

IBM WebSphere Commerce



# Security Guide

*Version 54*



IBM WebSphere Commerce



# Security Guide

*Version 54*

**Note:**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 103.

**First Edition, Sixth Revision (January 2004).**

This edition applies to version 5.4 of IBM WebSphere Commerce and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. You can send your comments by any one of the following methods:

1. Electronically to either of the network IDs listed below. Be sure to include your entire network address if you wish a reply.

Internet: [torrcf@ca.ibm.com](mailto:torrcf@ca.ibm.com)

IBMLink: [toribm\(torrcf\)](mailto:toribm(torrcf)@ca.ibm.com)

2. By FAX, use the following numbers:

United States and Canada: 416-448-6161

Other countries: (+1)-416-448-6161

3. By mail to the following address:

IBM Canada Ltd. Laboratory

B3/KB7/8200/MKM

8200 Warden Avenue

Markham, Ontario, Canada L6G 1C7

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

© Copyright International Business Machines Corporation 2002. All rights reserved.

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

---

# Contents

## Preface . . . . . v

Updates to this book . . . . .	v
Navigating through this document. . . . .	vi
Ongoing security assessment. . . . .	vi
Security improvements in WebSphere Commerce 5.4 . . . . .	vi
Enhancements for the Site Administrator . . . . .	vii
Enhancements for the System Administrator . . . . .	viii
Enhancements for the WebSphere Commerce Programmer . . . . .	viii
Security improvements in WebSphere Commerce Suite 5.1 Pro Edition . . . . .	ix
General Security Enhancements . . . . .	ix
Session Management . . . . .	x
Authentication. . . . .	x
Logging . . . . .	x
Conventions used in this book . . . . .	x
Where to find more information . . . . .	xi

---

## Part 1. WebSphere Commerce security model . . . . . 1

### Chapter 1. Introduction to the WebSphere Commerce security model . . . . . 3

Overview . . . . .	3
What is authentication?. . . . .	3
What is authorization?. . . . .	3
What are access control policies?. . . . .	3
What is an audit trail? . . . . .	4
What is confidentiality?. . . . .	4

### Chapter 2. Authentication . . . . . 5

WebSphere Commerce authentication model. . . . .	5
Challenge mechanisms . . . . .	6
Authentication mechanisms . . . . .	7
User registry . . . . .	7
Credentials . . . . .	7
WebSphere Commerce token . . . . .	7
WebSphere Application Server LTPA token . . . . .	8
Single sign-on . . . . .	8
Authentication policies . . . . .	8
Account policies . . . . .	8
Other authentication-related policies . . . . .	9
Session policies . . . . .	10

### Chapter 3. Authorization (Access Control) . . . . . 11

Organizational hierarchy . . . . .	11
Root organization . . . . .	12
Organizations (seller) . . . . .	13
Organizations (buyer) . . . . .	13
Roles . . . . .	13
Site operations . . . . .	14
Site and content development . . . . .	14
Logistics and operations . . . . .	14

Product management . . . . .	15
Sales management . . . . .	16
Marketing management . . . . .	16
Organizational management. . . . .	17
Access control policy . . . . .	17
Elements of an access control policy . . . . .	17
Access control policy concepts . . . . .	18
Resource and policy ownership. . . . .	23
Types of access control policies . . . . .	23
Levels of access control . . . . .	24
How access control prevents unauthorized actions . . . . .	26
Checking for authorization before performing a user-initiated action . . . . .	27
Using access control . . . . .	27
Evaluating access control policies . . . . .	27
Organizational hierarchy . . . . .	28
Users . . . . .	28
Roles . . . . .	28
Access Groups . . . . .	28
Documents . . . . .	28
Evaluating standard policies. . . . .	28
Evaluating template policies. . . . .	31

---

## Part 2. WebSphere Commerce site administrator security tasks . . . . . 33

### Chapter 4. Enhancing site security. . . . . 35

Views for security . . . . .	36
Login timeout . . . . .	36
Password invalidation. . . . .	36
Password protected commands. . . . .	37
Cross site scripting protection . . . . .	38
Enabling login timeout . . . . .	38
Activating password invalidation . . . . .	39
Enabling password protected commands . . . . .	39
Updating encrypted data . . . . .	40
Enabling cross site scripting protection . . . . .	41
Enabling access logging . . . . .	43
Setting up an account policy. . . . .	44
Setting up a password policy . . . . .	45
Setting up an account lockout policy . . . . .	46
Launching a security check . . . . .	47
Configuration Manager PDI Encrypt field . . . . .	48

### Chapter 5. Enabling WebSphere Application Server security . . . . . 49

Before you begin . . . . .	49
Enabling security with an LDAP user registry . . . . .	49
Enabling security with an operating system user registry. . . . .	53
Disabling WebSphere Commerce EJB security . . . . .	55
WebSphere Commerce security deployment options . . . . .	55

### Chapter 6. Session management . . . . . 57

Cookie based session management . . . . .	57
Using cookies for session management . . . . .	58
URL rewriting . . . . .	59
Using URL rewriting session management . . . . .	59
Writing JSP templates for URL rewriting. . . . .	59
<hr/>	
<b>Part 3. System administrator security tasks . . . . .</b>	<b>61</b>
<hr/>	
<b>Chapter 7. Setting and changing passwords . . . . .</b>	<b>63</b>
Quick reference to user IDs, passwords and Web addresses . . . . .	63
Changing the Configuration Manager Password . . . . .	66
Setting Your IBM HTTP Server Administrator Password . . . . .	67
Changing Your SSL Key File Password . . . . .	67
Generating WebSphere Commerce encrypted passwords. . . . .	67
Generating Payment Manager encrypted passwords . . . . .	68
<hr/>	
<b>Chapter 8. Enabling SSL for production with IBM HTTP Server . . . . .</b>	<b>69</b>
About security . . . . .	69
Configuring a security key file for production . . . . .	69
Request a secure certificate from a certifying authority . . . . .	71
Equifax users. . . . .	71
VeriSign users . . . . .	71
Receive and set your production key file as the current key file . . . . .	71
Test the production key file . . . . .	72
SSL Consideration for Payment Manager . . . . .	72
Enabling SSL on the IBM HTTP Server (iSeries) . . . . .	73
Using SSL with Payment Manager. . . . .	73
+ Enabling the SSL Accelerator option . . . . .	74

<b>Chapter 9. Enabling SSL for IBM SecureWay Directory Server (LDAP) . . . . .</b>	<b>75</b>
Set up SecureWay . . . . .	75
WebSphere Commerce. . . . .	75

<b>Chapter 10. Single sign-on . . . . .</b>	<b>77</b>
Prerequisites . . . . .	77
Enabling single sign-on . . . . .	77

<b>Part 4. WebSphere Commerce developer security tasks . . . . .</b>	<b>79</b>
--	-----------

<b>Chapter 11. Access control . . . . .</b>	<b>81</b>
Understanding access control . . . . .	81
Overview of resource protection in WebSphere Application Server . . . . .	81
Introduction to WebSphere Commerce access control policies . . . . .	83
Types of access control . . . . .	89
Access control interactions . . . . .	91
Protectable interface . . . . .	93
Groupable interface. . . . .	94
Finding more information about access control . . . . .	94
Implementing access control. . . . .	94
Identifying protectable resources . . . . .	94
Implementing access control in enterprise beans . . . . .	95
Implementing access control in data beans . . . . .	96
Implementing access control in controller commands. . . . .	97
Implementing access control policies in views . . . . .	99

<b>Part 5. Appendixes . . . . .</b>	<b>101</b>
-------------------------------------	------------

<b>Notices . . . . .</b>	<b>103</b>
Trademarks . . . . .	105

---

## Preface

This document describes the security features of WebSphere Commerce 5.4 and how to configure these features.

It details WebSphere Commerce security issues and features such as authentication, authorization, and access control policies. The objective of this document is to provide the persons responsible for security at your site (which likely includes a system administrator or WebSphere Commerce site administrator) with a comprehensive document to enable them to reliably secure a WebSphere Commerce production site.

The intended audience for this document is the chief security officer or the security administrator for a WebSphere Commerce site.

Note that many sections of this Guide have been derived from other documents in the WebSphere Commerce 5.4 information library such as the WebSphere Commerce 5.4 online help, the *WebSphere Commerce 5.4 Installation Guide*, and the *WebSphere Commerce 5.4 Programmer's Guide*. Specifically:

- The information in Chapter 3, "Authorization (Access Control)," on page 11 is also documented in the *WebSphere Commerce 5.4 Access Control Guide*.
- The information in Chapter 4, "Enhancing site security," on page 35 and Chapter 6, "Session management," on page 57 is also documented in the WebSphere Commerce 5.4 online help. The information in Chapter 5, "Enabling WebSphere Application Server security," on page 49 is also documented in the *WebSphere Commerce 5.4 Installation Guide*.
- The information in Part 3, "System administrator security tasks," on page 61 is also documented in the *WebSphere Commerce 5.4 Installation Guide*.
- The information in Part 4, "WebSphere Commerce developer security tasks," on page 79 is also documented in the *WebSphere Commerce 5.4 Programmer's Guide*.

### Important

This document covers only WebSphere Commerce security issues related to deploying an e-commerce site. Issues relating to vulnerabilities of your operating system are not covered. You should consult with your operating system vendor to determine the appropriate measures that you should take to secure your operating system.

---

## Updates to this book

The most recent version of this document is available as a PDF file from the Technical Library section of the following WebSphere® Commerce Web pages:

- Business Edition:  
[http://www.ibm.com/software/webservers/commerce/wc\\_be/lit-tech-general.html](http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html)
- Professional Edition:  
[http://www.ibm.com/software/webservers/commerce/wc\\_pe/lit-tech-general.html](http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html)

Updates from the last version of this book are identified by revision characters contained in the margin. This book uses the following conventions for revision characters:

- The "+" (plus) character identifies updates that have been made in the current revision of this book.
- The "|" (split vertical bar) character identifies cumulative updates that have been made in all previous revisions of this book.

To learn about last-minute changes to the product, see the current product README file, also available from the WebSphere Commerce Web site.

---

## Navigating through this document

This document is divided into the following parts:

- Part 1, "WebSphere Commerce security model," on page 1 discusses the WebSphere Commerce security model and provides a conceptual overview of WebSphere Commerce security. This part will be of interest to anyone that wants a general overview of WebSphere Commerce security or to plan for security at a WebSphere Commerce site.
- Part 2, "WebSphere Commerce site administrator security tasks," on page 33 discusses WebSphere Commerce site administration tasks pertaining to site security. This part will be of interest to anyone performing site administration tasks pertaining to site security.
- Part 3, "System administrator security tasks," on page 61 discusses WebSphere Commerce system administration tasks pertaining to system security. This part will be of interest to anyone performing system administration tasks and that is concerned with system security.
- Part 4, "WebSphere Commerce developer security tasks," on page 79 discusses WebSphere Commerce access control from a developer's standpoint. This part will be of interest to anyone wanting to understand access control concepts implementing access control policies in their code.

---

## Ongoing security assessment

The WebSphere Commerce product lines continually undergo security analysis from an independent group of IBM® Security experts. These experts perform security analysis from the point of view of a user with only access to WebSphere Commerce through a browser to the more privileged users that have an account on the same system that WebSphere Commerce server is running. The feedback from the security experts' analysis is used to continually improve the security of WebSphere Commerce.

---

## Security improvements in WebSphere Commerce 5.4

The following section lists the security enhancements in WebSphere Commerce 5.4 relative to WebSphere Commerce Suite 5.1. Most of these enhancements were made in the WebSphere Commerce Business Edition 5.1 release. These enhancements are generally applicable to the:

- WebSphere Commerce site administrator
- System administrator
- WebSphere Commerce developer

Note that sometimes these roles are interchangeable.



## Enhancements for the Site Administrator

The following are WebSphere Commerce 5.4 security enhancements that are generally targeted to a site administrator:

### Access control

- **Access control framework** — A key enhancement is that a new access control framework has been implemented in WebSphere Commerce 5.4. This new framework uses access control policies to determine if a given user is permitted to perform a given action on a given resource. The new access control framework provides fine-grained access control. It works in conjunction with, but does not replace the access control provided by the WebSphere Application Server. The new access control framework is described in detail in Chapter 11, “Access control,” on page 81.

The new access control framework enhances the previous access control in the following ways:

#### It is expressive...

It captures the intent of a large variety of access policies. The framework is generic so that it can handle a vast array of user groups, resource groups, actions groups and relationship groups.

#### It is hierarchical...

Access control policies owned by an organization are also applied to sub-organizations.

#### It is customizable...

Access control policies are externalized from the application code, so changes to policies can be made without recompiling code.

#### It is compact...

The new framework scales well. The number of access control policies grows with the number of business processes and not the number of objects. Most of the grouping framework is based on implicit conditions, so as long as the conditions are satisfied, the policy will apply.

- **Cross-site scripting** — Reject any user request that contain attributes or characters that are designated as not allowed, using the Cross Site Scripting Protection node of the WebSphere Commerce Configuration Manager. It is described in detail in Chapter 4, “Enhancing site security,” on page 35.

### Authentication

- **Password storage** — WebSphere Commerce 5.4 encrypts and stores a one-way hash of passwords using the SHA-1 hashing scheme in the WebSphere Commerce database, rather than storing the passwords themselves. This ensures that user passwords are not decipherable by anyone, including the site or system administrator.
- **Password Invalidation** — Require users to change their passwords when they are logging in to the system for the first time, using the Password Invalidation node of the WebSphere Commerce Configuration Manager. It is described in detail in Chapter 4, “Enhancing site security,” on page 35.
- **Account policy** — Set up an account policy for your site to define the account-related policies in use, by using the Account policy page of the

WebSphere Commerce Administration Console. It is described in detail in Chapter 4, “Enhancing site security,” on page 35.

- **Password policy** — Set up a password policy for your site to control a user’s password selection characteristics using the Password policy page of the WebSphere Commerce Administration Console. It is described in detail in Chapter 4, “Enhancing site security,” on page 35.
- **Account Lockout policy** — Set up an account lockout policy for your site to reduce the chances of a user account being compromised using the Account lockout policy page of the WebSphere Commerce Administration Console. It is described in detail in Chapter 4, “Enhancing site security,” on page 35.

#### **Authorization**

**Password protected commands** — Require users to enter their passwords if they are running requests that run designated commands, using the Password Protected Commands node of the WebSphere Commerce Configuration Manager. It is described in detail in Chapter 4, “Enhancing site security,” on page 35.

#### **Encrypted data**

**Database update tool** — Update encrypted data such as passwords and credit card information as well as the merchant key in a WebSphere Commerce database, using the Database Update Tool node of the WebSphere Commerce Configuration Manager. It is described in detail in Chapter 4, “Enhancing site security,” on page 35.

#### **Session management**

**Login Timeout** — Log off a user that is inactive for an extended period and request they log back on to the system, using the Login Timeout node. This enhancement is invoked through the WebSphere Commerce Configuration Manager and is described in detail in Chapter 4, “Enhancing site security,” on page 35.

#### **Logging**

**Access logging** — Quickly identify any security threats against WebSphere Commerce by enabling access logging. This enhancement is invoked through the WebSphere Commerce Configuration Manager and is described in detail in Chapter 4, “Enhancing site security,” on page 35.

## **Enhancements for the System Administrator**

The following are WebSphere Commerce 5.4 security enhancements that are generally targeted to a site administrator:

- An important security enhancement is the ability to configure the WebSphere Commerce administrative tools to run on a nonstandard port number (for example, port 8000 as opposed to port 443). By restricting access to this port, you can limit access to the administration tools to your local network or intranet.
- From the WebSphere Commerce Administration Console Launch a security program that checks and deletes temporary WebSphere Commerce files that may contain potential security exposures using the Launch security check page.

## **Enhancements for the WebSphere Commerce Programmer**

A key enhancement is that a new access control framework has been implemented in WebSphere Commerce 5.4. This new framework uses access control policies to determine if a given user is permitted to perform a given action on a given resource. The new access control framework provides fine-grained access control. It

works in conjunction with, but does not replace the access control provided by the WebSphere Application Server. The new access control framework is described in detail in Chapter 11, “Access control,” on page 81.

The new access control framework enhances the previous access control in the following ways:

**It is expressive...**

It captures the intent of a large variety of access policies. The framework is generic so that it can handle a vast array of user groups, resource groups, actions groups and relationship groups.

