** *user* on the navigation frame of the WebSphere Commerce Payments user interface, and you will return to the main WebSphere Commerce Payments Login window.
2. Type the **User ID** `Pat`.
3. Type the **Password** defined for Pat.
4. Click **OK**.

For the remainder of this tutorial, your role will be the user Pat, with Merchant Administrator authority for the Custom Store. Your view of the WebSphere Commerce Payments user interface is limited to merchant administration functions; where, as the Payments Administrator, you had a global view of both Merchant and Payments Administrator functions.

Step 5: Creating an account

So far, you have defined one merchant, the Custom Store, and enabled one payment cassette, the CustomOffline Cassette. Your first task as the Merchant Administrator is to establish an *account* for the CustomOffline Cassette.

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. For this tutorial, you will create one account for the CustomOffline Cassette.

To create an account, do the following:

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

Table 5. Add an account fields

Field name	**Description**
Account name	Type `Custom Account`. This is the name that you assign to the account. Its only function is to display in the user interface.

Table 5. Add an account fields (continued)

Field name	Description
Account number	Type 111111111. This is a number that either the hosting service provider or the Merchant Administrator, assigns to uniquely identify the account in all transactions.
Financial Institution name	Type `Custom Bank`. This is the name of the financial institution with which you hold this account. Its only function is to display in the user interface.
Payment method name	Type `COD` for the CustomOffline payment method name. This field is case-sensitive. In a hosted environment, you might want to customize this field.
Batch close time	The number of minutes past midnight that the cassette automatically closes batches for this account. A value of 0 (zero) represents midnight. 1439 is the maximum value allowed. A null value disables automatic batch closing.

6. Click **Create account** to create the account for the CustomOffline Cassette.

Step 6: Creating orders using the Sample Checkout

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

- Merchant-specific administration functions
- All payment processing functions

In a real business scenario, you may 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 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 this tutorial you will use the Sample Checkout tool to create three orders for payment processing. The Sample Checkout tool provides a user interface with which you can create sample orders to test your cassette implementation. To access Sample Checkout, you must first change the default user ID and password as described below.

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

1. From the directory:

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

   open the configuration file `SampleCheckout.xml`.
2. At the `SampleCheckout` element, change the following attributes:

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

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

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

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

   If you are using SSL with the Payments instance, use https:// instead.
5. At the Sample Checkout page enter the following information (note that the monospaced text must be entered in these fields for the tutorial):

Table 6. Sample Checkout CustomOffline fields

Field name	Description
Merchant number	Any number to represent a merchant number. For this tutorial use 987654321.
Order number	Any number to represent an order number.
Amount	Any number to represent the total numeric amount of the order.
Currency	From the drop-down list select US Dollar.
Payment method	Select the radio button for Cash on delivery.
Shipping Address	Type an address.

6. Click **Buy**.

Repeat these steps two more times so that you have three orders for which to process payments.

Step 7: Approving orders

Once you have created three orders using the Sample Checkout, you can approve those orders. Follow these steps to approve an order:

1. Point your browser again to http://*host_name:port*/webapp/PaymentManager and log on as Pat.

   If you are using SSL with the Payments instance, use https:// instead.
2. From the navigation frame, click **Approve**.
3. From the Approve page, select a box for one of the orders you placed.
4. Click **Approve Selected**. The Approve Results page displays the status of your approve request.

5. When your approval is complete, click **Return to the Approve screen**.

Two orders are still awaiting your approval. You could have approved them simultaneously from the Approve page for their full amount by clicking **Approve All**. However, to better demonstrate the approve functions, this tutorial describes how to work with each order individually.

Approving orders from the Order page

In this section, you will approve *part* of the total order amount of an order from the Order page. You may find it useful to approve only part of an order when some of the goods associated with the order are not available for delivery at order processing, for example, if merchandise is back-ordered.

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

Table 7. Order number approve fields

Field name	**Description**
Currency	The type of currency used to place this order. This is a read-only field.
Order Amount	The total amount of the order expressed in the currency used to place the order. This is a read-only field.
Approved Amount	This read-only field field displays zeros since no amount of the order has yet been approved.
Deposited Amount	This read-only field field displays zeros since no amount has yet been approved or deposited.
Approval Amount	This is the total amount of the order.
Authorization Code	The authorization code returned from the manual offline authorization request process. The payment state is changed to approved.
Decline Reason	The decline reason returned from the manual offline authorization process. The payment state is changed to declined.
AVS Result Code	The Address Verification System result code.

   Change the approval amount to `3.00`. Optionally, specify an authorization reason to approve the amount or decline reason to indicate that approval was declined.

4. Click **Approve** to approve this order for three dollars. When the approval processing is complete, the Order page refreshes and displays approval status.

Approving orders with the sale function

Because you approved *part* of the last order you worked with, you still have two order entries in the Approve page. In this step, 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 approves and deposits your order payment. Thus, you can think of sale as approve with auto-deposit. Use the sale function to expedite the delivery of goods to the buyer and guarantee your capture of funds, for example, when you are selling downloadable software or electronic information. However, with the sale function, you cannot set the authorization or decline reasons for the approval. Because sale merges approve and deposit into one transaction, it is also useful when you are charged on a per-transaction basis.

Do the following to approve an order with the sale function:
1. From the navigation frame click **Approve**.
2. From the Approve page, click **Sale All**. When processing is complete, the approval status displays for each order submitted for sale.
3. When your sale is complete, click **Return to the Approve Screen**.

Step 8: Depositing payments

The Deposit function allows you to deposit order payments. 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, do the following:
1. From the navigation frame click **Deposit**.
2. Select a box for one of the payments listed and click **Deposit Selected**. When the processing is complete, deposit status displays for the payment submitted for deposit.
3. When your deposit is complete, click **Return to the Deposit Screen**.

You can also deposit *part* of a payment. To deposit part of a payment, do the following:
1. From the Deposit page, click the **Payment number** for one of the payments awaiting deposit.
2. From the Payment page, click **Deposit**.
3. On the Order Payment page, change the deposit amount to `2.00` and click **Deposit**.

Step 9: Settling batches

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

To settle a batch, do the following:
1. From the navigation frame click **Batch Search**. Alternatively, you can click **Settle**.
2. In the Batch Search page you can enter the following information to narrow your search (note that for this tutorial, you do not need to complete these fields):

Table 8. Batch search fields

Field name	Description
Merchant	The name of the merchant whose batch you are searching for. If there are less than 500 merchants in the WebSphere Commerce Payments database, select the merchant name from the drop-down list. If there are more than 500 merchants in the WebSphere Commerce Payments database, type the name of a merchant.
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**
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 method, or protocol, used to place the order, for example, CustomOffline.
Batch Open Date	Use the after and before fields 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 to search for batches closed during the specified range in time: • **After** Specify a date to search for all batches closed on and after this date. • **Before** Specify a date to search for all batches closed on and before this date.
Account	The account for which this order is being processed. If more than 500 accounts have been defined, type the account number in the entry field.

