



IBM Software Group

WebSphere® Commerce V6

Payment administration



@business on demand.

© 2007 IBM Corporation
Updated September 18, 2007

Welcome to the WebSphere Commerce V6 presentation. This presentation describes the new payment components in WebSphere Commerce.

Unit objectives

- Describe the WebSphere Commerce V6 payment architecture
- Define payment rules
- Describe how payment rules are configured

This presentation introduces the new payment architecture and describes the use and configuration of payment rules.

Payment processing in WebSphere Commerce

- WebSphere Commerce payments (New)
- WebSphere Commerce payments (Old)
- Custom payment implementation

New WebSphere Commerce payments is the recommended solution for payment processing!

There are three ways to manage payments in WebSphere Commerce V6.

It is recommended to use the new implementation of WebSphere Commerce payments. This new payment subsystem implementation is completely integrated into the WebSphere Commerce runtime environment. It uses payment plug-ins and does not require additional configuration at instance creation time. Since the payment system is integrated into the WebSphere Commerce EAR, it can be cloned. The starter stores have been updated to use this new payment implementation by default.

Alternatively, it is also possible to continue using the multi-payment framework provided by the WebSphere Commerce payments component. The payments component must be selected during the custom installation and a payments instance must be created. A special payment plug-in called the WebSphere Commerce Payments Cassette plug-in is provided with WebSphere Commerce and must be configured to use with the traditional WebSphere Commerce payments component.

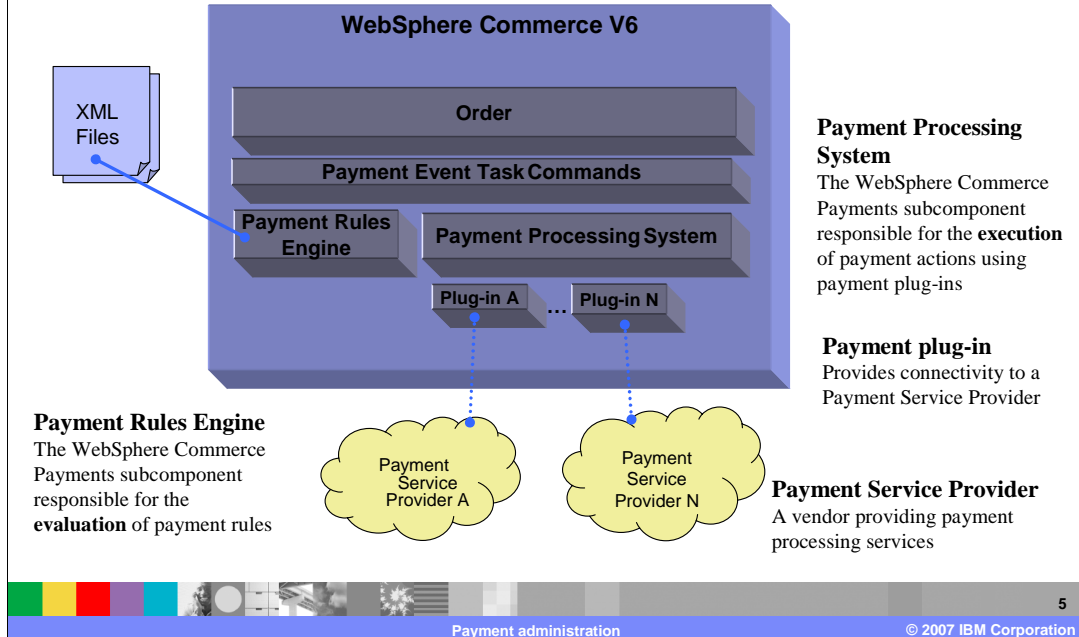
Finally, you can customize the V5 payment policy commands and work directly with an API provided by a payment service provider instead of migrating to the new payment infrastructure. Although this type of customization is still supported, IBM® is encouraging all customers to migrate to the new architecture.