**It is hierarchical...**

Access control policies owned by an organization are also applied to sub-organizations.

**It is customizable...**

Access control policies are externalized from the application code, so changes to policies can be made without recompiling code.

**It is compact...**

The new framework scales well. The number of access control policies grows with the number of business processes and not the number of objects. Most of the grouping framework is based on implicit conditions, so as long as the conditions are satisfied, the policy will apply.

---

## Security improvements in WebSphere Commerce Suite 5.1 Pro Edition

While Commerce Suite 5.1 represented a new e-commerce architecture and was a complete rewrite of the C++-based Commerce Suite 4.1, it contained all the security features of previous WebSphere Commerce Suite versions, plus it added new security improvements. These improvements have been inherited by WebSphere Commerce 5.4.

Commerce Suite 5.1 continued the protection against unauthorized access to WebSphere Commerce Suite administrators and shoppers resources that was provided by earlier releases by:

- Continuing support for access control features that ensure the WebSphere Commerce Suite user is either authenticated or in SSL mode before gaining access to or submitting sensitive information.
- Assigning WebSphere Commerce Suite commands to groups such that only the Site Administrator or Store level Administrators can execute a specific command, followed the same model as Commerce Suite 4.1.

### General Security Enhancements

With the rewrite of Commerce Suite 5.1 in Java™, a number of inherent security problems that plagues software written in C++ were removed. Java does not use pointers, thus it has eliminated the buffer overflow problem that is a security vulnerability of most C++ based software. By complying with the industry standard J2EE specifications, WebSphere Commerce Suite used strong type checking to ensure the server does not execute rogue statements specified by devious individuals.

The industry standard Triple DES (data encryption standard) algorithm was used to protect sensitive information in the WebSphere Commerce Suite system. The package containing the Triple DES algorithm is digitally signed such that if the package were tampered the WebSphere Commerce Suite server would not start.

## Session Management

The WebSphere Commerce Suite session management was completely rewritten for maximum security, using a unique technique to ensure cookies are not stolen. By using an authentication cookie that only flows over SSL (secure sockets layer) and consist of an encrypted timestamp, the rewritten session management design guarded against session hijacking.

## Authentication

System and application passwords needed by the WebSphere Commerce Suite server during execution were securely encrypted, using a merchant specified 12-bit key, and stored in the WebSphere Commerce Suite configuration files. Sensitive information that appears in the users URL entry box is encrypted to protect shoppers from unauthorized disclosure.

## Logging

The WebSphere Commerce Suite log system was designed with security as a key consideration so that sensitive information such as shopper's password and credit card information was not logged by default to the WebSphere Commerce Suite log files.

---

## Conventions used in this book

This book uses the following highlighting conventions:

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

*host\_name*

The fully qualified host name of your WebSphere Commerce Studio machine (for example, `ibm.com` is fully qualified).

 **Windows**

*drive*

The letter representing the drive on which you installed the product or component being discussed (for example, `C:`).



This icon marks a Tip - additional information that can help you complete a task.

---

 **Windows** indicates information that is specific to WebSphere Commerce for Windows NT<sup>®</sup> and Windows<sup>®</sup> 2000.

 **AIX** indicates information that is specific to WebSphere Commerce for AIX<sup>®</sup>.

 **Solaris** indicates information that is specific to WebSphere Commerce for Solaris<sup>™</sup> Operating Environment software.

 **400** indicates information specific to WebSphere Commerce for the IBM @server<sup>™</sup> iSeries<sup>™</sup> 400<sup>®</sup> (formerly called AS/400<sup>®</sup>).

► **Linux** indicates information specific to WebSphere Commerce for Linux.

► **Professional** indicates information specific to WebSphere Commerce Professional Edition.

► **Business** indicates information specific to WebSphere Commerce Business Edition.

---

## Where to find more information

For information on the WebSphere Commerce 5.4 product, refer to the following Web sites:

- ► **Business** [http://ibm.com/software/webservers/commerce/wc\\_be/lit-tech-general.html](http://ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html)
- ► **Professional** [http://www.ibm.com/software/webservers/commerce/wcs\\_pro/lit-tech-general.html](http://www.ibm.com/software/webservers/commerce/wcs_pro/lit-tech-general.html)

For information related to Commerce Studio, Professional Developer Edition 5.1 or earlier releases of WebSphere Commerce Studio, refer to the following Web site:

[http://www.ibm.com/software/webservers/commerce/commercestudio/  
lit-tech-general.html](http://www.ibm.com/software/webservers/commerce/commercestudio/lit-tech-general.html)



---

## **Part 1. WebSphere Commerce security model**

This part provides a conceptual overview of WebSphere Commerce security.





---

# Chapter 1. Introduction to the WebSphere Commerce security model

This chapter describes the WebSphere Commerce security model as well as various WebSphere Commerce security concepts.

---

## Overview

The information in this document describes the notions of authentication, authorization, policies, and confidentiality:

### What is authentication?

Authentication is the process of verifying that users or applications are who they claim to be. In a WebSphere Commerce system, authentication is required for all users and applications accessing the system, with the exception of guest users. The user authentication process is always performed under SSL. This ensures that a third party using network-sniffing programs cannot *snoop* on the network when a user submits a password. Passwords are never decrypted during the authentication process, as is the common security practice. All user passwords are hashed and encrypted using a 128-bit key, known as the *merchant key*. The merchant key is specified during installation and configuration of the WebSphere Commerce system.

The WebSphere Commerce system has its own passwords for administration purposes. These passwords should periodically be changed as part of a WebSphere Commerce site-wide security policy. For details how to change the WebSphere Commerce 5.4 system passwords, see Chapter 7, “Setting and changing passwords,” on page 63.

### What is authorization?

Authorization is the process of determining whether a user can perform a specific operation on a resource. Authorization is determined from the access control policies to WebSphere Commerce resources. In a WebSphere Commerce system, access control is needed in two areas:

- To protect the WebSphere Commerce Enterprise JavaBeans™ (EJB beans) from unauthorized access. This process is discussed in Chapter 5, “Enabling WebSphere Application Server security,” on page 49.
- To ensure that only authorized parties can execute different groups of WebSphere Commerce commands. This process is discussed in Chapter 11, “Access control,” on page 81.

### What are access control policies?

Assuming that you have finished defining the organizations and users that will participate in your e-commerce site, you can now manage their activities through a set of policies, a process referred to as *access control*.

An access control policy is a rule that describes which user or group of users is authorized to perform particular activities on your site. These activities can range from registration, to managing auctions, to updating the product catalog, and

granting approvals on orders, as well as any of the hundreds of other activities that are required to operate and maintain an e-commerce site.

The policies are what grants users access to your site. Unless they are authorized to perform their responsibilities through one or more access control policies, users have no access to any of your site's functions.

The access control model for WebSphere Commerce 5.4 is based upon the enforcement of access control policies. Access control policies are enforced by the access control Policy Manager. In general, when a user attempts to access a protectable resource, the access control policy manager first determines what access control policies are applicable for that user and then, based upon the applicable access control policies, it determines if the user is allowed to perform the requested operation on the given resource.

## What is an audit trail?

In computing, an *audit trail* is used to refer to electronic or paper logs that are used to track computer activity. For example, an employee might have access to a portion of a corporate network such as account receivable, but may not be authorized to access other portions of the system, such as payroll. If that employee attempts to access an unauthorized section by typing in passwords, this improper activity is recorded in the audit trail.

In e-commerce systems, audit trails are used to record customer activity. An audit trail records a customer's initial contact with the system as well as subsequent actions such as payment and delivery of the product or service. Companies can use the audit trail to respond to any inquiries or complaints. It can also use the audit trail to reconcile accounts, to provide analysis and historical information for future planning and budgeting, and to provide a record of sales in case of a tax audit.

Audit trails can also be used to investigate computer crimes over cyberspace and the internet. To expose an individual conducting malicious attacks on a system, investigators can follow the audit trail left by the perpetrator. Sometimes the perpetrators of cyber crimes unknowingly leave behind audit trails in activity logs with their internet service providers or perhaps through chat room logs.

## What is confidentiality?

Confidentiality is the process of protecting sensitive information from being deciphered by unintended recipients. In the WebSphere Commerce system, confidentiality is required when sensitive information flows from the user's browser to the WebSphere Commerce server, and back from the WebSphere Commerce server to the user's browser. As discussed in Chapter 8, "Enabling SSL for production with IBM HTTP Server," on page 69, using Secure Sockets Layer (SSL), provides confidentiality for this scenario.

Confidentiality is also a strong requirement in the area of session management. Because the Hypertext Transfer Protocol (HTTP) protocol is state less, a *cookie* is commonly used to continuously identify the user to the WebSphere Commerce server. If this cookie is stolen, then the user account can be compromised. This is commonly known as *session hijacking*. WebSphere Commerce prevents session hijacking by using unique features of the cookie specifications as discussed in Chapter 6, "Session management," on page 57.

---

## Chapter 2. Authentication

WebSphere Commerce views authentication as the process of verifying that users or applications are who they claim to be. This section describes the details of several aspects of WebSphere Commerce authentication.

---

### WebSphere Commerce authentication model

The WebSphere Commerce authentication model is based on the following concepts:

- Challenge mechanisms
- Authentication mechanisms
- User registry

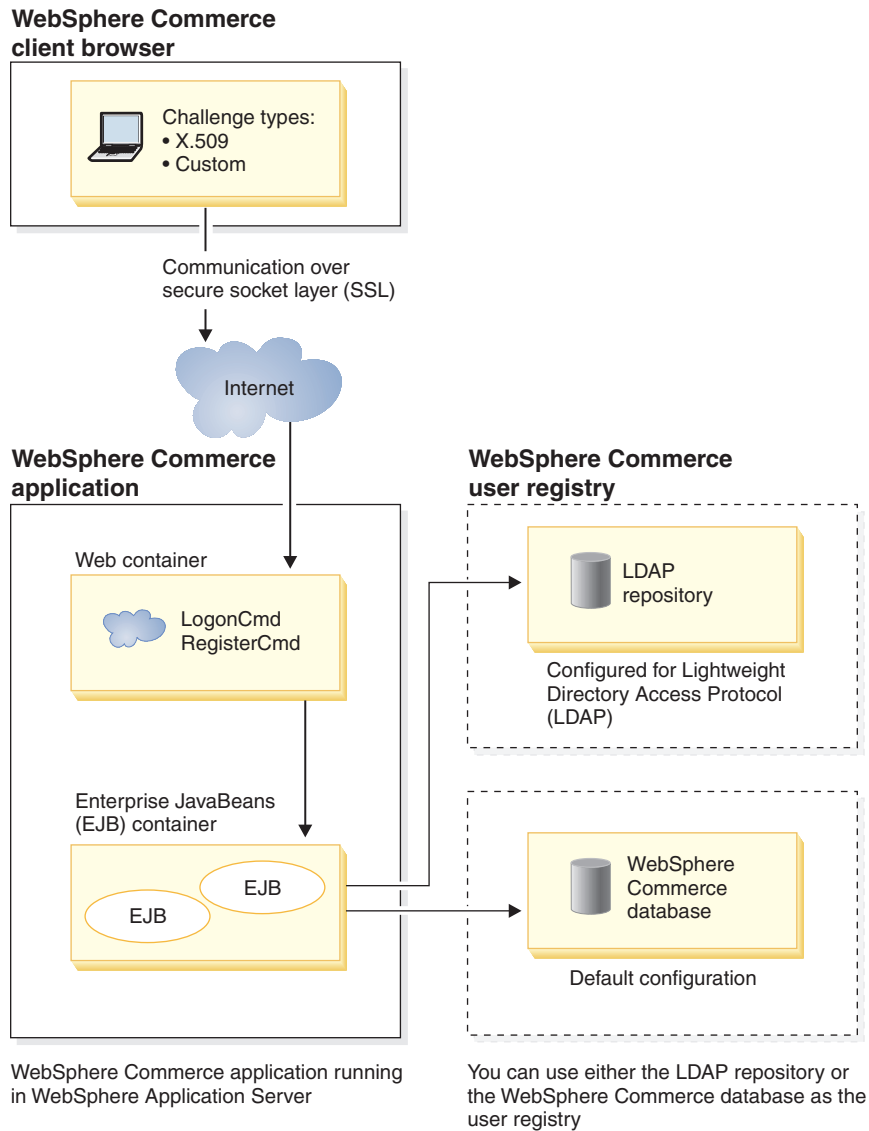


Figure 1. WebSphere Commerce 5.4 security model

## Challenge mechanisms

A challenge mechanism specifies how a server challenges and retrieves authentication data from a user. WebSphere Commerce 5.4 supports the following authentication methods or challenge mechanisms:

### Form-based or custom authentication

This authentication mechanism permits a site or store specific login through an HTML page or a JSP form.

### Certificate-based authentication (X.509 certificate)

The certificate challenge mechanism implies that the Web server is configured to perform mutual authentication over SSL. The client is required to present a certificate in order to establish the connection. This certificate is then credential mapped to a user registry.

## Authentication mechanisms

An *authentication mechanism* authenticates a user by verifying the user's authentication data against an associated user registry. WebSphere Commerce 5.4 issues an authentication token that is associated with a user on every subsequent request after the authentication process. It is terminated when the user logs off or closes the browser.

### Certificate validation

This is the process of verifying that the X.509 client certificate is trusted by the Web server and that it complies with the Web server's certificate policy. WebSphere Commerce also verifies the X.509 certificate against the WebSphere Commerce database. The Web server performs the coarse-grain access control on the certificate, while WebSphere Commerce performs a fine-grain access control on the certificate.

### LDAP bind

This is process of verifying the challenge information supplied is valid, by performing an LDAP bind operation to authenticate the user.

### Database bind

This is the process of verifying the user ID and password supplied during the authentication process is valid when compared to the authentication information stored in the WebSphere Commerce database.

## User registry

The user registry is a repository that contains user information, and the user's authentication information (for example, the password). Authentication information provided by a principal (that is, the representation of a human user or system entity in a user registry) can be verified or validated against the user registry.

WebSphere Commerce 5.4 supports user registries based on two user domains: LDAP user registry and the WebSphere Commerce database.

WebSphere Commerce 5.4 supports the following LDAP providers:

-  IBM SecureWay® Directory
-  Netscape® Directory Server
-  Windows 2000 Active Directory

---

## Credentials

The WebSphere Commerce 5.4 server supports authentication mechanisms based on validating credentials, such as certificates, tokens, or user ID and password pairs. Credentials are verified against a user registry that supports such a scheme.

## WebSphere Commerce token

WebSphere Commerce uses a secure authentication cookie to manage authentication data. An authentication cookie flows only over SSL, and is time-stamped for maximum security. This cookie is used to authenticate the user under SSL-connections whenever a sensitive command is executed, for example, the DoPaymentCmd, which asks for a users credit card number. There is minimal risk that this cookie could be stolen and used by an unauthorized user.

A second cookie that flows between the browser and server under either SSL or non-SSL connection is used for verification of the user under non-SSL connections.

## WebSphere Application Server LTPA token

An LTPA token is a piece of data that contains user information necessary to determine access permissions for a resource that is requested by the user. It contains the authentication data along with the digital signature of the WebSphere Application Server LTPA server.

In the case of the WebSphere Application Server Lightweight Third Party Authentication scheme, an LDAP directory containing the information about the users is the user registry against which authentication is performed. The resource server contacts the WebSphere Application Server Security Server and specifies LTPA to be the authentication mechanism. It also supplies the authentication data associated with the request. The WebSphere Application Server Security Server then validates the authentication data against the LTPA server and returns an LTPA token.

---

## Single sign-on

The philosophy behind the HTTP single sign-on is to preserve user authentication across multiple HTTP requests. Its goal is to: avoid prompting the user multiple times for security credentials within a given trust domain that includes:

- Cooperating but disparate WebSphere Application Server servers.
- Cooperating applications such as LDAP servers such as IBM SecureWay Directory Server.

In a single sign-on (SSO) scenario, an HTTP Cookie is used to propagate a user's authentication information to disparate Web servers relieving the user from entering authentication information for every new client-server session (assuming basic authentication).

For the steps to implement single sign-on with WebSphere Commerce, see Chapter 10, "Single sign-on," on page 77.

---

## Authentication policies

An authentication policy is a set of rules that are applied to the authentication process and to the verification of authentication data by WebSphere Commerce. WebSphere Commerce 5.4 supports account policies, other authentication-related policies, and session policies as described in the following sections.