3. Click **Search**.

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

To prune outdated information, you can delete the settled batch by clicking **Delete** on the Settle Results page after all orders and credits within the batch have been closed and deleted. When a batch is deleted, all ancillary information about that batch, that is, payments, credits, and cassette-specific data, is deleted, as well. If you need to retain all payment data, for example, for audit purposes, you should *not* delete a batch.

Step 10: Issuing a credit

Credits are issued against orders and can be given for any amount up to the total amount of the order.

To issue a credit, do the following:

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

Table 9. Order search fields

Field name	**Description**
Merchant	The name of the merchant whose order you are searching for. If there are less than 500 merchants in the WebSphere Commerce Payments database, select the merchant name from the drop-down list. If there are more than 500 merchants in the WebSphere Commerce Payments database, type the name of a merchant.
Order Number	A number assigned by the merchant that uniquely identifies the order.
State	The state of the order: • **Requested** • **Ordered** • **Refundable** • **Canceled** • **Closed**
Payment Type	Identifies the payment method, or protocol, used to place the order, for example, CustomOffline.
Order Date	Use the after and before fields below to search for orders opened during the specified range in time: • **After** Specify a date to search for all orders opened on and after this date. • **Before** Specify a date to search for all orders opened on and before this date.
Order Amount	• **Currency** The currency used to place this order. Select the currency type from the drop-down list. • **Greater than** Specify a value to retrieve all orders with order amounts that are greater than or equal to the value you specify. • **Less than** Specify a value to retrieve all orders with order amounts that are less than or equal to the value you specify.
Account	The account for which this order is being credited. If more than 500 accounts are defined, type the account number in the entry field.

3. Click **Search**.
4. From the Order Search Results page, click an order number for an order in Refundable state to view the details of that order.

5. From the Order page, click **Credit** to create a credit against this order.
6. The Create Credit page displays the following information:

Table 10. Create credit fields

Field name	Description
Currency	The type of currency used to place this order. This is a read-only field.
Order Amount	The total amount of the order expressed in the currency used to place the order. This is a read-only field.
Approved Amount	The total amount of the order that has been approved expressed in the currency used to place the order. This is a read-only field.
Deposited Amount	The total amount of the order that has been deposited expressed in the currency used to place the order. This is a read-only field.
Credit Amount	This field must be completed by the Merchant Administrator with the total amount to be credited to the shopper.
Authorization Reason	The authorization code returned from the manual offline authorization request process. The credit state is changed to Refunded.
Decline Reason	The decline reason returned from the manual offline authorization process. The credit state changes to Declined.

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

When credit processing has completed, the Order page refreshes and displays the credit status. The newly created credit displays under **Credits**.

Step 11: Viewing batch totals

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

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

1. From the navigation frame click **Reports**.
2. From the Reports page, click **Daily Batch Totals**.
3. In the Batch Totals Report page, type the **Date** for which you would like a batch totals report. Leave this field blank to generate a report for the current date.
4. Type or select the **Merchant name**. If you do not type a merchant name, a list of all of the batches for the specified date displays. If there are more than 500 batches, only the first 500 batches display.
5. Click **Search**.

The Daily Batch Totals report computes the totals for all batches that were closed on the date specified in the **Search** page. Since you did not specify a search date, the report that was generated contains the current day's batch totals. These totals are computed on a per-currency basis, so there is one line per currency. Note that these totals include all payments and credits made for all payment methods, not just those made through the CustomOffline Cassette.

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 WebSphere Commerce Payments. For more information on specific fields in the WebSphere Commerce Payments user interface, see the online help.

Chapter 5. Cashier profiles

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

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

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

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

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

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

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

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

The following are the CustomOffline Cassette Cashier profiles:

Table 11. CustomOffline Cassette Cashier profiles

Cashier profile file name	Function
`WC51_CustomOfflineBillMe.profile`	Uses full AVS information. Collects shipping and delivery address information.
`WC51_CustomOfflineCOD.profile`	Collects delivery address information.
`SampleCheckoutCustomOfflineBillMe.profile`	Uses full AVS information. Collects shipping and delivery address information for Sample Checkout.
`SampleCheckoutCustomOfflineCOD.profile`	Collects delivery address information for Sample Checkout.

Initially, the CustomOffline Cassette Cashier Sample Checkout profile is in the following directory:

Payments_installdir/cassettes/CustomOffline/SampleCheckout/profiles

When the cassette is added to an instance, the file is copied to the following directory:

Payments_installdir/wc.mpf.ear/SampleCheckout.war/profiles

If you want to change the profile for an instance, you must change the profile located in the following directory:

WAS_installdir/installedApps/*node_name*/*payments_instance*_Commerce_Payments_App.ear/
SampleCheckout.war/profiles

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

For details on designing and tailoring profiles, see the *WebSphere Commerce Payments Programming Guide and Reference*.

400 The CustomOffline Cassette Cashier profiles are in the directory:

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

Chapter 6. Command reference

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

- How (or whether) each optional framework parameter is supported by the CustomOffline Cassette
- Any special notes related to the CustomOffline Cassette handling of framework parameters
- All CustomOffline Cassette-specific protocol parameters

In each table, the Required/Optional column has a value of **R** for required or **O** for optional.

Financial commands

AcceptPayment

Table 12. Keywords for AcceptPayment command

Keyword	**Required / Optional**	**Value**
APPROVEFLAG	O	Fully supported.
PAYMENTAMOUNT	O/R	Fully supported. Required if APPROVEFLAG is set to 1.
DEPOSITFLAG	O	Fully supported.
BATCHNUMBER	O	Not allowed (must not be specified) since all batches are opened implicitly.
$METHOD	R	Indicates the manual payment method to be used. Must match one of the methods configured into the merchant's set of Manual accounts. ASCII string from 1 to 32 bytes long. No default payment method is preconfigured.
$COUNTRYCODE	O	Country code of buyer's address. Depending upon the manual payment method, this may be a billing or a shipping address. This field allows national language characters including DBCS.
$STREETADDRESS	O	Street address of buyer's address. Depending upon the manual payment method, this may be a billing or a shipping address. This field allows national language characters including DBCS.
$CITY	O	City of buyer's address. Depending upon the manual payment method, this may be a billing or a shipping address. This field allows national language characters including DBCS.
$STATEPROVINCE	O	State or province of buyer's address. Depending upon the manual payment method, this may be a billing or a shipping address. This field allows national language characters including DBCS.
$POSTALCODE	O	Postal (zip) code of buyer's address. Depending upon the manual payment method, this may be a billing or a shipping address. This field allows national language characters including DBCS.