WebSphere Commerce Payments

- Provides payment processing logic to other WebSphere Commerce components, mainly order and return
- Provides an architecture to integrate with payment service providers
- Provides new features due to improved runtime and architecture
 - ▶ J2EE compliant
 - ▶ Supports multiple payment instructions per order
 - ▶ Supports new order scenarios
 - ▶ Configurable payment rules
 - ▶ Payment plug-ins connect to back-end payment provider

WebSphere Commerce Payments provides payment processing logic to other WebSphere Commerce components and allows integration with payment services providers. Due to the improved runtime and architecture, some new features have become available, such as support for multiple payment instructions per order, support for new order scenarios, configurable payment rules, and so on. You can find the menus for the administration of payments using WebSphere Commerce Payments in WebSphere Commerce Accelerator.

WebSphere Commerce Payments in V6



This slide shows a high-level overview of the new payment architecture. The order component contains only order logic. New task commands call the Payment Rules engine to determine which rules should apply; the task commands then call the Payment Processing System that executes the applicable rule. The payment processing system is responsible for routing the required payment action to the payment plug-in. The plug-in provides the integration point to the back-end payment service provider.

Payments – old versus new

WebSphere Commerce V5	WebSphere Commerce V6
Payment cassettes (complex API) integrate with external payment provider.	Payment plug-ins (simple API) integrate with external payment provider.
Separate application is not J2EE compliant.	Integrated with WebSphere Commerce application which is J2EE compliant
Separate database must be managed.	Payment database tables are contained within the WebSphere Commerce database.
Payment actions are hard-coded in the application code.	Actions for payment events are configurable through XML files
	Supports new features like multiple payment instructions per order.

6

The table compares the old Payment Manager with the new payment infrastructure. The new implementation has a number of advantages over the old Payment Manager. It is J2EE compliant, provides simplified plug-in development, does not require a separate database, allows configuring payment events, and supports new features.

Payment plug-ins

- Simple offline plug-in
 - ▶ Enables payments to be processed offline or manually
 - ▶ Supports these payment methods out of the box:
 - VISA
 - MasterCard
 - AMEX
 - Cash on delivery
 - BillMeLater
 - PayLater
- Line of credit plug-in
 - ▶ Enables line of credit as a payment method
 - ▶ Does not connect to an account receivable system but can be used by customers as an example when writing their own line of credit plug-ins
- WebSphere Commerce Payments cassette plug-in
 - ▶ Enables backward compatibility with cassettes not yet migrated to the payment plug-in architecture
 - ▶ Supports new WebSphere Commerce Payments features

There are three payment plug-ins provided out of the box.

The **Simple Offline Plug-in** replaces the OfflineCard and CustomOffline cassettes. It supports the payment methods shown on the slide and can be customized to support additional offline payment methods. Transactions using this plug-in are recorded and maintained in the WebSphere Commerce database. A customer service representative can use the WebSphere Commerce Accelerator to approve or reject offline financial transactions manually.

The **Line of Credit Plug-in** is provided as a sample.

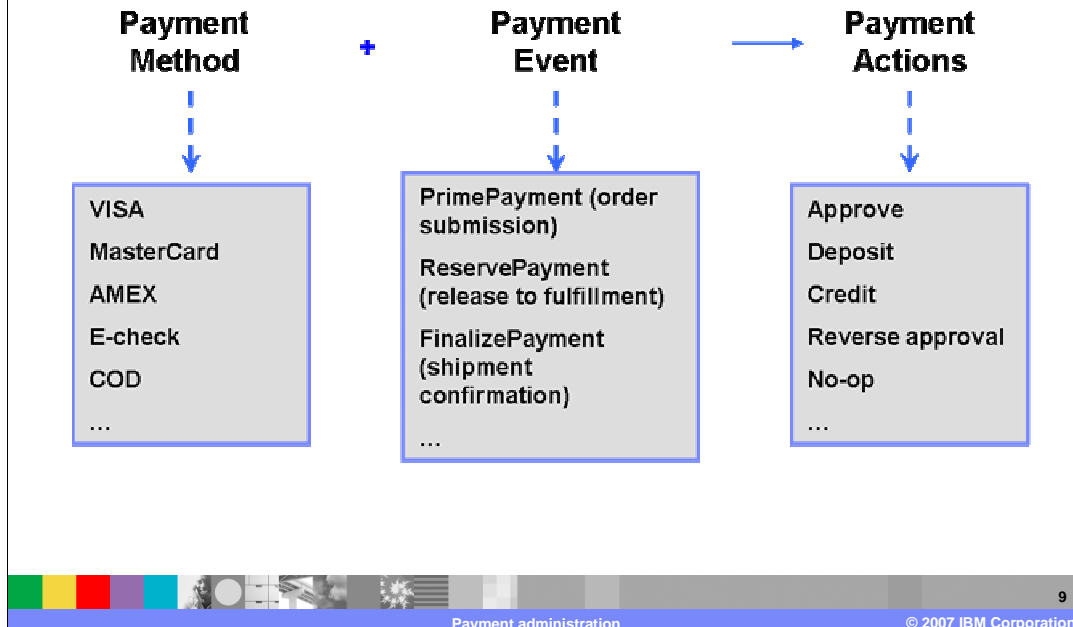
The **WebSphere Commerce Payments Cassette Plug-in** allows you to continue using Payment Manager for payment processing services along with any of your existing cassettes. This is intended for backwards compatibility while migrating to the new payment architecture. This plug-in supports the new WebSphere Commerce features such as multiple releases per order and multiple payment methods per order.