### Account policies

The following sections describe account policies available with WebSphere Commerce:

#### Account policy

The Account policy page of the WebSphere Commerce Administration Console allows you to set up an account policy. An account policy defines the account-related policies such as password and account lockout policies.

Once you have created an account policy, you can assign the policy to a user. Note that you cannot delete an account policy if it is in use (that is, a user is assigned the account policy).

For information on creating account policies, see "Setting up an account policy" on page 44.

Also see the reference topic "Default Authentication Policies" in the WebSphere Commerce online help.

### **Account lockout policy**

The Account lockout policy page of the WebSphere Commerce Administration Console allows you to set up an account lockout policy for different user roles within WebSphere Commerce. The account lockout policy disables a user account if malicious actions are launched against that account in order to reduce the chances that the actions compromise the account.

The account lockout policy enforces the following items:

- The account lockout threshold. This is the number of invalid logon attempts before the account is disabled.
- Consecutive unsuccessful login delay. This is the time period for which the user is not allowed to login, after two failed attempts to login. The delay gets incremented by the configured time delay value (for example, 10 seconds) with every consecutive login failure.

For information on creating account lockout policies, see "Setting up an account lockout policy" on page 46.

### **Password policy**

The Password policy page of the WebSphere Commerce Administration Console allows you to control a user's password selection in order to define the characteristics of the password to ensure that it complies with the security policy for your site.

This feature defines attributes with which the password must comply. The password policy enforces the following conditions:

- Whether the user ID and password can match.
- Maximum occurrence of consecutive characters.
- Maximum instances of any character.
- Maximum lifetime of the passwords.
- Minimum number of alphabetic characters.
- Minimum number of numeric characters.
- Minimum length of password.
- Whether the user's previous password can be reused.

For information on creating password policies, see "Setting up a password policy" on page 45.

Also see the reference topic "Default Authentication Policies" in the WebSphere Commerce online help.

## **Other authentication-related policies**

The following sections describe the other authentication-related policies available with WebSphere Commerce:

### **Password Invalidation**

Use the Password Invalidation node of the Configuration Manager to enable or disable the password invalidation feature. This feature, when enabled, requires WebSphere Commerce users to change their password if the user's password has

expired. In that case, the user is redirected to a page where they are required to change their password. Users are not able to access any secure pages on the site until they have changed their password.

For information on using the Password Invalidation node, see “Activating password invalidation” on page 39.

### **Password Protected Commands**

Use the Password Protected Commands node of the Configuration Manager to enable or disable the password protected commands feature. When this feature is enabled, WebSphere Commerce requires registered who are logged onto WebSphere Commerce to enter their password before continuing a request that runs designated WebSphere Commerce commands.

**Caution:** When you configure the password protected commands, some of the commands shown in the command selection list can be executed by generic or guest users. Configuring such commands as password protected will restrict generic and guest users from running them. Therefore, you should exercise caution when you configure commands to be password protected.

**Note:** WebSphere Commerce will only display the commands that are designated as authenticated or set with the `https` flag in the `URLREG` table in the list of available commands.

For information on using the Password Protected Commands node, see “Enabling password protected commands” on page 39.

## **Session policies**

In WebSphere Commerce 5.4, session policies are embodied in the login timeout policy.

With the login timeout policy, WebSphere Commerce will log off a user that is inactive for an extended period and request they log back on to the system using the Login Timeout node. This enhancement is invoked through the WebSphere Commerce Configuration Manager and is described in detail in “Enabling login timeout” on page 38.



---

## Chapter 3. Authorization (Access Control)

WebSphere Commerce views authorization as the process of verifying that users or applications have sufficient authority to access a resource. This section describes the details of several aspects of WebSphere Commerce access control.

Authorization or access control, in WebSphere Commerce is accomplished using access control policies. An access control policy is a rule that describes which group of users can perform a set of actions on a set of resources. WebSphere Commerce provides a set of default access control policies. These default access control policies are specified in XML format and are designed to address many of the typical access control requirements that an e-commerce site needs. In order to understand the access control component of WebSphere Commerce, you must first understand the typical organizational hierarchy of an e-commerce site.

---

### Organizational hierarchy

Users and organizational entities within the WebSphere Commerce member subsystem are organized into a hierarchy. This hierarchy emulates a typical organizational hierarchy, with entries for organizations and organizational units, and entries for users in the leaf nodes. The hierarchy includes an artificial organizational entity called a *root organization* at the top. All other organizational entities and users are descendants of this root organization. Under the root organization there can be one seller organization and several buyer organizations; all these organizations can have one or more sub-organizations under them. Buyer or Seller Administrators are the heads of the organizations, and they are responsible for maintaining their organizations. On the seller organization side, each sub-organization can have one or more stores within it. Store Administrators are responsible for maintaining the stores. The following diagram shows the organizational hierarchy of a business-to-business e-commerce site.

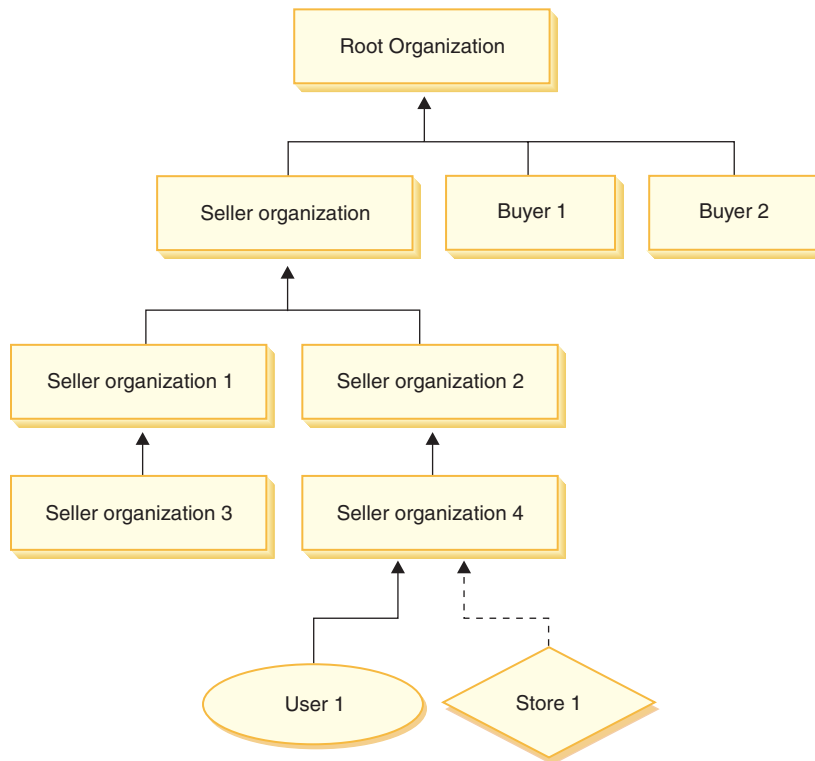


Figure 2. Organizational hierarchy of a business-to-business site

## Root organization

The root organization is at the top of the organizational hierarchy. A Site Administrator has super-user access to perform any operation within WebSphere Commerce. The Site Administrator installs, configures, and maintains WebSphere Commerce and its associated software and hardware. This role typically controls access and authorization (that is, creating and assigning members to the appropriate role) and manages the Web site. The Site Administrator can assign roles to users and specify the organization(s) for which the user plays the role. The Site Administrator must assign a password to each administrator to ensure that only authorized parties can access confidential information. This provides a way to control key responsibilities, such as updating a catalog or approving a request for quotation (RFQ).

**Note:** It is possible for a user to play roles in an organization other than their parent organization.

In a WebSphere Commerce site, there is one seller organization. In a business-to-business site, there are also one or more buyer organizations. The Site Administrator may define both the access control policies of the seller organization (that owns the store) as well as the access control policies of each organization that buys from the store. In a business-to-consumer site, there are no buyer organizations. Business-to-consumer customers are modeled as members of the default organization.

## Organizations (seller)

Both in business-to-business and business-to-consumer sites, the Site Administrator creates one top-level seller. Underneath this seller organization, other sub-organizations or organization units can be created. Any of these sell-side organizational entities can own one or more stores. The Site Administrator then defines any special access control policies for a seller organization, and assigns a Seller Administrator to manage that organization. The Seller Administrator registers users, assigns them different roles to fit the organization's business needs according to the access control policies pertaining to that organization.

The Seller Administrator's responsibilities are summarized as follows:

- Create sub-organizations that can own stores. Optionally, define which processes within the organization require approval. This step is only required in a business-to-business site.
- Assign roles to the sub-organizations.
- Create users.
- Assign roles to users.

## Organizations (buyer)

In a business-to-business site, the Site Administrator creates one or more buyer organizations, depending on the business needs. The Site Administrator then defines any special access control policies for a buyer organization and assigns a Buyer Administrator to manage the buyer organization. The Buyer Administrator registers users and assigns them different roles to fit the organization's business needs, according to the access control policies pertaining to that organization.

The Buyer Administrator's responsibilities are summarized as follows:

- Create and administer the sub-organizations within the buyer organization. Optionally, define which processes within the organization require approval. This step is only required in a business-to-business site.
- Assign roles to the sub-organizations.
- Create users.
- Assign roles to users.

**Note:** The Site Administrator can modify and manage the access control policies of the buyer organization if appropriate. For more information on the Site Administrator's tasks, see "Site Administrator" on page 14.

---

## Roles

As mentioned above, WebSphere Commerce provides default sets of roles. The Site Administrator must assign specific roles to every organization before assigning users to those roles. An organization can only take on roles that have been assigned to its parent organization. Similarly, a user can only take on roles that have been assigned to their parent organization .

All roles in WebSphere Commerce are scoped to an organization. For example, a user plays the Product Manager role for Organization X. The parent organization of this user must also be assigned the Product Manger role for itself. The access control policies could then be setup such that this user can only perform product management operations within the context of Organization X and its sub-organizations.

**Note:** Assigning roles to users and organizations is done in the MBRROLE table.

The default roles that come with WebSphere Commerce can be grouped into the following categories:

- Site operations
- Site and content development
- Marketing management
- Product management
- Sales management
- Logistics and operations management
- Organizational management

## Site operations

The following technical operations roles are supported by WebSphere Commerce:

- Site Administrator
- Store Administrator

### Site Administrator

The Site Administrator installs, configures, and maintains WebSphere Commerce and the associated software and hardware. The Administrator responds to system warnings, alerts, and errors, and diagnoses and resolves system problems. This role typically controls access and authorization (creating and assigning members to the appropriate role), manages the Web site, monitors performance, and manages load balancing tasks. The Site Administrator may also be responsible for establishing and maintaining several server configurations for different stages of development such as testing, staging, and production. This role also handles critical system backups and resolves performance problems.

### Store Administrator

The Store Administrator manages the store assets, and updates and publishes changes to taxes, shipping, and store information. The Store Administrator can also manage the access control policies for the organization. The Store Administrator, usually the lead on the store development team, is the only role on the team with the authority to publish a store archive (the Site Administrator can also publish a store archive). The Store Administrator is usually Web-literate and has a thorough knowledge of the store's business procedures.

## Site and content development

WebSphere Commerce supports the Store Developer site and content development role.

### Store Developer

Store Developers create Java Server Pages files and any necessary customized code and can modify any of the standard functionality included with WebSphere Commerce. Once a store archive has been created, Store Developers have the authority to make changes to it manually or by using the Store Profile notebook and Tax and Shipping notebooks. They do not have the authority to publish the store archive to the WebSphere Commerce Server.

## Logistics and operations

WebSphere Commerce supports the following logistics and operations management roles:

- Logistics Manager
- Operations Manager
- Receiver
- Returns Administrator
- Pick Packer

### **Logistics Manager**

**Business** The Logistics Manager, sometimes called the Shipping Manager, manages and negotiates bulk freight or shipping from carriers to warehouse, and to individual customers. This role is responsible for ensuring the company uses the best shippers at the best costs to meet company strategy. Shipping is an important aspect of customer service and may be a key success factor for the online business.

### **Operations Manager**

**B2C** This role manages order processing, ensuring that orders are properly fulfilled, payment is received, and orders are shipped. The Operations Manager can search for customer orders, view details, manage order information, and create and edit returns.

### **Pick Packer**

The Pick Packer picks products from fulfillment centers and packs the products for shipping to customers. The Pick Packer also manages pick tickets and packing slips, which are used to confirm shipment of products during order fulfillment.

### **Receiver**

The Receiver receives inventory at the fulfillment center, tracks expected inventory records and ad hoc receipts for ordered products, and receives returned products as a result of customer returns.

### **Returns Administrator**

The Returns Administrator manages the disposition of returned products.

- List returns
- Lists returned products
- Dispositions returned products

## **Product management**

The following product management roles are supported by WebSphere Commerce:

- Buyer (seller side)
- Category Manager
- Product Manager or Merchandising Manager

### **Buyer (seller-side)**

The buyer purchases merchandise for sale. The buyer handles relations with vendors or suppliers and negotiates to obtain the desired product with favorable terms for such things as delivery and payment options. The buyer may set prices. Inventory is managed by the buyer in order to determine the quantities to buy and ensure that stock is properly replenished.

### **Category Manager**

The category manager manages the category hierarchy by creating, modifying, and deleting categories. The category hierarchy organizes products or services offered by the store. The Category Manager also manages products, expected inventory records, vendor information, inventory, and return reasons.

## Product Manager/Merchandising Manager

The **Business** Merchandising or **B2C** Product Manager traces customer purchases, suggests discounts, and determines the best way to display, price, and sell products in the online store

- Performs all Category manager tasks
- Performs all Marketing manager tasks

## Sales management

The following business relationship management roles are supported by WebSphere Commerce:

- Sales Manager
- Account Representative
- Customer Service Supervisor
- Customer Service Representative

### Sales Manager

Sales Managers acquire and retain customers, meet sales forecasts, provide incentives for increased customer business, contract management, set pricing terms, work with product manager to establish inventory forecasts, and work with the Marketing Manager for promotions

### Account Representative

Account representatives work with individual accounts to build relationships, and manage customer service issues. They may be authorized to change contract pricing, negotiate contracts, profiles, and analyze profitability by account category.

### Customer Service Supervisor

This role has access to all customer service tasks. The Customer Service Supervisor manages customer inquiries (such as customer registration, orders, returns, and auctions) and has authority to complete tasks that cannot be accessed by a Customer Service Representative, such as approving system-denied returns records, and contacting customers regarding payment exceptions (such as credit card authorization failures).

### Customer Service Representative

No matter how well an online business is designed to provide a customer with self-service features, there will be some types of customers or some occasions when even the most web-literate customer will require personal contact. Most online businesses provide an e-mail, fax or contact number for the customer to obtain direct service. It is the responsibility of the customer service representative to handle all inquiries from the customer.

## Marketing management

WebSphere Commerce supports the marketing management role of Marketing Manager.

### Marketing Manager

The Marketing Manager communicates the market strategy and brand messages to the customers. This role monitors, analyzes, and understands customer behavior. In addition, the marketing manager creates or modifies customer profiles for targeted selling, and creates and manages campaigns and promotions. Campaign event planning can be handled by a team comprising the Merchant, Marketing Manager, and Merchandising Manager.

## Organizational management

WebSphere Commerce supports the following organizational management roles:

- Seller Administrator
- Buyer Administrator
- Buyer Approver

### **Seller Administrator**

The Seller Administrator manages the information for the selling organization. Seller administrators create and administer the sub-organizations within the selling organization and the various users in the selling organization, including the assignment of the appropriate business roles.

### **Buyer Administrator**

The Buyer Administrator manages the information for the buying organization. They create and administer the sub-organizations within the buying organization and manage the various users including approving users as buyers. Other buy-side roles such as buyer approvers and additional buyer organization administrators may be created and managed.

### **Buyer Approver**

A Buyer Approver is an individual in the buying organization who approves orders made by buyers before the order is submitted for purchase with the seller.

---

## Access control policy

An access control policy authorizes a group of users to perform a set of actions on a set of resources within WebSphere Commerce. Unless authorized through one or more access control policies, users have no access to any functions of the system. To understand access control policies you need to understand four main concepts: users, actions, resources, and relationships. Users are the people who use the system. Resources are objects in the system that need to be protected. Actions are the activities that users can perform on the resources. Relationships are optional conditions that exist between users and resources.