Table 12. Keywords for AcceptPayment command (continued)

Keyword	Required / Optional	Value
$AUXILIARY1	O	Binary string, 0 to 254 characters long. merchant-supplied payment data. Only the merchant software knows what is stored in this field.
$AULILIARY2	O	Binary string, 0 to 254 characters long. Merchant-supplied payment data. Only the merchant software knows what is stored in this field.
$AUTHCODE	O	Authorization code to be saved with the approved payment. This parameter is only used if APPROVEFLAG=1 is also specified on the AcceptPayment command. Must be an ASCII character string between 1 and 64 bytes long. If $AUTHCODE is specified, a payment is also created in PAYMENT_APPROVED state for the order amount and the ReferenceNumber is set to the $AUTHCODE value. If specified with $DECLINEREASON, the command will fail with PRC_INVALID_PARAMETER_COMBINATION, RC_CASSETTE_AUTHCODE_ AND_DECLINEREASON. If neither this parameter nor $DECLINEREASON is specified with APPROVEFLAG=1, an approved payment is created for the order amount with an empty ReferenceCode.
$DECLINEREASON	O	Decline code to be saved with the declined payment. This parameter is only used if APPROVEFLAG=1 is also specified on the AcceptPayment command. Must be an ASCII character string between 1 and 16 bytes long. If $DECLINEREASON is specified, a Payment is also created in PAYMENT_DECLINED state and the ReferenceNumber is set to the $DECLINEREASON value. If specified with $AUTHCODE, the command will fail with PRC_INVALID_PARAMETER_COMBINATION, RC_CASSETTE_AUTHCODE_AND_DECLINEREASON

Approve

Table 13. Keywords for Approve command

Keyword	Required / Optional	Value
BATCHNUMBER	O	Not allowed (must not be specified) since all batches are opened implicitly.
DEPOSITFLAG	O	Fully supported.
$AUTHCODE	O	Authorization code to be saved with the approved payment. Must be an ASCII character string between 1 and 16 bytes long. If $AUTHCODE is specified, the payment is created in PAYMENT_APPROVED state and the ReferenceNumber is set to the $AUTHCODE value. If neither this parameter nor $DECLINEREASON is specified, the payment is created in PAYMENT_APPROVED state with an empty ReferenceCode. If specified with $DECLINEREASON, the command will fail with PRC_INVALID_PARAMETER_COMBINATION, RC_CASSETTE_AUTHCODE_AND_DECLINEREASON
$DECLINEREASON	O	Decline code to be saved with the declined payment. Must be an ASCII character string between 1 and 256 bytes long. If $DECLINEREASON is specified, the payment is created in PAYMENT_DECLINED state and the declineReason field is set to the $DECLINEREASON value. If specified with $AUTHCODE, the command will fail with PRC_INVALID_PARAMETER_COMBINATION, RC_CASSETTE_AUTHCODE_AND_DECLINEREASON

BatchClose

Table 14. Keywords for BatchClose command

Keyword	Required / Optional	Value
$FIBATCHID	O	Specifies the financial institution's identifier for the batch. This parameters allows the merchant to correlate the local batch object to an entity at the financial institution. If specified, the value is saved as the fiBatchId value in the CustomOffline Cassette's extension to the Batch object. Otherwise, that field is left empty. Must be an ASCII character string between 1 and 32 bytes long.

Deposit

Table 15. Keywords for Deposit command

Keyword	Required / Optional	Value
BATCHNUMBER	O	Not allowed (must not be specified) since all batches are opened implicitly.

Refund

Table 16. Keywords for Refund command

Keyword	Required / Optional	Value
BATCHNUMBER	O	Not allowed (must not be specified) since all batches are opened implicitly.
$AUTHCODE	O	Authorization code to be saved with the approved credit. Must be an ASCII character string between 1 and 16 bytes long. If $AUTHCODE is specified, the credit is created in CREDIT_APPROVED state and the ReferenceNumber is set to the $AUTHCODE value. If neither this parameter nor $DECLINEREASON is specified, the credit is created in CREDIT_APPROVED state with an empty ReferenceCode. If specified with $DECLINEREASON, the command will fail with PRC_INVALID_PARAMETER_COMBINATION, RC_CASSETTE_AUTHCODE_AND_DECLINEREASON
$DECLINEREASON	O	Decline code to be saved with the declined credit. Must be an ASCII character string between 1 and 256 bytes long. If $DECLINEREASON is specified, the credit is created in CREDIT_DECLINED state and the declineReason field is set to the $DECLINEREASON value. If specified with $AUTHCODE, the command will fail with PRC_INVALID_PARAMETER_COMBINATION, RC_CASSETTE_AUTHCODE_AND_DECLINEREASON

Other supported financial commands

The following financial commands are supported without any optional or protocol data parameters:

- ApproveReversal modifies the approved amount of a payment.
- BatchPurge clears out a batch and returns the Batch object to the Open state.
- CancelOrder moves an order into the cancelled state.
- CloseOrder moves an order into the closed state.
- DepositReversal disassociates a payment from a batch.
- DeleteBatch removes the specified batch from the database tables.
- RefundReversal voids existing credit objects.

Unsupported financial commands

The following financial commands are not supported and always return primary and secondary return codes of `PRC_COMMAND_NOT_SUPPORTED` :and `RC_NONE`

- BatchOpen
- ReceivePayment

Administrative commands

CreateAccount

Table 17. Keywords for CreateAccount command

Keyword	Required / Optional	Value
$METHOD	R	Indicates the manual payment method. Must be an ASCII string between 1 and 32 bytes long. Each account within a merchant must have a unique method value.
$BATCHCLOSETIME	O	This parameter specifies the time of day at which the cassette should automatically close the open batches. If not specified, all batches must be explicitly closed through the BatchClose command. This value is specified as an integer representing the number of minutes past midnight (where 0 represents midnight). Valid values are 0 <= time <= 1439

ModifyAccount

Table 18. Keywords for ModifyAccount command

Keyword	Required / Optional	Value
$BATCHCLOSETIME	R	This parameter specifies the time of day at which the cassette should automatically close the open batches. If not specified, all batches must be explicitly closed through the BatchClose command. This value is specified as an integer representing the number of minutes past midnight (where 0 represents midnight). Valid values are 0 <= time <= 1439

Other supported administrative commands

The following administrative commands are supported without any optional or protocol data parameters:

- CassetteControl
- CreatePaySystem
- DeleteAccount
- DeletePaySystem
- ModifyCassette
- ModifyPaySystem

Unsupported administrative commands

The following administrative commands are not supported and always return primary and secondary return codes of `PRC_COMMAND_NOT_SUPPORTED` and `RC_NONE`:

- CreateMerchantCassetteObject
- CreateSystemCassetteObject
- DeleteMerchantCassetteObject
- DeleteSystemCassetteObject
- ModifyMerchantCassetteObject
- ModifySystemCassetteObject

Chapter 7. Object reference

The CustomOffline Cassette object model closely reflects the generic model of WebSphere Commerce Payments. This section describes both the financial and the administration objects and their cassette extensions to the various framework objects.