Terms and acronyms

- **Payment method**
 - ▶ A means by which payments are made
 - VISA, MasterCard, AMEX, e-Check, cash on delivery...
 - ▶ References a payment system defined by the payment processing system
- **Payment instruction**
 - ▶ An instance of a payment method with specific details bound to it
 - payment method=VISA + card number=4111111111111111 + expiry date=01/07 + cardholder name=John Doe + amount=100USD + ...
- **Payment action**
 - ▶ An operation against a payment instruction
 - approve, reverse approval, deposit, reverse deposit, refund, reverse refund, validate...
- **Payment rules**
 - ▶ Configurable rules that determine what payment actions should be performed (authorize payment) at specific times in the order and return merchandise authorizations life cycles (order submission)

This slide shows the list of terms and acronyms that are commonly used in the WebSphere Commerce Payments infrastructure.

Payment rules



Payment rules are a key part of the new payment architecture. There are certain payment actions that are part of any payment processing system. For example, payments need to be approved. The money needs to be deposited. Credits need to be issued. Depending on the payment method used, these actions may have to take place at different points in the order life cycle. For example, in the United States it is not legal to deposit a credit card payment until the product has been shipped. With e-checks, there are different rules. For example, payment can be captured upfront.

Payment rules configuration

File	Configuration Type	Comments
PaymentRules	Payment	Read-only file
PaymentSystemPluginMapping	Payment	Can be changed if required.
PaymentMethodConfigurations	Payment	Can be changed if required.
RefundMethodConfigurations	Refund	Can be changed if required.
AVSRules	Payment	Can be changed if required
CoreCancelOrderPaymentActions	Payment	Should not be changed
CoreEditActions	Payment	Should not be changed
CoreEditReversePaymentActions	Payment	Should not be changed
CorePaymentActions	Payment	Can be changed if required
EDPGlobalConfigs	System wide	Can be changed if required

10

Payment administration

© 2007 IBM Corporation

There are multiple configuration files involved with Payment Rules. The contents of the **PaymentRules** file cannot be changed. Also, the contents of the files whose names start with the keyword “Core” cannot be changed. The **PaymentRules** file is used to configure the payment rules for a store or store group. The files whose names start with the “Core” keyword determine the core behavior of payment actions when a payment attribute of an order is changed in some way. The attribute can be changed when the order is canceled, or payment is reversed, or the order amount changes due to an edit action.

In general, these files will only be changed when configuring a new plug-in.

The **AVSRules** file determines the Payment Rules responses when certain Address Verification System codes are received.

The **PaymentSystemPluginMapping** file maps payment systems to payment plug-ins. This file is used by the payment processing service to determine what plug-ins should be used for a particular payment system name in a Payment Rules configuration. You need to add new supported plug-ins to this file.

The **CorePaymentActions** file determines the core behavior of the payment actions. You should not have to update or create this file when configuring payment processing for your store if you are using the payments function provided with WebSphere Commerce.