## Elements of an access control policy

An access control policy consists of four elements:

### **Access group**

The group of users to which the policy applies.

### **Action Group**

A group of actions performed by the user on resources.

### **Resource group**

The resources controlled by the policy. A resource group may include business objects like contract or order, or a set of related commands such as, all the commands that users of a particular role can perform.

### **Relationship (optional)**

Each resource class can have a set of relationships associated with it. Each resource can have a set of users that fulfill each relationship. For example, a policy could specify that only the creator of an order can modify it. In this case, the relationship would be creator , and it is between the user and the order resource.

## Access control policy concepts

Access control policies grant users access to your site. Unless they are authorized to perform their responsibilities through one or more access control policies, users have no access to any of your site's functions.

Each access control policy takes the following form:

```
AccessControlPolicy [AccessGroup,ActionGroup,ResourceGroup,Relationship]
```

The elements in the access control policy specify that a user belonging to a specific access group is permitted to perform actions in the specified action group on resources belonging to the specified resource group, as long as the user satisfies a particular relationship with respect to the resource. The relationship is only specified when needed. For example, [AllUsers,UpdateDoc,doc,creator] specifies that all users can update a document, if they are the creator of the document.

The following sections describe conceptual information and terminology associated with access control.

### Member groups

The Member subsystem in WebSphere Commerce allows you to create member groups, which are groups of users categorized for various business reasons. The groupings can be used for many purposes, for example, access control purposes, approval purposes, as well as for marketing purposes such as calculating discounts and prices, and displaying products. A member group of type Access Group (-2) is for access control purposes, while a member group of type User Group (-1) is for general use. A member group is associated with member group types in the MBRGRPUSG table.

**Access groups:** A member group of type Access Group (-2) is for grouping users for access control purposes. An access group is one element of an access control policy, and is defined as a group of users defined specifically for access control purposes. The criteria for membership in an access group is usually based on roles, the organization to which the user belongs, or the user's registration status. For example, the access group called Buyer Administrators is a group whose users play the role of Buyer Administrator.

WebSphere Commerce includes a number of default roles, and corresponding to each role is a default access group that implicitly references that role. Roles can be used as attributes to add users to an access group based on the type of activities they perform in the site. For example, by default there is a role called Seller Administrator and a corresponding access group called Seller Administrators. A Site Administrator uses the WebSphere Commerce Administration Console to create, maintain, and delete access groups for a site. A Buyer Administrator or a Seller Administrator uses the WebSphere Commerce Organization Administration Console to assign roles to users or to explicitly assign users to access groups. Access groups can be implicit, explicit or both.

*Implicit access group:* An implicit access group is defined by a set of criteria. Anyone who satisfies the criteria is a member of the group. The criteria are usually based on a user's roles, parent organization, or registration status. The implicit conditions that define membership in a member group are in the CONDITIONS column of the MBRGRP table. Using implicit access groups that specify the attributes of users, makes it easy to authorize access to similar users without having to explicitly assign and unassign individual users. It also eliminates the need to update the members of a group when a user's attributes change. A simple criterion for an access group is to include everyone that has been assigned a specific role,



regardless for which organization the user plays the role. A more complex criterion would be to specify that only users that play one of a possible set of roles for a particular organization would belong to the access group.

*Explicit access group:* It is possible to explicitly add or remove a user from a member group. Both of these explicit specifications can be done using the MBRGRPMBR table. An explicit access group contains explicitly assigned users who may or may not share common attributes. This also allows you to exclude individuals that satisfy the conditions for inclusion in an implicitly defined group, but that you want excluded anyway.

**User groups:** A member group of type User Group (-1) is a collection of users defined by the merchant, who share a common interest. User groups are similar to clubs that are offered by large stores for their frequent or preferred customers. Being part of a user group can entitle customers to discounts or other bonuses for purchasing products. For example, if market research shows that senior customers repeatedly purchase travel books and luggage, you can assign these customers to a member group called Seniors' Travel Club. Likewise, you can create a user group to reward frequent customers for their business.

## Actions

Generally, an action is an operation that is performed on a resource. In role-based policies for controller commands, the action is Execute and the resource is the command being executed. In role-based policies for Views, the action is the name of the view, and the resource is `com.ibm.commerce.commands.ViewCommand`. For resource-level access control, actions typically map to WebSphere Commerce commands, and the resource is usually the remote interface of a protected EJB (Enterprise Java Bean). For example, the controller command `com.ibm.commerce.order.commands.OrderCancelCmd` operates on the `com.ibm.commerce.order.objects.Order` resource. Lastly, the Display action is used to activate databean resources.

The WebSphere Commerce Administration Console can be used by a Site Administrator to associate existing actions with action groups, but not for creating new actions. New actions can be created by defining them in an XML file and then loading them to the database. Actions are stored in the ACACTION table.

## Action groups

Action groups are groups of related actions. An example of an action group is the AccountManage group that includes the following commands:

- `com.ibm.commerce.account.commands.AccountDeleteCmd`
- `com.ibm.commerce.account.commands.AccountSaveCmd`

Only the Site Administrator can create, update, and delete action groups. This can be done from the WebSphere Commerce Administration Console and through XML. Action groups are stored in the AACTGRP table. Actions are associated with action groups in the AACTACTGP table.

## Resource category

Resource category refers to a class of resources that need to be protected by access control. Resources must implement the Protectable interface information. Resource categories are Java classes such as order, RFQ, and auction. Resources are the instances of these classes. For example, Auction1 created by Auction Administrator A is one resource; Auction2 created by Auction Administrator B is another resource. These two resources belong to the resource category: auction.

**Note:** For more information on the Protectable interface, see the *IBM WebSphere Commerce Programmer's Guide*.

Resource categories are defined in the ACRESGCRY table, and for convenience, are sometimes referred to as resources. A Site Administrator can associate existing resource categories with resource groups, using the WebSphere Commerce Administration Console. New resource categories can be created using XML.

## Resources

Resources are any objects in the system that need to be protected. For example, RFQs, auctions, users, and orders are some of the resources in WebSphere Commerce which need to be protected. Each resource has an owner. The ownership of the resource is used to determine which access control policies apply to it. Access control policies have an owner, which is an organizational entity. A policy is only applied to resources that are owned by the same organizational entity that owns the policy. Policies that are owned by ancestor organizational entities are also applied to the resource.

**Controller command resources:** For role-based access control for controller commands, the policy is structured such that the Execute action is being performed on the controller command resource. These policies are intended to restrict the execution of controller commands to users with a specified role. The access group for these policies is usually those with a single role, for example, Product Managers (those with the Product Manager role). Then, the resource group would be the set of controller commands that a product manager can execute.

While enforcing role-based access control on a controller command, the owner of the command must be determined. This is done by calling the `getOwner()` method on the command if it has been implemented. Usually this method is not implemented, so WebSphere Commerce Runtime will evaluate it by doing one of the following:

- Use the organization that owns the store that is currently in the command context.
- If there is no store in the command context, use the Root Organization as the owner.

**Data bean resources:** Not all data beans require protection. Within the existing WebSphere Commerce application, data beans that require protection already implement the required access control. The question of what to protect comes into play when you create new data beans. Deciding which resources to protect depends upon your application. A data bean should be protected (directly or indirectly), if the information to be displayed is not sufficiently protected by the role-based access control on the view, that corresponds to the JSP (Java Server Page) that contains the data bean.

If a data bean needs to be protected and can exist on its own, it should be directly protected. If the existence of a data bean depends upon the existence of another data bean, then it should delegate to the other data bean for protection. An example of a data bean that would be directly protected is the Order data bean. An example of a data bean that would be indirectly protected is the OrderItem data bean, as it cannot exist without Order data bean. Refer to the *WebSphere Commerce 5.4 Programmer's Guide* for more information on how to protect the data bean resource.

**Data resources:** Data resources refer to business objects that can be manipulated such as, auctions, orders, RFQs, and users. These are usually protected at the

enterprise bean level, but it is possible to protect any class, as long as it implements the Protectable interface. Data resources are protected using resource-level access control checks. The common way of doing this is by returning data resources in the `getResources()` method of a controller or task command. For more information see the *WebSphere Commerce 5.4 Programmer's Guide*.

## Resource groups

A resource group identifies a set of related resources. A resource group can include business objects such as a contract or a set of related commands. In access control, resource groups specify the resources to which the access control policy authorizes access.

Resource groups are defined in the ACRESGRP table. Site Administrators can manage resource groups and associate resources with resource groups using the WebSphere Commerce Administration Console, or by using XML.

**Implicit resource groups:** Implicit resource groups define resources that match a certain set of attributes. One of these attributes must be the Java class name. Other attributes may include status, store ID, price, etc. For example, you could create an implicit resource group that includes all orders that have pending status (`ORDERS.STATUS=P`). Implicit resource groups are usually used for grouping resources that will be used in resource-level policies, when the resources share a common attribute beyond the Java class name.

Implicit resource groups are defined using the CONDITIONS column of the ACRESGRP table. Simple implicit resource groups can be created using the WebSphere Commerce Administration Console. Increasingly complex groups can be created using XML.

**Explicit resource groups:** Explicit resource groups are specified by associating one or more resource categories to a resource group. This association is done in the ACRESGPRES table. Adding a resource category to a group explicitly, by listing its Java class name, lets you group individual resources that might not necessarily share common attributes.

## Relationships

Each resource may have some kind of relationship associated with it, and a set of members that fulfill each relationship. For example, all resources have a relationship of *owner*, which is fulfilled by the owner of the resource. Other relationships can include recipients of documents and the creator of an order. These resource relationships are important in determining who can perform certain actions on a particular instance of a resource. For example, the creator of a document may not be able to delete it, but perhaps an auditor may. Similarly, a reviewer may only be able to read and approve a document, but not forward it or perform other operations.

Relationships are stored in the ACRELATION table, and are optionally specified in an access control policy, using the ACRELATION\_ID column of the ACPOLICY table. When evaluating a policy that requires the fulfillment of a relationship between the user and the resource, the `fulfills(Long Member, String relationship)` method on the resource will be called to evaluate it. When comparing these relationships to relationship groups, these relationships are sometimes referred to as simple relationships.

**Relationship groups:** Access control policies can specify that a user must fulfill a particular relationship with respect to the resource being accessed, or they can

specify that a user must fulfill the conditions specified in a relationship group. In most cases, a relationship is sufficient. However, if more complex relationships are needed, a relationship group can be used instead. A relationship group allows you to specify multiple relationships and also a chain of relationships. Both of these are done using a relationship chain construct. A relationship chain is a construct that can express a simple relationship (directly between a user and the resource), but can also be used to express a series of relationships between the user and the resource. For example, in order to express that a user must have a role in an organization that has a relationship (other than the owner relationship) with the resource, one must use a relationship group. In this example, there is a role relationship between the user and the organization, and a relationship between the organization and the resource.

*Comparing relationships and relationship groups:* In most cases, using a relationship should satisfy the access control requirements for your application since, conceptually, most relationships are directly between a user and the resource. For example, the policy states that the user must be the creator of the resource. If however, you need to specify multiple relationships, a relationship group should be used. For example, the policy states that the user must be the creator or the submitter of the resource.

Relationship groups are also needed to express a chain of relationships between a user and the resource. In a chain of relationships, there is no direct relationship between the user and the resource for example, a user belongs to the buying organization specified by an order. In this case, the user has a child relationship with the organization, and that organization has a buying relationship with the order.

*Relationship chains:* Each relationship group consists of one or more RELATIONSHIP\_CHAIN open conditions, grouped by andListCondition or orListCondition elements. A relationship chain is a series of one or more relationships. The length of a relationship chain is determined by the number of relationships it consists of. This can be determined by examining the number of <parameter name= "X" value="Y"/> entries in the XML representation of the relationship chain. The following is an example of a relationship chain with a length of one.

```
<openCondition name="RELATIONSHIP_CHAIN">
<parameter name="RELATIONSHIP"
value="aValue"/>
</openCondition>
```

For relationship chains of length one, the <parameter name="Relationship" value="something"> element specifies a direct relationship between the user and the resource. The value attribute is the string representing the relationship between the user and the resource. It must also correspond to the relationship parameter of the fulfills() method on the protectable resource.

When a relationship chain has a length of two, it is a series of two relationships. The first <parameter name= "X" value="Y"/>, element is between a user and an organizational entity. The last, <parameter name= "X" value="Y"/>, element is between that organizational entity and the resource. The following is an example of a relationship chain with a length of two.

```
<openCondition name="RELATIONSHIP_CHAIN">
<parameter name="aValue1" value="aValue2"/>
<parameter name="RELATIONSHIP" value="aValue3"/>
</openCondition>
```

The `aValue1` possible values include `HIERARCHY` and `ROLE`. `HIERARCHY` specifies that there is a hierarchical relationship between the user and the organizational entity in the membership hierarchy. `ROLE` specifies that the user plays a role in the organizational entity.

If the value of `aValue1` is `HIERARCHY`, the possible values include `child`, which returns the organizational entity for which the user is a direct child in the member hierarchy. If the value of `aValue1` is `ROLE`, possible values include any valid entries in the `NAME` column of the `ROLE` table which return all of the organizational entities for which the current user plays this role.

The `aValue3` entry, is a string representing the relationship between one or more organizational entities retrieved from evaluating the first parameter and the resource. This value corresponds to the relationship parameter of the `fulfills()` method on the protectable resource. If more than one organizational entity was returned by evaluating parameter `aValue1`, this part of the `RELATIONSHIP_CHAIN` is satisfied if at least one of these organizational entities satisfies the relationship specified by parameter `aValue2`.

**Note:** A relationship group that consists of a single relationship chain with a single parameter element, is functionally equivalent to a simple relationship. In this case, it is easier to use relationship instead of relationship group in the policy.

## Resource and policy ownership

All policies are owned by an organizational entity. All access control resources also have an owner that is usually an organizational entity; for example, an order is owned by the organization that owns the store where the order was placed. Users can also own resources, for example a registered user owns his own user registration information. Ownership of resources and access control policies is important when determining which policies to apply to a certain resource. For a given resource, the policies that belong to its owning organizational entity and that owner's ancestor organizational entities are applied.

## Types of access control policies

There are two types of access control policies:

- Standard policies
- Template policies

### Standard policies

Standard policies have a fixed owner. For example, if a standard policy is owned by Seller Organization, it will only apply to resources that are owned by Seller Organization and to resources that are owned by its descendant organizational entities, if they exist. Since the Root Organization is the ancestor organization of all other organizations in WebSphere Commerce, any policy that is owned by Root Organization (member ID = -2001), by definition applies to all resources in the site. Thus, standard policies that are owned by the Root Organization are sometimes referred to as Site-level policies.

Standard policies that are not owned by Root Organization are referred to as organizational level policies, since they do not apply site-wide; only to the resources that are owned by the policy owner or by any of its descendant organizational entities. A store administrator can manage the policies for his own organizational entity and its descendant organizational entities. Site administrators can modify all policies.

## Template policies

Template policies have a dynamic owner. Template policies apply dynamically to the organizational entity that owns the resource and its ancestor organizational entities. For example, consider that there are 10 organizations under Root Organization, and each one wants to ensure that Store Administrators can modify only resources that are owned by the Organization for which they play their role. There are two ways to set this up:

1. Have one template policy that will dynamically apply to any of the 10 organizations, depending on the resource that is being accessed. The criteria for the access group in the template policy can also be dynamic. For example, if a user is trying to access a resource owned by Organization 3, the owner of the template policy will dynamically change to Organization 3, and the access group will also dynamically scope itself to Organization 3, that is, the user must play the role of Store Administrator for Organization 3.
2. Have 10 policies, each one owned by one of the 10 organizations. The access group for Organization 1 would specify that the user must play the Store Administrator role for Organization 1. The access group for Organization 2 would specify that the user must play the Store Administrator role for Organization 2, and so on.

The advantage of the first solution is that there is only one physical copy of the policy, but 10 logical copies. Template policies can be managed by a site administrator.