Financial objects

The CustomOffline Cassette financial object model mirrors that of the generic model in that an CustomOfflineCassette object is defined to augment each generic financial object. The CustomOffline Cassette objects appear as extensions to the generic objects. Those extensions are as follows:

Order

Table 19. Order object cassette properties

Field name	Syntax	Description
auxiliary1	Binary string, 0 to 254 characters long.	Merchant-supplied payment data. Only the merchant software knows what is stored in this field.
auxiliary2	Binary string, 0 to 254 characters long.	Merchant-supplied payment data. Only the merchant software knows what is stored in this field.
country	Character string, 1 – 50.	ISO 3166 country code of cardholder’s country of residence or the country name. This value is only present if a non-null $COUNTRYCODE value was specified on the ACCEPTPAYMENT command.
street	Character string, 1 to 128 bytes long.	Cardholder’s street address. This value is only present if a non-null $STREETADDRESS value was specified on the ACCEPTPAYMENT command.
city	Character string, 1 to 50 bytes long.	Cardholder’s city of residence. This value is only present if a non-null $CITY value was specified on the ACCEPTPAYMENT command.
state	Character string, 1 to 50 bytes long.	Cardholder’s state or province of residence. This value is only present if a non-null $STATEPROVINCE value was specified on the ACCEPTPAYMENT command.
postalCode	Character string, 1 to 14 bytes long.	Cardholder’s postal code. This value is only present if a non-null $POSTALCODE value was specified on the ACCEPTPAYMENT command.

Payment

Table 20. Payment object cassette properties

Field name	Syntax	Description
authReason	Character string, 1 to 254 bytes long.	The authorization code entered through the $AUTHCODE parameter of the approve command. This field is present only when an authCode value exists for this approved payment.
declineReason	Character string, 1 to 254 bytes long.	The decline reason entered through the $DECLINEREASON parameter of the approve command. This field is present only when a declineReason exists for this declined payment.

Credit

Table 21. Credit object cassette properties

Field name	Syntax	Description
authReason	Character string, 1 to 254 bytes long.	The authorization code entered through the $AUTHCODE parameter of the Refund command. This field is present only when an authCode value exists for this credit.
declineReason	Character string, 1 to 254 bytes long.	The decline reason entered through the $DECLINEREASON parameter of the Refund command. This field is present only when a declineReason exists for this refund.

Batch

Table 22. Batch object cassette properties

Field name	Syntax	Description
FIBatchID	Character string, 1 to 32 bytes long.	The financial institution's identifier for the batch as entered through the $FIBATCHID parameter on the BatchClose command. The merchant should use this identifier when contacting the financial institution with inquiries concerning this batch.

Administrative object

The CustomOffline extends one framework administrative object - the AccountAdmin object. Only one account is supported for each merchant. The CustomOffline account extensions appear as extensions to the generic objects.

Those extensions are as follows:

Account

The CustomOffline account extensions appear as extensions to the generic objects. Those extensions are as follows:

Table 23. Account object cassette properties

Field name	Syntax	Description
method	Character string, 1 to 32 bytes long.	The manual payment method. For example, ″COD,″ ″BillMe,″ or some other merchant-defined method. This value is always present.
batchCloseTime	Integer value representing the number of minutes past midnight.	The time of day at which the currently open batches should be closed. This value is present only if it has been specified on the account configuration.

Chapter 8. CustomOffline Cassette return codes

CustomOffline return codes follow:

Secondary Return Code	Value	Description
RC_BUNDLE_ID_MISMATCH	20001	The CustomOffline cassette resource bundle ID mismatch.
RC_ACCOUNT_SELECT_SQL_FAILURE	21000	An SQL exception occurred while querying the CustomOfflineAccount table.
RC_ACCOUNT_SELECT_CLOSE_FAILURE	21001	An SQL exception occurred while closing the query session on the CustomOfflineAccount table.
RC_ACCOUNT_CREATE_ROW_FAILURE	21002	An SQL exception occurred while adding a row to the CustomOfflineAccount table.
RC_ACCOUNT_CREATE_SQL_FAILURE	21003	An SQL exception occurred while inserting a row to the CustomOfflineAccount table.
RC_ACCOUNT_UPDATE_ROW_FAILURE	21004	An SQL exception occurred while updating a row in the CustomOfflineAccount table.
RC_ACCOUNT_UPDATE_SQL_FAILURE	21005	An SQL exception occurred while updating the CustomOfflineAccount table.
RC_ACCOUNT_DELETE_ROW_FAILURE	21006	An SQL exception occurred while deleting a row from the CustomOfflineAccount table.
RC_ACCOUNT_DELETE_SQL_FAILURE	21007	An SQL exception occurred while deleting the CustomOffline account table.
RC_ACCOUNT_NULL_BATCH_NUMBER	21009	An attempt to retrieve a batch for a transaction with an empty batch number was made.
RC_BATCH_SELECT_SQL_FAILURE	22002	An SQL exception occurred while querying the CustomOfflineBatch table.
RC_BATCH_SELECT_CLOSE_FAILURE	22003	An SQL exception occurred while closing a query session of CustomOfflineBatch table.
RC_BATCH_CREATE_SQL_FAILURE	22005	An SQL exception occurred while inserting row into the CustomOfflineBatch table.
RC_BATCH_UPDATE_ROW_FAILURE	22006	An SQL exception occurred while updating a row in the CustomOfflineBatch table.
RC_BATCH_UPDATE_SQL_FAILURE	22007	An SQL exception occurred while updating the CustomOfflineBatch table.
RC_BATCH_DELETE_ROW_FAILURE	22008	An SQL exception occurred while deleting a row from the CustomOfflineBatch table.
RC_BATCH_DELETE_SQL_FAILURE	22009	An SQL exception occurred while deleting rows from the CustomOfflineBatch table.
RC_BATCH_NULL_ORDER_FOR_PAYMENT	22010	A payment's order cannot be NULL.
RC_BATCH_NULL_PAYMENT	22011	A payment object cannot be empty within a batch.
RC_BATCH_BAD_BATCH_IN_PAYMENT	22012	The batch number referenced in the payment object is different from that in the associated batch object.
RC_BATCH_NULL_ORDER_FOR_CREDIT	22013	A credit's order cannot be NULL.