Payment rules and mappings

```

<PaymentRule name="Early Approval">
  <PrimePaymentEvent targetState="APPROVED" />
  <ReservePaymentEvent targetState="APPROVED" />
  <FinalizePaymentEvent targetState="DEPOSITED" />
</PaymentRule>

<PaymentRule name="Early Deposit">
  <PrimePaymentEvent targetState="DEPOSITED" />
  <ReservePaymentEvent targetState="DEPOSITED" />
  <FinalizePaymentEvent targetState="DEPOSITED" />
</PaymentRule>

```

What is the rule?

PaymentRules.xml

Which methods use it?

```

<Mapping      paymentMethod="VISA"
              paymentConfiguration="CreditCardOffline"
              paymentActionRule="Early Approval" />

<Mapping      paymentMethod="Check"
              paymentConfiguration="CheckOffline"
              paymentActionRule="Early Deposit" />

```

PaymentSystemPluginMapping.xml

11

Payment administration

© 2007 IBM Corporation

The **PaymentRules** file defines what payment actions should be triggered when an order passes through different stages. The **PaymentSystemPluginMapping** file defines which payment methods use a particular payment rule. The PaymentRules file should not be modified. In this example, credit cards use the Early Approval rule and checks use the Early Deposit rule. The mappings file also defines which payment configuration a particular method is associated with.

Payment method configurations

```
<PaymentMethodConfiguration
    name="CreditCardOffline"
    paymentSystemName="SimpleOffline"
    systemEditable="true"
    humanEditable="true"
    refundAllowed="true"
    minimumAmount="0"
    maximumAmount="Unbounded"
    priority="MEDIUM"
    partiallyConsumable="true" />
```

PaymentMethodConfigurations.xml

On the previous slide, VISA payment method was associated with the CreditCardOffline payment method configuration. The **PaymentMethodConfigurations** file maps the payment method to the back-end payment service provider that will do the processing. In this case, the payment system is SimpleOffline. In other cases, this might be VisaNet, Paymentech, or some other payment service provider.

Other configuration properties for the payment method can also be defined in this file.

Payment systems and plug-ins

```

<PaymentSystemName name="SimpleOffline" >
  <Mapping paymentConfigurationId="default" pluginName="SimpleOffline" >
    <Keyword name="cc_cvc" mask="-" plain="0" removeAfterApproval="true"/>
    <Keyword name="cc_nameoncard" mask="*" plain="0" removeAfterApproval="true"/>
    <Keyword name="account" mask="*" plain="-5" searchable="true"/>
  </Mapping>
</PaymentSystemName>

<PaymentSystemName name="LOCPlugin" >
  <Mapping paymentConfigurationId="default" pluginName="LOCPlugin" >
  </Mapping>
</PaymentSystemName>

<PaymentSystemName name="Paymentech" >
  <Mapping paymentConfigurationId="default" pluginName="WCPaymentsPlugin" >
    <Property name="ProfileName" value="WCPPlugin_Paymentech"/>
    <Property name="SupportsOnlineTransaction" value="yes"/>
    <Keyword name="cc_cvc" mask="-" plain="0" removeAfterApproval="true"/>
    <Keyword name="cc_nameoncard" mask="*" plain="0" removeAfterApproval="true"/>
    <Keyword name="account" mask="*" plain="-5" searchable="true"/>
  </Mapping>
</PaymentSystemName>

```

PaymentSystemPluginMapping.xml

So far, you have seen that a credit card like Visa or Mastercard uses the CreditCardOffline configuration. That configuration is mapped to a payment service provider called SimpleOffline. In the **PaymentSystemPluginMapping** file, the payment service provider name is mapped to the plug-in used for communication between the system and WebSphere Commerce.

Multiple payment configurations can be defined to allow different stores and store groups to support different combinations of payment methods. The payment configuration ID specifies which payment configuration this mapping is applicable for. The default is "default". The default is used by all stores. The PaymentSystemName and pluginName do not have to match. The PaymentSystemName is the name of the payment service provider; the pluginName is the name of the plug-in that interfaces with the payment service provider.