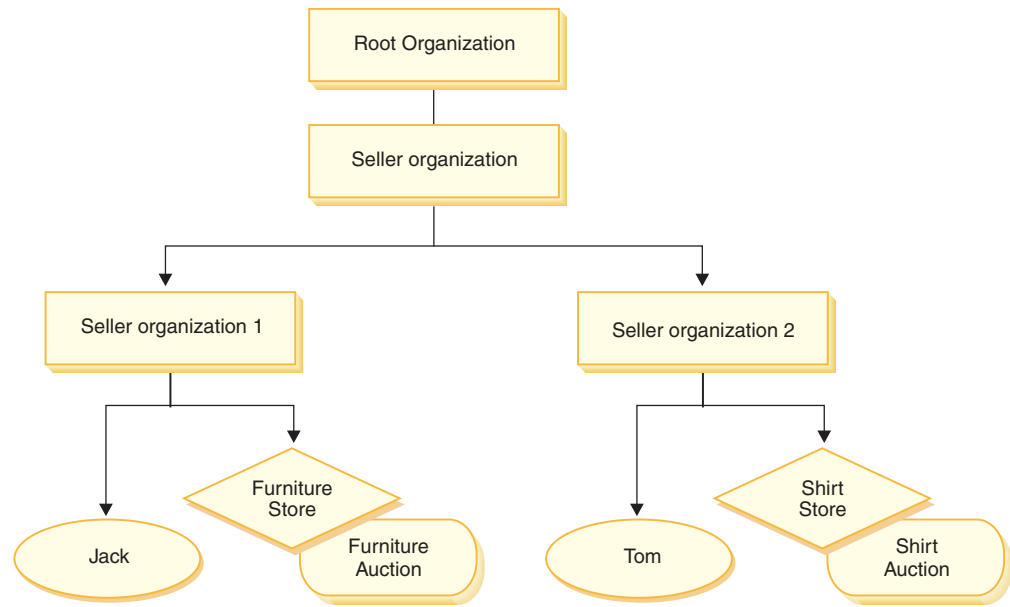
**Overriding Template Policies:** Another feature of template policies is that they can be overridden for specified organizational entities. Going back to the example above, if an 11th organization entity is added to the WebSphere Commerce site, but this newest organizational entity does not want the above template policy to apply to it, there is a way of specifying this. An entry must be added to the ACORGPOL table, specifying the policy id of the template policy, and the organizational entity ID of the 11th organization. This can also be done through the WebSphere Commerce Administration Console, when a Store Administrator deletes or updates a template policy, in the context of particular organization.

When overriding a template policy for a descendant organization of Root Organization, the template policy will still apply at the Root Organization level. If the template policy is being overridden with a more restrictive policy at the descendant organization level, you should override the template policy at the Root Organization level as well. The only way to override a template policy for the Root Organization is through the database, by running the following SQL:

```
insert into ACORGPOL (acpolicy_id, member_id) values ( (select acpolicy_id from ACPOLICY where policyname = 'policyToOverride'), -2001)
```

## Levels of access control

There are two broad levels of access control in WebSphere Commerce: command level (also known as role-based) and resource level (also known as instance-level).



### Command-level or role-based access control

Command-level or role-based access control is coarse access control. It determines "who can do what". With role-based access control, you can specify that all users of a particular role can execute certain commands. Consider the access control policy, Sellers can execute sellers commands. In this policy, one of the sellers commands is the Modify Auction command. In the figure above, Jack and Tom both are sellers, so both of them can modify auctions.

Role-based access control is used for controller commands and views. This type of access control does not consider the data resource that the command acts upon. It only determines if the user is allowed to execute a particular controller command or view.

This level of access control is mandatory and is enforced by the Runtime. All controller commands must be protected by command-level access control. In addition, any view that can be called directly, or that can be launched by a redirect from another command (in contrast to being launched by forwarding to the view) must be protected by command-level access control.

**Command-level access control for controller commands:** Whenever you run a controller command, an access control policy must exist that grants users to perform the Execute action on the command resource. The resource is the interface name of the controller command. The access group is usually geared to a single role. For example, you can specify that users with the Account Representative role can execute any command in the AccountRepresentativesCmdResourceGroup resource group.

**Command Level Access Control for Views:** When a view is called directly from the URL, or is the result of a redirect from a command, it must have an access control policy. Such a policy must have the viewname specified as an action, in the ACACTION table. This action must then be associated with an action group, using the ACACTACTGP table. This action group must then be referenced in the appropriate command level policy, in the ACPOLICY table.

## Instance-level or resource-level access control

Instance-level or resource-level access control policies provide granular access control, determining who can do what command on which resources. The previous example of a role-based access control policy that allows Sellers to modify auctions, can be fine-tuned for resource-level access control to be, Sellers can modify auctions owned by the organization for which they play their role. In 25, Jack has the seller role for Seller Organization 1. Tom has the seller role for Seller Organization 2. Jack creates a furniture auction at the furniture store. Tom creates a shirt auction at the shirt store. Jack can modify the furniture auction, but *not* the shirt auction. Tom can modify the shirt auction, but *not* the furniture auction.

To summarize, first the system does a command-level access check. If the user is allowed to execute a command, a subsequent resource-level access control policy is done to determine if the user can access the resource in question.

Resource level access control applies to commands and databeans.

**Resource-level access control for commands:** After the command level access control checking has been completed, if access has been granted, then resource level checking is done in one of the following two cases:

- The command implements `getResources()` — this method specifies the instances of resources that need to be checked against the current action; where the command is now the action. The WebSphere Commerce Runtime will enforce that the current user has access to all of the resources specified by `getResources()`. By default, `getResources()` returns null, that is, it does not perform any resource level checking.
- The command calls `checkIsAllowed(Object Resource, String Action)` — in cases where the command writer does not know which resources need to be checked at the time that `getResources()` is called by the Runtime, the command can call this `checkIsAllowed()` method, as needed, to determine if the current action and resource pair is authorized. The action is usually the interface name of the current command. When this method is called, if access is denied, an exception will be thrown: `ECAppl icationException( ECMessage._ERR_USER_AUTHORITY, ..)`

**Resource level access control for databeans:** As explained above, views are protected by command level policies, which are usually based on roles. For example, the command level policy may specify that a Seller Administrator has access to a specific view. It is often necessary to further ensure that the databeans on the JSP are all related to the organization for which the user plays the Seller Administrator role. This is done by having all databeans that need protection (directly or indirectly), implement the Delegator interface. These databeans delegate to a primary (independent) databean which in turn implements the Protectable interface. A primary databean would delegate to itself, and therefore implement both interfaces. Then, whenever a databean is invoked using the Databean Manager's `activate()` method, the WebSphere Commerce Runtime will ensure that there is a policy which grants the current user the authority to perform the `Display` action on the primary databean resource.

---

## How access control prevents unauthorized actions

This section explains how policy-based access control works to ensure that users can perform only actions for which they are authorized.



## Checking for authorization before performing a user-initiated action

*Policy Manager* is the access control component that determines whether or not the current user is allowed to execute the specified action on the specified resource. Access control policies are specified in XML format. During instance creation, the default policies are loaded into the appropriate database tables. When WebSphere Commerce Application Server is started up, the access control information is cached in memory so the Policy Manager can quickly check a user's authorization when called to do so. If access control information is changed in the database through the WebSphere Commerce Administration Console, or by loading XML policy data, the access control cache needs to be updated. This can be done by updating the Access Control registry in the WebSphere Commerce Administration Console. Restarting WebSphere Commerce will also result in updating the cache.

When a user attempts to perform an access control protected action, an access control check will be done to make sure that the user is authorized. The Policy Manager looks for all the access control policies that apply to the organization that owns the resource. Then it checks those policies to evaluate if the user is authorized to perform the action on the target resource. If there is at least one such policy, the Policy Manager grants access, otherwise, access is denied.

## Using access control

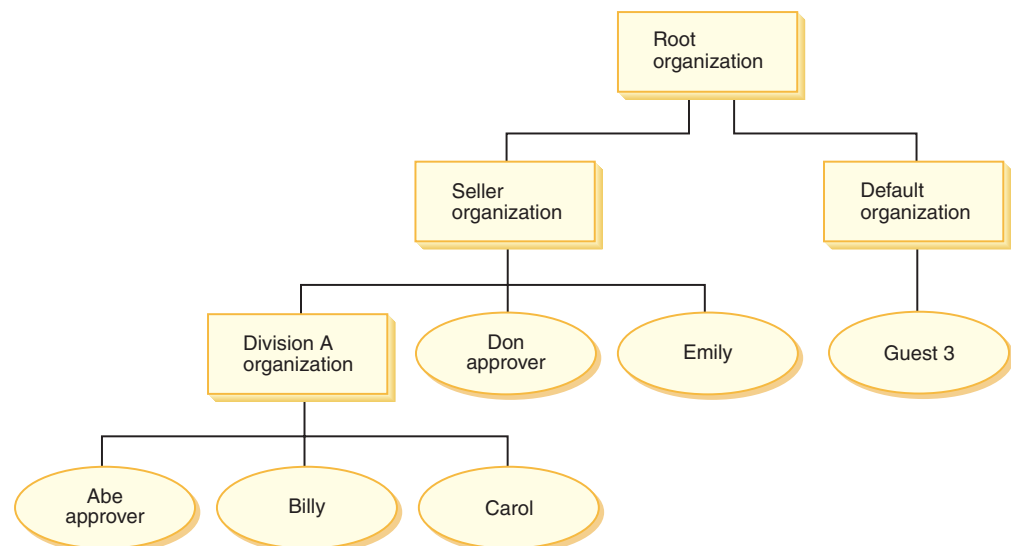
For more information on tasks such as customizing default access control policies, customizing scenarios, and using XML files to customize access control policies, see the WebSphere Commerce Access Control Guide.

---

## Evaluating access control policies

This section can be used as a guide to evaluating access control policies. In this section, you are presented with a scenario and guided through an example of how to evaluate a standard and a template access control policy. Each section begins with a description of related policies, and scenarios using each policy. For more information on standard and template policies, see "Types of access control policies" on page 23.

The following diagram graphically displays the scenario:



## Organizational hierarchy

From the diagram, you can see the following four organizations are in the site:

- Root organization
- Seller organization
- Default organization
- Division A organization

As you can see, Root organization is the parent of Seller organization and Default organization. Seller organization is the parent of Division A organization

## Users

In the diagram, Don and Emily are registered to the Seller Organization. Abe, Billy and Carol are registered to Division A organization. Guest 3 has not registered, but for access control purposes, implicitly belongs to the Default Organization.

## Roles

Don has the approver role for the Seller Organization. Abe has the approver role for the Division A organization.

## Access Groups

The following access groups are used in this scenario:

- Registered users: This group implicitly includes all users that are registered.
- Approvers for Seller: This group implicitly includes all users that have the role of approver for the Seller organization.
- Approvers for Division A: This group implicitly includes all users that have the role of approver for the Division A organization.

## Documents

The document object is a protected resource. The owner of a document is defined to be the organization where it was created.

### Access control requirements for updating documents

The following are the access control requirements for updating documents:

1. Registered users can update a document of which they are the creator.
2. Approvers for Division A can update documents owned by Division A, but not documents owned by Seller. Approvers for Seller organization can update documents owned by both Division A, and Seller organization.

## Evaluating standard policies

This section guides you through the standard policies and the scenarios to evaluate them.

### Access control policies related to updating documents

The following is the policy format and the access control policies that relate to updating documents:

Policy Format: [Access Group, Action Group, Resource Group, Relationship]

#### Policy 1:

[Registered Users, Execute Command Action Group, Update Document Resource Group, - ]

This is a standard role-based policy owned by the Root organization. In this policy, registered users can execute Update Document commands.

**Policy 2:**

[Registered Users, Update Document Action Group, document, creator ]

This is a standard resource-level policy owned by the Root organization. In this policy, registered users can update a document if they are the creator of that document.

**Policy 3:**

[Approvers for Seller, Update Document Action Group, document, - ]

This is a standard resource-level policy owned by Seller organization. In this policy, approvers for Seller can update documents that are owned by Seller.

**Policy 4:**

[Approvers for Division A, Update Document Action Group, document, - ]

This is a standard resource-level policy owned by Division A organization. In this policy, Approvers for Division A can update documents that are owned by Division A.

## Scenarios

**Scenario 1 : Billy attempts to update his own document:** The following is the access control evaluation for this scenario:

*Command - level check:*

1. There is no store ID specified, so the owner of the command is set to Root organization. So, only policies owned by Root Organization will be used to evaluate whether the user has command-level access: policies 1 and 2 are owned by Root organization.
2. Policy 1 grants access, since Billy is a member of the Registered Users access group and he is performing the Execute action on the Update Document command resource.

*Resource - level check:*

1. The Update Document command specifies that the document resource is to be protected. Billy's document is owned by Division A. So, only policies owned by Division A and its ancestor organizations will apply: policies 1, 2, 3 and 4.
2. Policy 2 grants access since Billy is a member of the Registered Users access group, he is performing the Update Document command action on the document resource, and he fulfills the creator relationship with the document.

Since Billy passed both the command-level and resource-level access control checks, he can update his own document.

**Scenario 2: Don attempts to update Carol's document:** The following is the access control evaluation for this scenario:

*Command - level check:*

1. There is no store ID specified, so the owner of the command is set to Root organization. So, only policies owned by Root Organization will be used to evaluate whether the user has command-level access: policies 1 and 2 are owned by Root organization.
2. Policy 1 grants access, since Don is a member of the Registered Users access group and he is performing the Execute action on the Update Document command resource.

*Resource - level check:*

1. The Update Document command specifies that the document resource is to be protected. Carol's document is owned by Division A. So, only policies owned by Division A and its ancestor organizations will apply: policies 1, 2, 3 and 4.
2. Policy 4 grants access since Don is a member of the Approvers for Seller access group, and he is performing the Update Document command action on the document resource

Since Don passed both the command-level and resource-level access control checks, he can update Carol's document.

**Scenario 3: Abe attempts to update Emily's document:** The following is the access control evaluation for this scenario:

*Command - level check:*

1. There is no store ID specified, so the owner of the command is set to Root organization. So, only policies owned by Root Organization will be used to evaluate whether the user has command-level access: policies 1 and 2 are owned by Root organization.
2. Policy 1 grants access, since Abe is a member of the Registered Users access group and he is performing the Execute action on the Update Document command resource.

*Resource - level check:*

1. The Update Document command specifies that the document resource is to be protected. Emily's document is owned by Seller organization. So, only policies owned by Seller organization and its ancestor organizations will apply: policies 1, 2 and 3.
2. Policy 3 does NOT grant access since Abe is NOT a member of the Approvers for the Seller access group.

Although Abe passed the command-level check, since he failed the resource-level access control check, he cannot update Emily's document.

**Scenario 4: Guest 3 attempts to update his own document:** The following is the access control evaluation for this scenario:

*Command - level check:*

1. There is no store ID specified, so the owner of the command is set to Root organization. So, only policies owned by Root Organization will be used to evaluate whether the user has command-level access: policies 1 and 2 are owned by Root organization.
2. Policy 1 does NOT grant access, since guest 3 is NOT a member of the Registered Users access group.

*Resource - level check:*

1. Resource-level checking is NOT even done since the Command-level check failed

Since Guest 3 failed the command-level check, he cannot update his own document.

## Evaluating template policies

This example is based on the previous scenario.

### Access control policies related to updating documents

When evaluating template policies, access control policies 1 and 2 used for evaluating standard policies still apply, however, standard policies 3 and 4 are now replaced by template policy 5. For more information on policies 1 and 2 see, "Evaluating standard policies" on page 28.

#### Policy 5:

[Approvers for Organization, Update Document Action Group, document, - ]

This policy is a template resource-level policy. Approvers for the organization that owns the document, can update documents.

We also need a new parameterized access group to be used by this template policy. The following access group is added to this scenario:

- Approvers for Organization: This group implicitly includes all users that have the role of approver for ? organization. (the ? parameter will be dynamically changed to the policy owner, as the template policy is applied at runtime).

### Scenarios

The following scenarios use policies 1, 2, and 5 only.

**Scenario 1: Don attempts to update Carol's document:** The following is the access control evaluation for this scenario:

*Command - level check:*

1. There is no store ID specified, so the owner of the command is set to Root organization. So, only policies owned by Root Organization will be used to evaluate whether the user has command-level access: policies 1 and 2 are owned by Root organization. During policy evaluation, template policies dynamically change ownership to the organization that owns the resource, and subsequently that organization's ancestors, so policy 5 will also apply.
2. Policy 1 grants access, since Don is a member of the Registered Users access group and he is performing the Execute action on the Update Document command resource.