Secondary Return Code	Value	Description
RC_BATCH_NULL_CREDIT	22014	A credit object cannot be empty within a batch.
RC_BATCH_BAD_BATCH_IN_CREDIT	22015	The batch number referenced in the credit object is different fromthat in the associated batch object.
RC_BATCH_PURGE_INCOMPLETE	22016	The batch purge operation is incomplete.
RC_ORDER_SELECT_ORDER_MISSING	23001	An SQL query for an order returns nothing.
RC_ORDER_SELECT_SQL_FAILURE	23002	An SQL query for orders failed.
RC_ORDER_SELECT_CLOSE_FAILURE	23003	An SQL exception occurred while closing an order query session.
RC_ORDER_CREATE_ROW_FAILURE	23004	An SQL exception occurred while inserting a row into the CustomOfflineOrder table.
RC_ORDER_CREATE_SQL_FAILURE	23005	An SQL exception occurred while inserting rows into the CustomOfflineOrder table.
RC_ORDER_UPDATE_ROW_FAILURE	23006	An SQL exception occurred while updating a row in the CustomOfflineOrder table.
RC_ORDER_UPDATE_SQL_FAILURE	23007	An SQL exception occurred while updating rows in the CustomOfflineOrder table.
RC_ORDER_DELETE_ROW_FAILURE	23008	An SQL exception occurred while deleting a row from the CustomOfflineOrder table.
RC_ORDER_DELETE_SQL_FAILURE	23009	An SQL exception occurred while deleting rows from the CustomOfflineOrder table.
RC_PAYMENT_SELECT_ROW_FAILURE	24000	An SQL exception occurred while querying a row from the CustomOfflinePayment table.
RC_PAYMENT_SELECT_PAYMENT_MISSING	24001	An error occurred while querying the CustomOfflinePayment table.
RC_PAYMENT_SELECT_SQL_FAILURE	24002	An SQL exception occurred while querying the CustomOfflinePayment table.
RC_PAYMENT_SELECT_CLOSE_FAILURE	24003	An SQL exception occurred while closing a query session.
RC_PAYMENT_CREATE_ROW_FAILURE	24004	An SQL exception occurred while inserting a row into the CustomOfflinePayment table.
RC_PAYMENT_CREATE_SQL_FAILURE	24005	An SQL exception occurred while inserting rows into the CustomOfflinePayment table.
RC_PAYMENT_UPDATE_SQL_FAILURE	24007	An SQL exception occurred while updating rows in the CustomOfflinePayment table.
RC_PAYMENT_DELETE_SQL_FAILURE	24009	An SQL exception occurred while deleting rows from the CustomOfflinePayment table.
RC_CREDIT_SELECT_CREDIT_MISSING	25001	An error occurred while querying the CustomOfflineCredit table.
RC_CREDIT_SELECT_SQL_FAILURE	25002	An SQL exception occurred while querying the CustomOfflineCredit table.
RC_CREDIT_SELECT_CLOSE_FAILURE	25003	An SQL exception occurred while closing a query session from the CustomOfflineCredit table.
RC_CREDIT_CREATE_SQL_FAILURE	25005	An SQL exception occurred while inserting rows into the CustomOfflineCredit table.
RC_CREDIT_UPDATE_ROW_FAILURE	25006	An SQL exception occurred while updating a row in the CustomOfflineCredit table.

Secondary Return Code	Value	Description
RC_CREDIT_UPDATE_SQL_FAILURE	25007	An SQL exception occurred while updating rows in the CustomOfflineCredit table.
RC_CREDIT_DELETE_ROW_FAILURE	25008	An SQL exception occurred while deleting a row from the CustomOfflineCredit table.
RC_CREDIT_DELETE_SQL_FAILURE	25009	An SQL exception occurred while deleting rows from the CustomOfflineCredit table.
RC_QUERY_ORD_SELECT_SQL_FAILURE	26000	An SQL exception occurred while querying the CUSTOMORDERVIEW view.
RC_QUERY_PAY_SELECT_SQL_FAILURE	26001	An SQL exception occurred while querying the CUSTOMPAYMENTVIEW view.
RC_QUERY_CRE_SELECT_SQL_FAILURE	26002	An SQL exception occurred while querying the CUSTOMCREDITVIEW view.
RC_QUERY_BAT_SELECT_SQL_FAILURE	26003	An SQL exception occurred while querying the CUSTOMBATCHVIEW view.
RC_QUERY_ACC_SELECT_SQL_FAILURE	26004	An SQL exception occurred while querying the CUSTOMACCOUNTVIEW view.

Chapter 9. CustomOffline Cassette messages

CustomOffline error messages follow:

CEPCustomOffline1000: The CustomOffline Cassette has started.

Severity: Information

Explanation: The Cassette is now accepting requests.

User Response: None

CEPCustomOffline1001: The CustomOffline Cassette has stopped.

Severity: Information

Explanation: The Cassette is no longer accepting requests.

User Response: None

CEPCustomOffline1002: This method has not been implemented yet. The method is *method_name***.**

Severity: Error

Explanation: Check with development as to the availability of this method if you have a need for it or use a different method.

User Response: Contact your IBM support representative.

CEPCustomOffline1003: The cassette's resource bundle ID does not match the ID passed by the framework. Expected ID is *expected_id* **and framework ID is** *framework_id***.**

Severity: Error

Explanation: None

User Response: Check with your developer for possible reasons as to why this happened.

CEPCustomOffline2000: An SQL exception occurred while selecting existing accounts from the CUSTOMACCOUNT table.

Severity: Error

Explanation: An SQL exception occurred while accessing the CUSTOMACCOUNT table. This could be due to an error connecting to or accessing the database, or due to an error in the content of the data.

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.

CEPCustomOffline2001: An SQL exception occurred while closing a query on the CUSTOMACCOUNT table.

Severity: Error

Explanation: An SQL exception occurred while closing CUSTOMACCOUNT table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline2003: An SQL exception occurred while inserting a row into the CUSTOMACCOUNT table. The merchant number is *merchant_number***, the Account number is** *account_number***, and the number of rows affected is** *rows***.**

Severity: Information

Explanation: This insertion is only supposed to insert one row, but more than one row was inserted. This condition should not happen but if it does, contact support.

User Response: None

CEPCustomOffline2004: An SQL exception occurred while inserting an account into the CUSTOMACCOUNT table. The merchant number is *merchant_number***, and account number is** *account_number***.**

Severity: Information

Explanation: An SQL exception occurred while inserting a row into the CUSTOMACCOUNT table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

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. Also make sure that the database user is authorized to access the WebSphere Commerce Payments database.

CEPCustomOffline2005: An SQL update should have changed only one row but multiple rows were affected. The merchant number is *merchant_number***, the account number is** *account_number***, and number of rows affected are** *number***.**

Severity: Information

Explanation: The update should have changed only one row but more than one row were affected.

User Response: None

CEPCustomOffline2006: An SQL exception occurred while updating an existing account in the CUSTOMACCOUNT table. The merchant number is *merchant_number***, and account number is** *account_number***.**

Severity: Error

Explanation: SQL exception occurred while reading a WebSphere Commerce Payments 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 could be the result of a disruption in communication between WebSphere Commerce Payments and the database server, or a discrepancy between the table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline2007: An SQL exception occurred while deleting a row from the CUSTOMACCOUNT table. The merchant number is *merchant_number***, the Account number is** *account_number***, and the number of rows affected is** *rows***.**

Severity: Error

Explanation: The deletion is only supposed to delete one row, but more than one row was deleted. This condition should not happen but if it does, contact support.

User Response: Contact your IBM support representative.

CEPCustomOffline2008: An SQL exception occurred while deleting an account from the CUSTOMACCOUNT table. The associated merchant number is *merchant_number* **and account number is** *account_number***.**

Severity: Error

Explanation: An SQL exception occurred while inserting a row into the CUSTOMACCOUNT table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

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. Also make sure that the database user is authorized to access the WebSphere Commerce Payments database.