Payment administrators

- No separate payment administrator roles
 - ▶ Seller
 - ▶ Sales manager
 - ▶ Customer service representative
 - ▶ Customer service representative supervisor

All roles listed on the slide have the ability to manage orders, returns, and payments for a customer. No separate configuration is necessary. A user who is assigned one of these roles will automatically have the ability to work with payments.

Troubleshooting payments

- Payment rules engine
 - ▶ Trace component: com.ibm.websphere.commerce.WC_EDP
- Payment processing engine
 - ▶ Trace component: com.ibm.websphere.commerce.WC_PPC
 - ▶ Trace component: com.ibm.websphere.commerce.WC_PPC_PLUGIN
- Plug-ins
 - ▶ Trace component: com.ibm.websphere.commerce.WC_PPC_SIMPLEOFFLINE
 - ▶ Trace component: com.ibm.websphere.commerce.WC_PPC_LOCPPLUGIN
 - ▶ Trace component: com.ibm.websphere.commerce.WC_PPC_WCPPLUGIN

Information regarding a payment problem might be found in a variety of places besides log files.

If failures occur with a payment back-end system, tickler messages can be generated. When investigating problems, customer service representatives should pay particular attention to any tickler messages that show up in the tickler queue for **order** or **RMA** and take the appropriate action.

Customer service representatives can obtain information about the overall payment status of an order, payment instruction status, and payment history by displaying order details through the WebSphere Commerce Accelerator or the IBM Sales Center application. The information about the activity includes the store ID and order ID, the timestamp of the operation and result, and the amount and type of the operation.

A Payment Rules transaction is committed at the end of the order command execution and depends on the order command execution result, regardless of the financial result of the payment action. Every payment action is recorded in an internal payment activity history table. If a payment action has been attempted, the Payment Rules service keeps a record of what has been performed for the order. A record will be created in the database if a payment action fails. However, if an exception (other than financial failure) occurs, the transaction will roll back and there will be no record created in the Payment Rules database table.

Compatibility

- **WebSphere Commerce Payments Cassette Plug-in**
 - ▶ For customers using cassettes that are not yet migrated to the payment plug-in architecture
 - ▶ Limitations:
 - Requires Payment Manager
- **DoPaymentActionsCompatiblePolicyCmdImpl**
 - ▶ An alternative implementation of the DoPaymentActions policy command that calls the V5 DoXXX payment policy commands instead of the payment processing system
 - ▶ For customers who have customized the V5 DoXXX payment policy commands to integrate directly with payment service providers.
 - ▶ Limitations:
 - Supports only one payment instruction per order (same as V5)
 - ▶ Customers are encouraged to migrate their code to the payment plug-in architecture

The WebSphere Commerce Payments Cassette Plug-in allows you to continue using Payment Manager for payment processing services along with any of your existing cassettes. This is intended for backwards compatibility while migrating to the new payment architecture.

If you have customized the V5 payment policy commands to integrate directly with payment service providers, you can use DoPaymentActionsCompatiblePolicyCmdImpl until you have the opportunity to migrate to the new plug-in architecture. This command is an alternative implementation of the DoPaymentActions policy command that calls the V5 payment policy commands instead of the V6 payment processing system. If using this command, you will not be able to use some of the new order subsystem features such as supporting multiple payment instructions per order.

Recommended courses

Formal education exists for this product and you can find information on recommended training paths and certification tests at these locations:

- Application developer for WebSphere Commerce V6
<http://www-304.ibm.com/ict03001c/services/learning/ites.wss/us/en?pageType=page&c=a0011792>
- Business user for WebSphere Commerce V6
<http://www-304.ibm.com/ict03001c/services/learning/ites.wss/us/en?pageType=page&c=a0011793>
- System administrator for WebSphere Commerce V6
<http://www-304.ibm.com/ict03001c/services/learning/ites.wss/us/en?pageType=page&c=a0011794>



IBM provides training paths for the skill or certification you want to explore.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject= Feedback about wcs60_PaymentAdministration.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20wcs60_PaymentAdministration.ppt)



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM WebSphere

J2EE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.