*Resource - level check:*

1. The Update Document command specifies that the document resource is to be protected. Carol's document is owned by Division A. So, only policies owned by Division A and its ancestor organizations will apply: policies 1, 2. During policy evaluation, template policies dynamically change ownership to the organization that owns the resource, and subsequently that organization's ancestors, so policy 5 will also apply.
2. Template policy 5 is first applied to the organization that owns the resource: Division A. At this moment policy 5 essentially behaves like policy 5a:  
[Approvers for Division A, Update Document Action Group, document, - ] standard resource-level policy owned by Division A.

3. Policy 5a does NOT grant access, since Don is NOT a member of the Approvers for Division A access group.
4. Template policy 5 will next be applied to the parent organization of Division A: Seller organization. At this moment policy 5 essentially behaves like policy 5b:  
[Approvers for Seller, Update Document Action Group, document, - ] standard resource-level policy owned by Seller
5. Policy 5b does grant access since Don is a member of Approvers for Seller access group, and he is performing the Update Document command action on the document resource.

Since Don passed both the command-level and resource-level access control checks, he can update Carol's document.

**Scenario 2: Abe attempts to update Emily's document:** The following is the access control evaluation for this scenario:

*Command - level check:*

1. There is no store ID specified, so the owner of the command is set to Root organization. So, only policies owned by Root Organization will be used to evaluate whether the user has command-level access: policies 1 and 2 are owned by Root organization. During policy evaluation, template policies dynamically change ownership to the organization that owns the resource, and subsequently that organization's ancestors, so policy 5 will also apply.
2. Policy 1 grants access, since Abe is a member of the Registered Users access group and he is performing the Execute action on the Update Document command resource.

*Resource - level check:*

1. The Update Document command specifies that the document resource is to be protected. Emily's document is owned by Seller organization. So, only policies owned by Seller and its ancestor organizations will apply: policies 1, 2. During policy evaluation, template policies dynamically change ownership to the organization that owns the resource, and subsequently that organization's ancestors, so policy 5 will also apply.
2. Template policy 5 is first applied to the organization that owns the resource: Seller organization. At this moment policy 5 essentially behaves like policy 5a:  
[Approvers for Seller, Update Document Action Group, document, - ] standard resource-level policy owned by Seller
3. Policy 5a does NOT grant access, since Abe is NOT a member of the Approvers for Seller access group.
4. Template policy 5 will next be applied to the parent organization of Seller organization: Root organization. At this moment policy 5 essentially behaves like policy 5b:  
[Approvers for Root, Update Document Action Group, document, - ] standard resource-level policy owned by Root
5. Policy 5b does NOT grant access since Abe is NOT a member of Approvers for Root access group.
6. Root organization does not have a parent organization, so template policy 5 has been completely evaluated.

Although Abe passed the command-level check, since he failed the resource-level access control check, he cannot update Emily's document.

---

## **Part 2. WebSphere Commerce site administrator security tasks**

This part describes the security tasks that can typically be performed by the WebSphere Commerce site administrator.





---

## Chapter 4. Enhancing site security

To enhance the security of your WebSphere Commerce site, you can enable any of the following features in WebSphere Commerce Configuration Manager:

- Log off a user that is inactive for an extended period and request they log back on to the system, using the Login Timeout node. For details, see “Enabling login timeout” on page 38.
- Require users to change their passwords when they are logging in to the system for the first time, using the Password Invalidation node. For details, see “Activating password invalidation” on page 39.
- Require users to enter their passwords if they are running requests that run designated commands, using the Password Protected Commands node. For details, see “Enabling password protected commands” on page 39.
- Update encrypted data such as passwords and credit card information as well as the merchant key in a WebSphere Commerce database, using the Database Update Tool node. For details, see “Updating encrypted data” on page 40.
- Reject any user request that contain attributes or characters that are designated as not allowed, using the Cross Site Scripting Protection node. For details, see “Enabling cross site scripting protection” on page 41.
- Quickly identify any security threats against WebSphere Commerce by enabling access logging. For details, see “Enabling access logging” on page 43.

In addition, you can enable the following features from the Security drop-down in the WebSphere Commerce Administration Console:

- Set up an account policy for your site to define the account-related policies in use, by using the Account policy page. For details, see “Setting up an account policy” on page 44.
- Set up a password policy for your site to control a user’s password selection characteristics using the Password policy page (only if users are authenticated against the WebSphere Commerce database). For details, see “Setting up a password policy” on page 45.
- Set up an account lockout policy for your site to reduce the chances of a user account being compromised, using the Account lockout policy page (only if users are authenticated against the WebSphere Commerce database). For details, see “Setting up an account lockout policy” on page 46.
- Launch a security program that checks and deletes temporary WebSphere Commerce files that may contain potential security exposures using the Launch security check page. For details, see “Launching a security check” on page 47.

For information on related concepts, see the following topics in the WebSphere Commerce online help:

- Configuration Manager
- WebSphere Commerce configuration file
- Administration Console
- Security

For information on related tasks, see the following topics in the WebSphere Commerce online help.

- Launch the Configuration Manager

- Open the Administration Console

---

## Views for security

Before using certain security features of WebSphere Commerce, you are required to define the associated views for your store before you can use that feature. The following information describes how to define the views for:

- Login timeout (see “Login timeout”)
- Password invalidation (see “Password invalidation”)
- Password protected commands (see “Password protected commands” on page 37)
- Cross site scripting protection (see “Cross site scripting protection” on page 38)

For general information on creating views and developing your store front, see the *Store Developer’s Guide*.

### Login timeout

To use the login timeout security feature, you need to define the `LoginTimeoutErrorView` and `ReLogonFormView` views for your store.

#### LoginTimeoutErrorView

If the login timeout information is incorrect, WebSphere Commerce redirects the user’s browser to this view. If this occurs, it is likely because someone has tampered with the cookie.

*Table 1. LoginTimeoutErrorView attributes*

<code>ECConstants.EC_LOGIN_TIMEOUT_ERROR_MSGCODE</code>	1	Expiry time is set to wrong value.
	2	Logon time is set to wrong value.
	3	Expiry or logon time set to wrong value.

#### ReLogonFormView

This view is displayed to users after their session has expired. It needs to provide the user with a form to enter the user’s logon ID and password. The submit button will invoke the Logon command. There should also be a Cancel button to redirect the user to another page, in most cases, the store front page.

There are no attributes for `ReLogonFormView`.

*Table 2. ReLogonFormView form attributes*

<code>ECUserConstants.EC_UREG_LOGONID</code>	The user’s logon id.
<code>ECUserConstants.EC_UREG_LOGONPASSWORD</code>	The user’s logon password.
<code>ECUserConstants.EC_RELOGIN_URL</code>	The URL that is displayed if the credentials provided are invalid. In most cases, it will be name of this view.
<code>ECConstants.EC_STORE_ID</code>	The store identifier.
<code>ECConstants.EC_URL</code>	The URL that is displayed when the credentials that are entered belong to different user. In most cases, this should be a store home page, or the same URL that is used in a store logon page.

### Password invalidation

To use the password invalidation security feature, you need to define the `ChangePassword` view for your store.

## ChangePassword

This view is displayed if a user's password has expired. It should provide the user with a form to enter the current (expired) password and a new password. The Submit button invokes the ResetPassword command. There should also be a Cancel button that redirects the user to another page, in most cases, the store front page.

*Table 3. ChangePassword attributes*

ECConstants.EC_PASSWORD_EXPIRED_FLAG	1	The user's password has expired. This attribute is required in order to distinguish this view from the view used for the password change feature as they are the same. The view for the password change could be invoked by a user, and the JSP assigned to this view should be the same for both cases. The JSP should look for this attribute in order to decide what to display.
ECUserConstants.EC_UREG_LOGONID	null	The attribute is not on a URL . This is normal password change behavior
ECConstants.EC_LOGIN_RETURN_URL		The current user logon id. The URL to which the browser is redirected after a successful password change. This URL will be passed to an action command under the name ECConstants.EC_URL.

*Table 4. ChangePassword form attributes*

ECUserConstants.EC_UREG_LOGONID		The logon ID id of the user. The current logon ID has been passed in to the view.
ECUserConstants.EC_UREG_LOGONPASSWORDOLD		The old password.
ECUserConstants.EC_UREG_LOGONPASSWORD		The new password.
ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY		The new password verification.
ECConstants.EC_URL		The URL where users are redirected after a successful password change. The value has been passed in to the view.
ECUserConstants.EC_RELOGIN_URL		The URL where the browser is redirected if the password change is not successful.

## Password protected commands

To use the password protected commands security feature, you need to define the PasswordReEnterErrorView and the PasswordReEnterFormView views for your store.

### PasswordReEnterErrorView

This view is used in the following scenarios:

- A user fails to provide the correct password and is logged off.
- The authentication has failed.

In both cases, the user should have a way to continue to another page through a link on the current page.

*Table 5. PasswordReEnterErrorView attributes*

ECConstants.EC_PASSWORD_REREQUEST_MSGCODE	0	A problem occurred when attempting to authenticate the user.
	null	The attribute is not on a URL. The user failed to provide the password is logged off.

### PasswordReEnterFormView

This view is displayed when user tries to execute a password protected command. It should provide user with form to enter password. There should be two entry fields for password.

*Table 6. PasswordReEnterFormView attributes*

ECConstants.EC_PASSWORD_REREQUEST_URL	The URL is run using the Submit button of the form.
ECConstants.EC_PASSWORD_REREQUEST_MSGCODE	The message code specifying the message that is shown to the user:
1	The passwords that were entered do not match.
2	A password was not entered.
3	An incorrect password was entered.

ACTION: The URL is passed in as a parameter named:

*Table 7. PasswordReEnterFormView form attributes*

ECConstants.EC_PASSWORD_REREQUEST_PASSWORD1	The first password.
ECConstants.EC_PASSWORD_REREQUEST_PASSWORD2	The second password.

## Cross site scripting protection

To use the cross site scripting security feature, you need to define the ProhibitedAttrsErrorView, ProhibitedCharacterErrorView, and ProhibCharEncodingErrorView views for your store.

### ProhibitedAttrsErrorView

This view is shown to the user when the request is not processed because it contained prohibited attributes.

### ProhibitedCharacterErrorView

This view is shown to the user when the request is not processed because it contained prohibited characters

### ProhibCharEncodingErrorView

This is the same as ProhibitedCharacterErrorView, above.

---

## Enabling login timeout

**Note:** To use the login timeout security feature for a store, you need to define the LoginTimeoutErrorView and ReLogonFormView views for the store as described in “Login timeout” on page 36.

Use the Login Timeout node of the Configuration Manager to enable or disable the login timeout feature. When this feature is enabled, a WebSphere Commerce user that is inactive for an extended period of time is logged off the system and requested to log back on. If the user subsequently logs on successfully, WebSphere Commerce runs the original request that was made by the user. If the user logon fails, the original request is discarded and the user remains logged off the system.

Note that for WebSphere Commerce tools (such the Administration Console, WebSphere Commerce Accelerator, Store Services, and so on), the login timeout feature does not present a relogin page to the user. Instead, it closes the browser window and it is up to the user to log back on to the tool. Thus, in the case of tools, the original request that the user submits is not processed.

To enable this feature:

1. Launch the Configuration Manager and traverse to the Login Timeout node for your instance as follows: **WebSphere Commerce** > *host\_name* > **Instance List** > *instance\_name* > **Instance Properties** > **Login Timeout**
2. To activate the login timeout feature, click the **Enable** check box.
3. Enter the login timeout value, in seconds, in the Value field.
4. To apply your changes to Configuration Manager, click **Apply**.

5. Upon successfully updating the configuration for your instance, you will receive a message indicating a successful update.
6. From the WebSphere Application Server Administration Console, stop then restart the WebSphere Commerce Server instance.

Note that the login timeout value is stored in the *instance.xml* file in milliseconds, while the value in Configuration Manager is entered in seconds.

---

## Activating password invalidation

**Note:** To use the password invalidation security feature, you need to define the ChangePassword view for your store as described in “Password invalidation” on page 36.

Use the Password Invalidation node of the Configuration Manager to enable or disable the password invalidation feature. Password invalidation, when enabled, requires WebSphere Commerce users to change their password if the user’s password has expired. In this case, the user is redirected to a page where they are required to change their password. Users are not able to access any secure pages on the site until they have changed their password. To enable this feature:

1. Launch the Configuration Manager and traverse to the Password Invalidation node for your instance as follows: **WebSphere Commerce** > *host\_name* > **Instance List** > *instance\_name* > **Instance Properties** > **Password Invalidation**
2. To activate the password invalidation feature, click the **Enable** check box.
3. To apply your changes to Configuration Manager, click **Apply**.
4. Upon successfully updating the configuration for your instance, you will receive a message indicating a successful update.
5. From the WebSphere Application Server Administration Console, stop then restart the WebSphere Commerce Server instance.

---

## Enabling password protected commands

**Note:** To use the password protected commands security feature, you need to define the PasswordReEnterErrorView and the PasswordReEnterFormView views for your store as described in “Password protected commands” on page 37.

Use the Password Protected Commands node of the Configuration Manager to enable or disable the password protected commands feature. When this feature is enabled, WebSphere Commerce requires registered who are logged onto WebSphere Commerce to enter their password before continuing a request that runs designated WebSphere Commerce commands.

**Caution:** When you configure password protected commands, some of the commands shown in the command selection list can be executed by generic or guest users. Configuring such commands as password protected will restrict generic and guest users from running them. Therefore, you should exercise caution when you configure commands to be password protected.

To enable this feature:

1. Launch the Configuration Manager and traverse to the Password Protected Commands node for your instance as follows: **WebSphere Commerce** > *host\_name* > **Instance List** > *instance\_name* > **Instance Properties** > **Password Protected Commands**
2. In the General tab:
  - a. To activate the password protected commands feature, click **Enable**.
  - b. Enter number of retries in the Retries field. (The default number of retries is 3.)
3. In the Advanced tab:
  - a. Select a WebSphere Commerce command you wish to protect from the list in the Password Protected Command List window and click **Add**. The command you have selected is listed in the Current Password Protected List window.
  - b. If you wish to disable password protection for any WebSphere Commerce command, select the command in the Current Password Protected Command list window and click **Remove**.
4. To apply your changes to Configuration Manager, click **Apply**.
5. Upon successfully updating the configuration for your instance, you will receive a message indicating a successful update.
6. From the WebSphere Application Server Administration Console, stop then restart the WebSphere Commerce Server instance.


**Note:** WebSphere Commerce will only display the commands that are designated as authenticated or set with the https flag in the URLREG table in the list of available commands.

---

## Updating encrypted data

Use the Database Update Tool available from the Database node of the Configuration Manager to update all encrypted data (for example, passwords or credit card numbers) as well as the merchant key in a WebSphere Commerce database for a given instance. To use the tool:

1. Launch the Configuration Manager and traverse to your specific database entry as follows: **WebSphere Commerce** > *host\_name* > **Instance List** > *instance\_name* > **Instance Properties** > **Database** > *database\_name*
2. Right-click on *database\_name* and select **Run Database Update Tool**
  - Select **Update all databases for this instance** to migrate encrypted data for all databases for the selected instance.
 

 As iSeries supports a single database configuration, this option does not apply to iSeries.
  - Select **Update selected database** to migrate encrypted data for a specific database by selecting the database from the drop-down list (default).
3. Select an action you want to run from the Action Item box, and fill in the required information in the Parameter field:

Actions	Parameters	Action Required
---------	------------	-----------------

Change Merchant Key	Old Merchant Key	Enter your existing merchant key that you used when you created your current WebSphere Commerce instance.
	New Merchant Key	Enter your new merchant key. This is a 16-digit hexadecimal number for the Configuration Manager to re-encrypt the currently encrypted data. The Merchant Key must have at least one alphanumeric character (a to f) and at least one numeric character (0 to 9). Any alphanumeric character must be entered in lower case letters, and you cannot have the same character entered more than four times in a row.

4. Click **OK** to run the database update tool for your selected WebSphere Commerce database or for all your WebSphere Commerce databases.
5. Upon successfully updating the configuration for your instance, you will receive a message indicating a successful update.
6. From the WebSphere Application Server Administration Console, stop then restart the WebSphere Commerce Server instance.