CEPCustomOffline2009: Unable to close connection on port *port number* **to the CustomOffline CashRegister at** *hostname***.**

Severity: Error

Explanation: The socket connection that performs the communication between the cassette and the CustomOffline CashRegister failed to close successfully.

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

CEPCustomOffline2010: An attempt to retrieve a batch failed because the transaction did not contain a batch number. The merchant number is *merchant_number* **, the account number is** *account_number***, the order number is** *order_number***, and transaction number is** *transaction_number***.**

Severity: Error

Explanation: The transaction used in the request is not a valid one. This is not expected to happen on a production system.

User Response: Contact your IBM support representative.

CEPCustomOffline3000: An SQL query of CUSTOMBATCH table returned incorrect result.

Severity: Error

Explanation: An SQL query of CUSTOMBATCH table should have returned only one row but multiple rows were returned. The merchant number is *merchant_number* and batch number is *number*.

User Response: Contact your IBM support representative.

CEPCustomOffline3001: An SQL query of CUSTOMBATCH table returned incorrect result.

Severity: Error

Explanation: An SQL query of CUSTOMBATCH table should have returned only one row but multiple rows were returned. The merchant number is *merchant_number* and batch number is *number*.

User Response: Contact your IBM support representative.

CEPCustomOffline3002: An SQL exception occurred while querying a batch from the CUSTOMBATCH table. The merchant number is *merchant_number* and batch number is *number*.

Severity: Error

Explanation: An SQL exception occurred while reading the CUSTOMBATCH table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline3003: An SQL exception occurred while closing a query operation from the CUSTOMBATCH table.

Severity: Error

Explanation: An SQL exception occurred while closing the CUSTOMBATCH table query. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline3004: An SQL exception occurred while inserting a row into the CUSTOMBATCH table. The merchant number is *merchant_number*, the batch number is *batch_number*, and the number of rows affected is *rows*.

Severity: Error

Explanation: This insertion is only supposed to insert one row, but more than one row was inserted. This condition should not happen but if it does, contact support.

User Response: Contact your IBM support representative.

CEPCustomOffline3005: An SQL exception occurred while updating an existing account in the CUSTOMBATCH table. The merchant number is *merchant_number* and batch number is *batch_number*.

Severity: Error

Explanation: SQL exception occurred while inserting into the CUSTOMBATCH table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline3006: An SQL exception occurred while updating a row from the CUSTOMBATCH table. The merchant number is *merchant_number*, the batch number is *batch_number*, and the number of rows affected is *rows*.

Severity: Error

Explanation: The update is only supposed to affect one row, but more than one row was updated. This condition should not happen but if it does, contact support.

User Response: Contact your IBM support representative.

CEPCustomOffline3007: An SQL exception occurred while updating an existing batch in the CUSTOMBATCH table. The merchant number is *merchant_number* and the batch number is *batch_number*.

Severity: Error

Explanation: An SQL exception occurred while reading the CUSTOMBATCH table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline3008: An SQL exception occurred while deleting a batch from the CUSTOMBATCH table. The merchant number is *merchant_number* batch number is *number* and number of rows affected is *rows*.

Severity: Error

Explanation: An SQL delete of CUSTOMBATCH table should have returned only one row but multiple rows were returned.

User Response: Contact your IBM support representative.

CEPCustomOffline3009: An SQL exception occurred while deleting a batch from the CUSTOMBATCH table. The merchant number is *merchant_number* and batch number is *number*.

Severity: Error

Explanation: An SQL exception occurred while reading the CUSTOMBATCH table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline3010: The Supervisor could not retrieve an order referenced by the framework's batch payment list. The merchant number is *merchant_number*, the batch number is *batch_number*, the order number is *order_number*, and payment number is *payment_number*.

Severity: Error

Explanation: This is a sanity check in the processing of request. A user may not see this.

User Response: Contact your IBM support representative.

CEPCustomOffline3011: Supervisor could not retrieve a payment referenced by the framework's batch payment list. The merchant number is *merchant_number*, the batch number is *batch_number*, the order number is *order_number*, and the payment number is *payment_number*.

Severity: Error

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline3012: The batch number referenced by a payment does not match the batch number of the batch that contains the payment. The merchant number is *merchant_number*, the batch number is *batch_number*. the order number is *order_number*, and the payment number is *merchant_number*.

Severity: Error

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline3013: The Supervisor could not retrieve an order referenced by the framework's batch credit list. The merchant number is *merchant_number*, the batch number is *batch_number*. the order number is *order_number*, and the credit number is *credit_number*.

Severity: Error

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline3014: The Supervisor could not retrieve a credit referenced by the framework's batch credit list. The merchant number is *merchant_number*, the batch number is *batch_number*. the order number is *order_number*, and the credit number is *credit_number*.

Severity: Error

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline3015: The batch number referenced by a credit does not match the batch number of the batch that contains the credit. The merchant number is *merchant_number*, the batch number is *batch_number*, the order number is *order_number*, and the credit number is *credit_number*.

Severity: Error

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline3016: While performing a batch purge, the DepositReversal failed. Even though this payment has not been successfully purged, purging will continue. The merchant number is *merchant_number*, the batch number is *batch_number*, the order number is *order_number*, the payment number is *payment_number*, the primary return code is *prc*, and the secondary return code is *src*.

Severity: Warning

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline3017: While performing a batch purge, the Supervisor failed to retrieve this payment. Even though this payment has not been successfully purged, purging will continue. The merchant number is *merchant_number*, the batch number is *batch_number*, the order number is *order_number*, the payment number is *payment_number*, the primary return code is *prc*, and the secondary return code is *src*.

Severity: Warning

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline3018: While performing a batch purge, a RefundReversal failed. Even though this credit has not been successfully purged, purging will continue. The merchant number is *merchant_number*, the batch number is *batch_number*, the order number is *order_number*, the credit number is *credit_number*, the primary return code is *prc*, and the secondary return code is *src*.

Severity: Warning

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline3019: While performing a batch purge, the Supervisor failed to retrieve this credit. Even though this credit has not been successfully purged, purging will continue. The merchant number is *merchant_number*, the batch number is *batch_number*, the order number is *order_number*, the credit number is *credit_number*, the primary return code is *prc*, and the secondary return code is *src*.

Severity: Warning

Explanation: This is an internal test to assure that further processing does not fail.

User Response: Contact your IBM support representative.

CEPCustomOffline4000: An assertion failure occurred while querying a row from the CUSTOMORDER table. The merchant number is *merchant_number* and the order number is *order_number*.

Severity: Error

Explanation: An SQL query of the CUSTOMORDER table should have returned only one row but multiple rows were returned.

User Response: Contact your IBM support representative.

CEPCustomOffline4001: An assertion failure occurred while querying a row from the CUSTOMORDER table. The merchant number is *merchant_number* and the order number is *order_number*.

Severity: Error

Explanation: An SQL query of the CUSTOMORDER table should have returned one row but nothing was returned.