---

## Enabling cross site scripting protection

**Note:** To use the cross site scripting security feature for a store, you need to define the `ProhibitedAttrsErrorView`, `ProhibitedCharacterErrorView`, and `ProhibCharEncodingErrorView` views for the store as described in “Cross site scripting protection” on page 38.

Use the Cross Site Scripting Protection node of the Configuration Manager to enable or disable cross site scripting protection for your instance. When enabled, cross site scripting protection rejects any user requests that contain attributes or strings that are designated as not allowable. You can specify the disallowed attributes and strings in this node of the Configuration Manager. You can also exclude commands from cross site scripting protection by allowing the values of specified attributes for that particular command to contain prohibited strings. Cross site scripting protection is disabled by default.

**Warning:** Cross site scripting protection is a restrictive feature in that it will restrict the execution of the commands based on the configuration. The feature does not check what attributes or strings have been defined as prohibited, so when you configure it, make sure that prohibited attributes are not those used by the commands. Also make sure the prohibited strings are not values that are usually passed to the commands. Use extreme caution when configuring this feature.

To enable this feature:

1. Launch the Configuration Manager and traverse to the Cross Site Scripting Protection node for your instance as follows: **WebSphere Commerce** > *host\_name* > **Instance List** > *instance\_name* > **Instance Properties** > **Cross Site Scripting Protection**
2. Use the General tab to activate the cross site scripting protection feature, as follows:
  - a. Click **Enable**.

- b. To add attributes that you wish to disallow for WebSphere Commerce commands, right-click on the Prohibited Attributes table and select **Add row**. Type the attribute that you wish to disallow. You can only specify one attribute per row.
- c. To remove attributes from the Prohibited Attributes table, highlight and right-click the line containing the attribute in the table and select **Delete row**.
- d. To add strings that you wish to disallow for WebSphere Commerce commands, right-click on the Prohibited Characters table and select **Add row**. Add the string that you wish to disallow. You can only specify one string per row.
- e. To remove characters from the Prohibited Characters table, highlight and right-click the line containing the character in the Prohibited Characters table and select **Delete row**.

**Note:** The following strings are specified by default in the prohibited characters fields. These strings are most commonly used as scripting tags in malicious cross site scripting attacks:

- <SCRIPT
- &lt;SCRIPT
- <% and &lt;%;

3. Use the Advanced tab to exclude WebSphere Commerce commands from cross site scripting protection by allowing the values of specified attributes for that particular command to contain prohibited strings as follows:
  - a. Select the commands from the Command List box.
  - b. Type in a list of attributes, separated by commas, for which prohibited characters are allowed in the List of Excepted Attributes window and click **Add**.
  - c. To remove a command along with its attributes, select the command from the List of Excepted Commands window and click **Remove**.

You can also remove specific attributes from a command by selecting the attribute and clicking **Remove**.
4. To apply your changes to Configuration Manager, click **Apply**.
5. Upon successfully updating the configuration for your instance, you will receive a message indicating a successful update.
6. From the WebSphere Application Server Administration Console, stop then restart the WebSphere Commerce Server instance.

**Notes:**

1. When commands are excluded from cross site scripting protection, the values of specified attributes will be encoded using HTML encoding of symbols. For example, the command `cmd1?user=<Thomas>` is encoded as `ascmd1?user=&#60;Thomas&#62;`
2. When you specify the string in the prohibited characters fields, be aware that:
  - A certain sequence of characters can cause the string to be converted to a single character in compliance with URL encoding standards. For example, the string `<%bb` would be converted into a string `<X` where X is a single character which has a hexadecimal representation value of HEX 'bb' (decimal 187). In this case the string `<%bb` will not be caught by cross site scripting protection if passed in a URL.



- A certain sequence of characters can cause the string conversion to fail if they do not comply with URL encoding standards. For example, the string `<%gg` would cause conversion to fail since HEX 'gg' is not a valid hexadecimal value representation. In this case, the string `<%gg` will cause an exception, resulting in no response to the URL request containing such a string, whether or not cross site scripting protection is enabled.

**Example:** Consider the following examples:

- Prohibited strings: `<SCRIPT, <%`  
Prohibited attributes: `mycomment, description`

Command	Status
<code>cmd1?description=Available...</code>	rejected
<code>cmd2?userid=Thomas...</code>	accepted
<code>cmd3?mycomment=&lt;SCRIPT&gt;...</code>	rejected
<code>cmd4?password=&lt;%...%&gt;...</code>	rejected

- If you wish to allow the attribute text of the `cmd1` command to contain prohibited strings (`<SCRIPT, <%`), and not for other attributes, for example, the attribute `txt`, you can exclude `cmd1` and specify `txt` as the excepted attribute.

Command	Status
<code>cmd1?text=&lt;SCRIPT&gt;...</code>	accepted
<code>cmd1?text=&lt;%...%&gt;...</code>	accepted
<code>cmd1?txt=&lt;SCRIPT&gt;...</code>	rejected
<code>cmd1?txt=&lt;%..%&gt;...</code>	rejected

## Enabling access logging

When enabled, the access logging feature logs either all incoming requests to the WebSphere Commerce server or only the requests resulting in access violations. Examples of access violations are authentication failure, insufficient authority to execute a command. When enabled, access logging allows a WebSphere Commerce administrator to quickly identify security threats to the WebSphere Commerce system.

When an authentication failure or authorization failure event occurs, the following information is logged to the access log file database tables, `ACCCLOGMAIN` and `ACCCLOGSUB`:

- Host name of the client
- ID of the thread running the command
- User ID of the client
- Time the event occurred
- Command that was run
- Store for which the command was run
- Resource on which the operation was performed
- Result of the access control check

To enable access logging, do the following:

1. Launch the Configuration Manager.

2. Select **Host name > Instance > Instance\_List** and then open the **Components** folder.
3. Select **AccessLoggingEventListener**.
4. In the General panel, activate the **Enable Component** check-box.
5. Select the Advanced panel and enable **Start**.
6. Click **Apply**.
7. Exit Configuration Manager.
8. Restart the WebSphere Application Server.

To change the size of the log file, or to specify whether all requests are logged or not, you need to manually edit the *instance.xml* file for your WebSphere Commerce instance located in the WebSphere Commerce instances subdirectory:

1. Open the *instance.xml* file for your instance in an editor.
2. Locate the following node, which is located in the <LogSystem>/<activitylog> node:

```
<accessLogging cacheSize="aa" logAllRequests="bbbb" />
```

where:

- *aa* is an integer value specifying the maximum number of entries that will be logged to memory before entries are written to the database. Generally a higher number will result in improved performance with respect to access logging. The default value is 32.
  - *bbbb* is either true or false. A value of true means that all incoming requests are logged. A value of false means that only access violations are logged. To prevent excessive or unnecessary logging, a value of false is recommended. Use true only when you suspect authentication problems or security contravention at your site. The default value is false.
3. When you have completed your updates, save the *instance.xml* file for your WebSphere Commerce instance.
  4. Restart the WebSphere Application Server.

In the following example, the access logging keeps 3 entries in memory before logging entries to the database tables. In addition, it logs all incoming requests to the WebSphere Commerce server:

```
<accessLogging cacheSize="3" logAllRequests="true" />
```

---

## Setting up an account policy

The Account Policy page of the WebSphere Commerce Administration Console allows you to set up an account policy. This page lists all existing account policies including any predefined ones supplied with WebSphere Commerce by default. An account policy defines the account-related policies such as password and account lockout policies. On this page:

- You can create a new account policy by clicking **New**.
- You can change the characteristics an existing account policy by selecting the policy in the list and clicking **Change**.
- You can delete an existing account policy by selecting the policy in the list and clicking **Delete**.

To create a new account policy:

1. Open the Administration Console.

2. From the Security drop-down menu of the Administration Console, click **Account Policy**.
3. On the Account Policy page, click **New** to create a new account policy.
4. Enter a name for the account policy in the Name field (for example, my\_account\_policy).
5. From the Password policy menu, select a preexisting password policy.
6. From the Account lockout policy menu, select a preexisting account lockout policy.
7. Click **OK**.

Once you have created an account policy, you can assign the policy to a user. Note that you cannot delete an account policy if it is in use (that is, a user is assigned to the account policy).

Also see the reference topic "Default Authentication Policies" in the WebSphere Commerce online help.

---

## Setting up a password policy

The Password Policy page of the WebSphere Commerce Administration Console allows you to control a user's password selection in order to define the characteristics of the password to ensure that it complies with the security policy for your site. This page lists all existing password policies including any predefined ones supplied with WebSphere Commerce by default.

A password policy defines attributes with which the password must comply. The password policy enforces the following conditions:

- Whether the user ID and password can match.
- Maximum occurrence of consecutive characters.
- Maximum instances of any character.
- Maximum lifetime of the passwords.
- Minimum number of alphabetic characters.
- Minimum number of numeric characters.
- Minimum length of password.
- Whether the user's previous password can be reused.
- You can create a new password policy by clicking **New**.
- You can change the characteristics an existing password policy by selecting the policy in the list and clicking **Change**.
- You can delete an existing policy by selecting the password policy in the list and clicking **Delete**.

To create a new password policy:

1. Open the Administration Console.
2. From the Security drop-down menu of the Administration Console, click **Password Policy**.
3. On the Password Policy page, click **New** to create a new password policy.
4. Enter a name for the password policy in the Name field (for example, my\_password\_policy)
5. Update the following as required to modify any of the values from the default value for shoppers:

- **Can the userID and password match?** Defines whether the userID and password can be identical or not. Select either Yes or No from the list.
- **Maximum consecutive character types.** Defines the maximum occurrence of consecutive characters in a password. The minimum value is 2 consecutive characters. For example, with a value of 2, a user cannot enter a password such as aaabc.
- **Maximum instances of any character.** Defines the maximum number of times the same character can appear in a password. The minimum value is 1 instance of a character. For example, with a value of 2, a user cannot enter a password such as abcaabc.
- **Maximum lifetime of the passwords.** Defines the maximum amount of time, in days, that a password can exist. The minimum value is 1 day. After this time period, a user is prompted to change their password.
- **Minimum number of alphabetic characters.** Defines the minimum number of alphabetic characters that need to be in a password. The minimum value is 0 alphabetic characters.
- **Minimum number of numeric characters.** Defines the minimum number of numeric characters that need to be in a password. The minimum value is 0 numeric characters.
- **Minimum length of password.** Defines the smallest length of a password, in characters. The minimum value is 1 character.
- **Can the password be reused?** Defines whether a user's previous password can be reused. Select either yes or no from the list.

6. Click **OK**.

**Notes:**

1. You cannot delete a password policy if it is in use (that is, a user is assigned to the password policy).
2. Password policies are enforced only if users are authenticated against the WebSphere Commerce database.

Also see the reference topic "Default Authentication Policies" in the WebSphere Commerce online help.

---

## Setting up an account lockout policy

The Account Lockout Policy page of the WebSphere Commerce Administration Console allows you to set up an account lockout policy for different user roles within WebSphere Commerce. This page lists all existing account lockout policies including any predefined ones supplied with WebSphere Commerce by default. An account lockout policy disables a user account if malicious actions are launched against that account in order to reduce the chances that the actions compromise the account.

An account lockout policy enforces the following items:

- The account lockout threshold. This is the number of invalid logon attempts before the account is disabled.
- Consecutive unsuccessful login delay. This is the time period for which the user is not allowed to login, after two failed attempts to login. The delay gets incremented by the configured time delay value (for example, 10 seconds) with every consecutive login failure.

To set the account lockout policy:

1. Open the Administration Console.
2. From the Security drop-down menu of the Administration Console, click **Account Lockout Policy**.
3. The Account Lockout Policy page lists all existing account lockout policies. On this page:
  - You can create a new policy by clicking **New**.
  - You can change the characteristics an existing policy by selecting the policy in the list and clicking **Change**.
  - You can delete an existing policy by selecting the policy in the list and clicking **Delete**.

For a new account lockout policy, in the Account Lockout Policy page:

1. Enter a name for the account lockout policy in the Name field (for example, my\_policy).
2. Enter an account lockout threshold in the Account lockout threshold field. For example, enter 6 (for six attempts)
3. Enter the consecutive unsuccessful login delay in seconds in the Wait time field. For enter 10 (for ten seconds).
4. Click **OK**.

**Notes:**

1. You cannot delete an account lockout policy if it is in use (that is, a user is assigned to the account lockout policy).
2. Account lockout policies are enforced only if users are authenticated against the WebSphere Commerce database.

## Launching a security check

**400** This feature is not applicable on WebSphere Commerce for iSeries.

The Launch Security Check page of the WebSphere Commerce Administration Console allows you to manually launch a security program that checks and deletes temporary WebSphere Commerce files that may contain potential security exposures. Normally the security check program runs as a scheduled job and by default is set to run once a month.

To invoke the security check program:

1. Open the Administration Console.
2. From the Security drop-down menu of the Administration Console, click **Security Checker**.
3. On the Launch Security Check page, click **Launch**.

The results of the security check, including all actions taken by the program are written to the Security check log window and to the sec\_check.log file in the log subdirectory:


**NT** `drive:\WebSphere\Commerce\instances\instance_name\log`

**2000** `drive:\Program Files\WebSphere\Commerce\instances\instance_name\log`

**AIX** `/usr/lpp/Commerce/instances/instance_name/log`

**Solaris** `/opt/WebSphere/Commerce/instances/instance_name/log`

 /opt/WebSphere/Commerce/instances/*instance\_name*/log

 On non-Windows platforms, file permissions are automatically set by WebSphere Commerce in order that sensitive files cannot be accessed by unauthorized users. On Windows platforms, you need to set the permissions manually as follows. This procedure ensures that only the Administrators group has the read/write/execute right in for sensitive files:

1. In Windows Explorer, right-click on the *drive*:\WebSphere folder.
2. Click **Properties** and **Security**. By default the "Everyone" group has the **all** permission for this folder.
3. Click **Add**.
4. A window displays (Select users, computers...). In this window, select the **Administrators** Group.

**Note:** This can be a bit ambiguous here, since you may see Administrator as a user, but you need to add the Administrator group, not the Administrator user.

Click **Add** and then click **OK**.

5. In the Security tab, the Administrators Group has been added. You need to remove "Everyone". Select **Everyone** and uncheck the box that says "Allow inheritable permission...."
6. Click **Remove** on the Security window that is displayed.

---

## Configuration Manager PDI Encrypt field

When configuring your WebSphere Commerce instance, it is recommended that you select the PDI Encrypt check box. Enabling the PDI Encrypt field specifies that information in the ORDPAYINFO and ORDPAYMTHD tables should be encrypted. By selecting the check box, payment information is stored in the WebSphere Commerce database in encrypted format.

---





## Chapter 5. Enabling WebSphere Application Server security

This chapter describes how to enable security for WebSphere Application Server. Enabling WebSphere Application Server security prevents all Enterprise JavaBean components from being exposed to remote invocation by anyone.

### Important note on WebSphere Application Server default certificates

WebSphere Application Server security and Payments are configured to use the DummyServerKeyFile.jks and DummyServerTrustFile.jks files with the default self-signed certificate *out-of-the-box*. Using the dummy key and trust file certificates is not safe and consequently you should generate your own certificate to replace the dummy certificates immediately. Please refer to the WebSphere Application Server Security Guide for more information on the dummy key and trust file certificates and how to replace them. You can access this Guide at:

[http://www.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/pdf/nav\\_Securityguide.pdf](http://www.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/pdf/nav_Securityguide.pdf)





**Note:**     When enabling WebSphere Application Server security it is strongly recommended that your machine meets the following requirements:


- A minimum machine memory of 1 GB.
- A minimum heap size of 384 MB, for the WebSphere Commerce application.


---

### Before you begin

Before you begin to enable security, you will need to know how the WebSphere Application Server where you are enabling security, validates user IDs. WebSphere Application Server can use either LDAP or the operating system's user registry as the WebSphere Application Server user registry.


    For information on the latest eFixes required to run WebSphere Application Server security, reference the latest WebSphere Commerce 5.4 README document that available from the WebSphere Commerce Web site at:


 [http://www.ibm.com/software/webservers/commerce/wc\\_be/lit-tech-general.html](http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html)

 [http://www.ibm.com/software/webservers/commerce/wc\\_pe/lit-tech-general.html](http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html)

---

### Enabling security with an LDAP user registry

 To enable WebSphere Application Server security when you are using LDAP as the WebSphere Application Server user registry, log into the system as a user with administrative authority, and perform the following steps.