User Response: Contact your IBM support representative.

CEPCustomOffline4002: An SQL exception occurred while querying an existing order from the CUSTOMORDER table. The merchant number is *merchant_number* and the order number is *order_number*.

Severity: Error

Explanation: An SQL exception occurred while querying the CUSTOMORDER table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline4003: An SQL exception occurred while closing a query order from the CUSTOMORDER table.

Severity: Error

Explanation: An SQL exception occurred while closing the CUSTOMORDER table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline4004: An assertion failure occurred while inserting a row into the CUSTOMORDER table. The merchant number is *merchant_number***, the order number is** *order_number***, and the number of rows affected is** *rows***.**

Severity: Error

Explanation: An SQL insertion into the CUSTOMORDER table should have inserted one row but multiple rows were affected.

User Response: Contact your IBM support representative.

CEPCustomOffline4005: An SQL exception occurred while inserting a new order into the CUSTOMORDER table. The merchant number is *merchant_number* **and the order number is** *order_number***.**

Severity: Error

Explanation: An SQL exception occurred while inserting into the CUSTOMORDER table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline4006: An assertion failure occurred while updating the CUSTOMORDER table. The merchant number is *merchant_number***, the order number is** *order_number***, and the number of rows affected is** *rows***.**

Severity: Error

Explanation: An SQL update of the CUSTOMORDER table should have updated one row but affected multiple rows.

User Response: Contact your IBM support representative.

CEPCustomOffline4007: An SQL exception occurred while updating an existing order in the CUSTOMORDER table. The merchant number is *merchant_number* **and the order number is** *order_number***.**

Severity: Error

Explanation: An SQL exception occurred while updating the CUSTOMORDER table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline4008: An assertion failure occurred while removing a row from the CUSTOMORDER table. The merchant number is *merchant_number***, the order number is** *order_number***, and the number of rows affected is** *rows***.**

Severity: Error

Explanation: An SQL delete of the CUSTOMORDER table should have deleted one row but affected multiple rows.

User Response: Contact your IBM support representative.

CEPCustomOffline4009: An SQL exception occurred while deleting an order from the CUSTOMORDER table. The merchant number is *merchant_number* and the order number is *order_number*.

Severity: Error

Explanation: An SQL exception occurred while deleting an order from the CUSTOMORDER table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline4010: An assertion failure occurred while closing the CUSTOMORDER table. The merchant number is *merchant_number*, the order number is *order_number*, the batch number is *batch_number*, the payment number is *payment_number*, and the actual order state is *order_state*.

Severity: Error

Explanation: While closing a batch, it was discovered that the batch contained a payment whose order is in the wrong state.

User Response: Contact your IBM support representative.

CEPCustomOffline4011: An assertion failure occurred while closing a batch. The merchant number is *merchant_number*, the order number is *order_number*, the batch number is *batch_number*, the credit number is *credit_number*, and the actual order state is *order_state*.

Severity: Error

Explanation: While closing a batch, it was discovered that the batch contained a credit whose order is in the wrong state.

User Response: Correct the state and close the batch again.

CEPCustomOffline5000: An assertion failure occurred while querying the CUSTOMPAYMENT table. The merchant number is *merchant_number*, the order number is *order_number*, and the payment number is *payment_number*.

Severity: Error

Explanation: While querying for a payment, it returned multiple entries. This condition should not have occurred.

User Response: Contact your IBM support representative.

CEPCustomOffline5001: An assertion failure occurred while querying the CUSTOMPAYMENT table. The merchant number is *merchant_number*, the order number is *order_number*, and the payment number is *payment_number*.

Severity: Error

Explanation: While querying for a payment that should have returned one row, nothing was returned. This condition should not have occurred.

User Response: Contact your IBM support representative.

CEPCustomOffline5002: An SQL exception occurred while querying from the CUSTOMPAYMENT table. The merchant number is *merchant_number*, the order number is *order_number*, and the payment number is *payment_number*.

Severity: Error

Explanation: An SQL exception occurred while querying from the CUSTOMPAYMENT table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline5003: An SQL exception occurred while closing a query operation from the CUSTOMPAYMENT table.

Severity: Error

Explanation: An SQL exception occurred while closing from the CUSTOMPAYMENT table query. The SQL exception text describes the exception and provides

SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline5004: An assertion failure occurred while inserting a row into the CUSTOMPAYMENT table. The merchant number is *merchant_number*, the order number is *order_number*, the payment number is *payment_number*, and the number of rows affected is *rows*.

Severity: Error

Explanation: An SQL insertion into the CUSTOMPAYMENT table should have inserted one row but affected multiple rows.

User Response: Contact your IBM support representative.

CEPCustomOffline5005: An SQL exception occurred while inserting a new payment into the CUSTOMPAYMENT table.

Severity: Error

Explanation: An SQL exception occurred while inserting a new payment into from the CUSTOMPAYMENT table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline5006: An assertion failure occurred while updating a row in the CUSTOMPAYMENT table. The merchant number is *merchant_number*, the order number is *order_number*, the payment number is *payment_number*, and the number of rows affected is *rows*.

Severity: Error

Explanation: While updating a payment that should have affected one row, multiple rows were returned. This condition should not have occurred.

User Response: Contact your IBM support representative.

CEPCustomOffline5007: An SQL exception occurred while updating an existing payment in the CUSTOMPAYMENT table. The merchant number is *merchant_number*, the order number is *order_number*, and the payment number is *payment_number*.

Severity: Error

Explanation: An SQL exception occurred while updating the CUSTOMPAYMENT table. The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline5008: An assertion failure occurred while deleting a row from the CUSTOMPAYMENT table. The merchant number is *merchant_number*, the order number is *order_number*, the payment number is *payment_number*, and the number of rows affected is *rows*.

Severity: Error

Explanation: While deleting a payment that should have affected one row, multiple rows were returned. This condition should not have occurred.

User Response: Contact your IBM support representative.

CEPCustomOffline5009: An SQL exception occurred while deleting a payment from the CUSTOMPAYMENT table. The merchant number is *merchant_number*, the order number is *order_number*, and the payment number is *payment_number*.

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline5010: An assertion failure occurred while closing a batch. The merchant number is *merchant_number***, the order number is** *order_number***, the batch number is** *batch_number***, the payment number is** *payment_number***, and the actual credit state is** *credit_state***.**

Severity: Error

Explanation: While closing a batch, it was discovered that the batch contained a payment in the wrong state.

User Response: Contact your IBM support representative.

CEPCustomOffline6000: An assertion failure occurred while querying the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, and the credit number is** *credit_number***.**

Severity: Error

Explanation: While querying the CUSTOMCREDIT table, it was supposed to return one row but it returned multiple entries.

User Response: Contact your IBM support representative.

CEPCustomOffline6001: An assertion failure occurred while querying the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, and the credit number is** *credit_number***.**

Severity: Error

Explanation: While querying the CUSTOMCREDIT table, it should have returned one row but nothing was returned.

User Response: Contact your IBM support representative.

CEPCustomOffline6002: An SQL exception occurred while querying a credit from the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, and the credit number is** *credit_number***.**

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline6003: An SQL exception occurred while closing a query on the CUSTOMCREDIT table.

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline6004: An assertion failure occurred while querying the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, the credit number is** *credit_number***, and the number of rows affected is** *rows***.**

Severity: Error

Explanation: While querying the CUSTOMCREDIT table, it should have returned one row but multiple rows are returned.

User Response: Contact your IBM support representative.

CEPCustomOffline6005: An SQL exception occurred while inserting a new credit into the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, and the credit number is** *credit_number***.**

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

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

Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline6006: An assertion failure occurred while updating the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, the credit number is** *credit_number***, and the number of rows affected is** *rows***.**

Severity: Error

Explanation: While updating the CUSTOMCREDIT table, it should have updated one row but multiple rows are affected.

User Response: Contact your IBM support representative.

CEPCustomOffline6007: An SQL exception occurred while updating an existing credit in the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, and the credit number is** *credit_number***.**

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline6008: An assertion failure occurred while removing a row from the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, the credit number is** *credit_number***, and the number of rows affected is** *rows***.**

Severity: Error

Explanation: While deleting a row from the CUSTOMCREDIT table, it should have deleted one row but multiple rows were affected.

User Response: Contact your IBM support representative.

CEPCustomOffline6009: An SQL exception occurred while deleting a credit from the CUSTOMCREDIT table. The merchant number is *merchant_number***, the order number is** *order_number***, and the credit number is** *credit_number***.**

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline6010: An assertion failure occurred while closing a batch. The merchant number is *merchant_number***, the batch number is** *batch_number***, the order number is** *order_number***, the credit number is** *credit_number***, and the actual credit state is** *credit_state***.**

Severity: Error

Explanation: While closing a batch, it was discovered that the batch contained a credit in the wrong state.

User Response: Contact your IBM support representative.

CEPCustomOffline7000: An SQL exception occurred while processing the output from query on cassette orders. The table queried is CUSTOMORDER.

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline7001: An SQL exception occurred while processing the output from query on cassette payments. The table queried is CUSTOMPAYMENT.

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline7002: An SQL exception occurred while processing the output from query on cassette credits. The table queried is CUSTOMCREDIT.

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline7003: An SQL exception occurred while processing the output from query on cassette batches. The table queried is CUSTOMBATCH.

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

CEPCustomOffline7004: An SQL exception occurred while processing the output from query on cassette accounts. The table queried is CUSTOMACCOUNT.

Severity: Error

Explanation: The SQL exception text describes the exception and provides SQL 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 table definition and the definition expected by WebSphere Commerce Payments.

User Response: Refer to the SQL state information to get specific details about the nature of the problem. Test the database server connection and verify that the table definition matches the definition expected by WebSphere Commerce Payments.

The specified order was not found.

Severity:

Explanation: Return code pair - PRC_DATABASE_ERROR, RC_ORDER_SELECT_ORDER_MISSING.

User Response:

Appendix. Notices

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

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

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

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

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

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

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

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

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

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

IBM Corporation
Department TL3B/Building 503
P.O. Box 12195
3039 Cornwallis Road
RTP, NC 27709-2195
U.S.A.

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, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

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

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

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

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

SET and the SET logo are trademarks of SET Secure Electronic Transactions LLC.

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

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

Glossary

A

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

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

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

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

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

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

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

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

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

B

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

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

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

batch. (1) A collection of payment transactions, such as captures, credits, capture reversals, and credit reversals, processed as a group. A batch is submitted as a single unit to the Acquirer's financial system. Business guidelines regarding the use of batch processing are developed by credit acquiring institutions. Merchants also establish policies that align with these guidelines. (2) Within IBM e-commerce, one of the fundamental WebSphere Commerce Payments objects is the Batch. A Batch is an object with which Payment and Credit objects are associated. Transfer of funds is to occur when the batch is closed. (3) A group of records or data processing jobs brought together for processing or transmission.

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

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

C

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

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

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

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

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

- signature
- key encipherment
- certificate signature
- CRL signature

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

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

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

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

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

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

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

D

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

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

DTD. See document type definition.

E

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

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

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

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

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

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

F

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

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

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

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

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

G

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

H

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

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

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

HTML. See Hypertext Markup Language.

HTTP. See Hypertext Transfer Protocol.

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

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

I

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

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

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

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

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

J

J2EE. (Java 2 Enterprise Edition) Any deployable unit of J2EE functionality. This can be a single module or a group of modules packaged into an .ear file with a J2EE application deployment desriptor.

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

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

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

K

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

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

L

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

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

M

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

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

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

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

N

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

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

O

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

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

order amount. The amount of the order.

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

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

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

P

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

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

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

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

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

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

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

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

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

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

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

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

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

R

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

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

1. Multiple payments for a fixed amount
2. Repeated billings

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

S

sale. In the credit card world, a sale occurs when a transaction is authorized and marked for capture all at once rather than using a two-step process.

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

Secure Electronic Transaction. See SET Secure Electronic Transaction.

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

server. (1) A functional unit that provides services to one or more clients over a network. (2) A computer that provides shared services to other computers over a network; for example, a file server, a print server, or a mail server.

servlet. An application program, written in the Java programming language, that is executed on a Web server. A reference to a servlet appears in the markup for a Web page, in the same way that a reference to a graphics file appears. The Web server executes the servlet and sends the results of the execution (if there are any) to the Web browser. Contrast with applet.

SET. See SET Secure Electronic Transaction.

SET Secure Electronic Transaction™. An industry standard developed for secure credit card and debit card payments over open networks such as the Internet.

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

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

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

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

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

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

SSL. See Secure Sockets Layer.

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

T

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

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

U

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

URL. See uniform resource locator.

V

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

W

wallet. Software that enables a user to make approved payments to authenticated merchants over public networks and to manage payment card accounts and purchases.

WAR file. A Web Archive (WAR) file is a Java archive file used to store one or more of the following: servlets; JavaServer Pages (JSP) files; utility classes; static documents (such as HTML files, images and sound); client-side applets, beans and classes; descriptive meta-information. Its standard file extension is .war. WAR files are used to package Web modules.

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

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

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

Web site. A Web server that is managed by a single entity (an organization or an individual) and contains information in hypertext for its users, often including hypertext links to other Web sites. Each Web site has a home page. In a uniform resource locator (URL), the Web site is indicated by the fully qualified domain name. For example, in the URL `http://www.as400.ibm.com/icswg.html` , the Web site for IBM AS/400 is indicated by `www.as400.ibm.com` , which is the fully qualified domain name.

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

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

X

XML. A standard metalanguage for defining markup languages that was derived from and is a subset of SGML. XML omits the more complex and less-used parts of SGML and makes it much easier to write applications to handle document types, to author and manage structured information, and to transmit and share structured information across diverse computing systems. XML is defined by the World Wide Web Consortium (W3C).

Index

Printed in U.S.A.