 To enable WebSphere Application Server security when you are using LDAP as the WebSphere Application Server user registry, log into the system, and perform the following steps.

 To enable WebSphere Application Server security when you are using LDAP as the WebSphere Application Server user registry, log into the system as wasuser and perform the following steps.

1. Start the WebSphere Application Server Administration Server and open the WebSphere Application Server Administrator's Console.
2. In the Console, modify the global security settings as follows:
  - a. From the Console menu, select **Security Center**.
  - b. On the General tab, select the **Enable Security**.
  - c. On the **Authentication** tab, select Lightweight Third Party Authentication (LTPA). Fill in the LTPA settings, and uncheck the **Enable Single Sign On** check box if you do not want to use this functionality. Fill in the **LDAP Settings** tab as follows, depending on the type of directory server you are using:



Table 8. SecureWay Users

Field Name	Definition	Sample Values	Notes
Security Server ID	User ID	<i>user_ID</i>	<ul style="list-style-type: none"> <li>• This must not be the LDAP administrator.</li> <li>• Do not use a user that has been specified as cn=xxx.</li> <li>• Ensure that the objectclass of this user is compatible with the objectclass specified in the User Filter field of the LDAP Advanced Properties window.</li> </ul>
Security Server password	User Password	<i>password</i>	
Directory Type	Type of LDAP server	SecureWay	
Host	Host name of the LDAP server	<i>hostname.domain.com</i>	
Port	Port that the LDAP server is using		This field is not required
Base Distinguished Name	Distinguished Name under which searching occurs	o=ibm,c=us	
Bind Distinguished Name	Distinguished Name for binding to the directory when searching		This field is not required
Bind Password	Password for the Bind Distinguished Name		This field is not required



Table 9. Netscape Users

Field Name	Definition	Sample Values	Notes
Security Server ID	User ID	<i>user_ID</i>	<ul style="list-style-type: none"> <li>• This must not be the LDAP administrator.</li> <li>• Do not use a user that has been specified as cn=xxx.</li> <li>• Ensure that the objectclass of this user is compatible with the objectclass specified in the User Filter field of the LDAP Advanced Properties window.</li> </ul>
Security Server password	User Password	<i>password</i>	
Directory Type	Type of LDAP server	Netscape	
Host	Host name of the LDAP server	<i>hostname.domain.com</i>	
Port	Port that the LDAP server is using		This field is not required
Base Distinguished Name	Distinguished Name under which searching occurs	<i>o=ibm</i>	
Bind Distinguished Name	Distinguished Name for binding to the directory when searching		This field is not required
Bind Password	Password for the Bind Distinguished Name		This field is not required

Table 10. Domino Users

Field Name	Definition	Sample Values	Notes
Security Server ID	Short Name/User ID	<i>user_ID</i>	Ensure that the objectclass of this user is compatible with the objectclass specified in the User Filter field of the LDAP Advanced Properties window.
Security Server password	User Password	<i>password</i>	
Directory Type	Type of LDAP server	Domino™ 5.0	

Table 10. Domino Users (continued)

Field Name	Definition	Sample Values	Notes
Host	Host name of the LDAP server	<i>hostname.domain.com</i>	
Port	Port that the LDAP server is using		This field is not required
Base Distinguished Name	Distinguished Name under which searching occurs		This field is not required
Bind Distinguished Name	Distinguished Name for binding to the directory when searching		This field is not required
Bind Password	Password for the Bind Distinguished Name		This field is not required



**Windows**

Table 11. Active Directory Users



Field Name	Definition	Sample Values	Notes
Security Server ID	sAMAccountName	<i>user_ID</i>	<ul style="list-style-type: none"> <li>• User Logon Name of any ordinary user.</li> <li>• Do not use a user that has been specified as cn=xxx.</li> <li>• Ensure that the objectclass of this user is compatible with the objectclass specified in the User Filter field of the LDAP Advanced Properties window.</li> </ul>
Security Server password	User Password	<i>password</i>	
Directory Type	Type of LDAP server	Active Directory	
Host	Host name of the LDAP server	<i>hostname.domain.com</i>	
Port	Port that the LDAP server is using		This field is not required
Base Distinguished Name	Distinguished Name under which searching occurs	CN=users, DC=domain1, DC=domain2, DC=com	




Table 11. Active Directory Users (continued)







Field Name	Definition	Sample Values	Notes
Bind Distinguished Name	Distinguished Name for binding to the directory when searching	CN= <i>user_ID</i> , CN=users, DC=domain1, DC=domain2, DC=com	The <i>user_ID</i> value is the Display Name. This is not necessarily the same as the User Logon Name.
Bind Password	Password for the Bind Distinguished Name	<i>bind_password</i>	This should be the same as the Security Server Password.

- d.   Restart the WebSphere Application Server Administration Server, and then reopen the WebSphere Application Server Administrator's Console.
  - e. On the **Role Mapping** tab, select the WCS appserver and click the **Edit Mappings...** button.
    - 1) Select the WCSecurity Role and click the **Select...** button.
    - 2) Check the **Select users/groups** check box and add the userID that was entered in step 2c on page 50.
  - f. Click **Finish**.
3. Close the administrative console, and stop and restart the WebSphere Application Server Administration Server. From now on, when you open the WebSphere Application Server Administrator's Console, you will be prompted for the Security Server ID and password.
  4. Open the WebSphere Commerce Configuration Manager and select **Instances > instance\_name > Instance Properties > Security** and click the **Enable** check box. You are prompted to enter the user name and password that you entered in step 2c on page 50. Click **Apply** then exit Configuration Manager.
  5. Stop and restart the WebSphere Application Server administration server.

## Enabling security with an operating system user registry

  To enable WebSphere Application Server security when you are using the operating system user validation as the WebSphere Application Server user registry, log in as a user with administrative authority and perform the following steps.

   To use the operating system as a user registry, WebSphere Application Server needs to be run as root. Run WebSphere Application Server as root and perform the following steps.






1.    Login as root.
2.    Start the WebSphere Application Server and launch the WebSphere Application Server Administration Console while logged in as root:
 



```
export DISPLAY=fully_qualified_host_name:0.0
cd WAS_HOME/bin
./startupServer.sh &
./adminclient.sh remote_WAS_host_name port
```

where *fully\_qualified\_host\_name* is the name of the computer you are using to access the WebSphere Application Server Administration Console, *remote\_WAS\_host\_name* is the fully-qualified host name of the WebSphere

Application Server, and port is the port through which you are accessing the WebSphere Application Server (the default port is 2222).

3. In the WebSphere Application Server Administration Console, modify the global security settings as follows:
  - a. From the Console menu, select **Security Center**.
  - b. On the General tab, select the **Enable Security** checkbox.
4. Select the **Authentication** tab and select the **Local Operating System** radio button
5. Enter your security server ID in the **Security Server ID** field. Enter the user name you as follows:

Field Name	Sample Values	Notes
User ID	<i>user_ID</i>	<p> The user ID with operating system administrative privileges that you logged in with. If the machine belongs to a domain, use the fully-qualified user ID. For example: DomainXYZ\user_id. Ensure that this account exists in the domain server and is a member of the Administrator's group.</p> <p>   A user ID that is root or has root authority.</p> <p> The userid on iSeries should have the *SEC0FR authority.</p>
Security Server Password	<i>password</i>	This is the password belonging to the user with operating system administrative privileges that you logged in with.

6.   Restart the WebSphere Application Server Administration Server, and then reopen the WebSphere Application Server Administrator's Console.
7. On the **Role Mapping** tab, select the WC enterprise application and click the **Edit Mappings...** button.
  - a. Select the WCSecurityRole and click the **Select...** button.
  - b. Select the Select users/groups check box, enter the user ID that was used in step 5 into the Search field, and click **Search**. Select that user from the Available Users/Groups list and click **Add** to add it to the Selected Users/Groups list. Then click **OK** on each panel until you exit the Security Center.
8. Open the WebSphere Commerce Configuration Manager and select **Instances List** → *instance\_name* → **Instance Properties** → **Security** and select the **Enable Security** check box. Select **Operating System User Registry** for the authentication mode, and to enter the user name and password that you entered in step 5. Click **Apply** then exit Configuration Manager.

9. Stop and restart the WebSphere Application Server administration server. From now on, when you open the WebSphere Application Server Administration Console, you will be prompted for the Security Server ID and password.

---

## Disabling WebSphere Commerce EJB security

WebSphere Commerce Business Edition allows you to disable EJB security. To disable WebSphere Commerce EJB Security, do the following:

1. Start the WebSphere Application Server Administration Console.
2. Click **Console** → **Security Center...** and deselect the **Enable Security** check box on the **General** tab.
3. Open the WebSphere Commerce Configuration Manager and select **Instances List** → *instance\_name* → **Instance Properties** → **Security** and clear the **Enable Security** check box.
4. Exit the WebSphere Application Server Administration Console.
5. Stop and restart the WebSphere Application Server administration server.

---

## WebSphere Commerce security deployment options

WebSphere Commerce supports various security deployment configurations. The following table illustrates the security deployment options available to you.

*Table 12. Single machine security scenarios*

WebSphere Application Server security is enabled.	<ul style="list-style-type: none"> <li>• Use the operating system as the WebSphere Application Server registry.</li> <li>• Use the database as the WebSphere Commerce registry.</li> </ul>
WebSphere Application Server security is disabled, and your WebSphere Commerce site is located behind a firewall.	<ul style="list-style-type: none"> <li>• Use LDAP as the WebSphere Application Server registry.</li> <li>• Use LDAP as the WebSphere Commerce registry.</li> </ul>
	<ul style="list-style-type: none"> <li>• Use LDAP as the WebSphere Application Server registry.</li> </ul>
	<ul style="list-style-type: none"> <li>• A WebSphere Application Server registry is not required.</li> <li>• Use the database as the WebSphere Commerce registry.</li> </ul>
	<ul style="list-style-type: none"> <li>• A WebSphere Application Server registry is not required.</li> <li>• Use LDAP the WebSphere Commerce registry.</li> </ul>

Table 13. Multiple machine security scenarios

<p>WebSphere Application Server security is enabled. LDAP is always deployed.</p>	<ul style="list-style-type: none"> <li>• Use LDAP as the WebSphere Application Server registry.</li> <li>• Use LDAP as the WebSphere Commerce registry.</li> </ul>
<p>WebSphere Application Server security is disabled, and your WebSphere Commerce site is located behind a firewall.</p>	<ul style="list-style-type: none"> <li>• Use LDAP as the WebSphere Application Server registry.</li> <li>• Use a database as the WebSphere Commerce registry.</li> <li>• You will need to set up LDAP, and place one administrative entry into the LDAP registry.</li> </ul>
	<ul style="list-style-type: none"> <li>• Use a database as the WebSphere Commerce registry.</li> <li>• A WebSphere Application Server registry is not required.</li> <li>• Single sign-on is not supported.</li> </ul>
	<ul style="list-style-type: none"> <li>• Use LDAP as the WebSphere Application Server registry.</li> <li>• A WebSphere Application Server registry is not required.</li> </ul>

**Note:** If you operate your WebSphere Commerce site from behind a firewall, you can disable WebSphere Application Server security. You should only disable WebSphere Application Server security if you are sure that no malicious applications are running behind the firewall.

---

## Chapter 6. Session management

Web browsers and e-commerce sites use HTTP to communicate. Since HTTP is a stateless protocol (meaning that each command is executed independently without any knowledge of the commands that came before it), there must be a way to manage sessions between the browser side and the server side.

WebSphere Commerce supports two types of session management: cookie-based and URL rewriting. The administrator can choose to support either only cookie-based session management or both cookie-based and URL rewriting session management. If WebSphere Commerce only supports cookie-based, shoppers' browsers must be able to accept cookies. If both cookie-based and URL rewriting are selected, WebSphere Commerce first attempts to use cookies to manage sessions; if the shopper's browser is set to not accept cookies then URL rewriting is used.

---

### Cookie based session management

When cookie-based session management is used, a message (cookie) containing user's information is sent to the browser by the Web server. This cookie is sent back to the server when the user tries to access certain pages. By sending back the cookie, the server is able to identify the user and retrieves the user's session from the session database; thus, maintaining the user's session. A cookie-based session ends when the user logs off or closes the browser. Cookie-based session management is secure and has performance benefits. Cookie-based session management is secure because it uses an identification tag that only flows over SSL. Cookie-based session management offers significant performance benefits because the WebSphere Commerce caching mechanism only supports cookie-based sessions, and not URL rewriting. Cookie-based session management is recommended for shopper sessions.

If you are not using URL rewriting and you want to ensure that users have cookies enabled on their browsers, check **Cookie acceptance test** on the Session Management page of Configuration Manager. This informs the shopper that if their browser does not support cookies, or if they have turned off cookies, they need a browser that supports cookies to browse the WebSphere Commerce site.

For security reasons, cookie-based session management uses two types of cookies:

- Non-secure session cookie
  - Used to manage session data. Contains the session ID, negotiated language, current store and the shoppers preferred currency when the cookie is constructed. This cookie can flow between the browser and server under either SSL or non-SSL connection. There are two types of non-secure session cookies:
    - A WebSphere Application Server session cookie is based on the servlet HTTP session standard. WebSphere Application Server cookies persist to memory or to the database in a multinode deployment. For more information, search for "session management" in the WebSphere Application Server InfoCenter available at <http://www.ibm.com/software/webservers/appserv/infocenter.html>.
    - A WebSphere Commerce session cookie is internal to WebSphere Commerce and does not persist to the database.

To select which type of cookie to use select WCS or WAS for the **Cookie session manager** parameter on the Session Management page of Configuration Manager.

- Secure authentication cookie

Used to manage authentication data. An authentication cookie flow over SSL and is time stamped for maximum security. This is the cookie used to authenticate the user; whenever a sensitive command is executed, for example, the DoPaymentCmd which asks for a users credit card number. There is minimal risk that this cookie could be stolen and used by an unauthorized user. Authentication code cookies are always generated by WebSphere Commerce whenever cookie based session management is in use.

Both the session and authentication code cookies are required to view secure pages.

For cookie errors the CookieErrorView is called under the following circumstances:

- The user has logged in from another location with the same Logon ID.
- The cookie became corrupted, or was tampered with or both.
- If cookie acceptance is set to "true" and the user's browser does not support cookies.

## Using cookies for session management

To use cookies in WebSphere Commerce, do the following:

1. Open Configuration Manager.
2. Select the **Instance**, then open the **Session Management** folder.
3. Select the appropriate session values.
  - Cookie acceptance test  
Select this check box to check if the customer's browser accepts cookies for a site that only supports cookies.
  - Cookie session manager  
Select whether you want WebSphere Commerce or WebSphere Application Server to manage your cookies. The default is WebSphere Commerce.
    - A WebSphere Application Server session cookie is based on the servlet HTTP session standard. WebSphere Application Server cookies persist to memory or to the database in a multinode deployment. For more information, search for "session management" in the WebSphere Application Server InfoCenter available at <http://www.ibm.com/software/webservers/appserv/infocenter.html>.
    - A WebSphere Commerce session cookie is internal to WebSphere Commerce and does not persist to the database.
4. Click the **Advanced** Tab. Select the appropriate session values.
  - Cookie path  
Usually, this field should not be altered. Specifies the path for the cookie, which is the subset of URLs to which a cookie should be sent.
  - Cookie age  
This field should not be altered. The default is for a cookie to expire when the browser is closed.
  - Cookie domain  
Usually, this field should not be altered. Specifies a domain restriction pattern. A domain specifies the servers that should see a cookie. By default the cookie is only sent back to the WebSphere Commerce Server that issued



them. By default, cookies are returned only to the host that saved them. Specifying a domain name pattern overrides this. The pattern must begin with a dot and must contain at least two dots. A pattern matches only one entry beyond the initial dot. For example, ".ibm.com" is valid and matches a.ibm.com and b.ibm.com but not www.a.ibm.com. For details on domain patterns, see Netscape's Cookie Specification and RFC 2109.

5. Click **Apply**.
6. Close Configuration Manager.
7. From the WebSphere Application Server Administration Console, stop then restart the instance.

---

## URL rewriting

With URL rewriting, all links that are returned to the browser or that get redirected have the session ID appended to them. When the user clicks these links, the rewritten form of the URL is sent to the server as part of the client's request. The servlet engine recognizes the session ID in the URL and saves it for obtaining the proper object for this user. To use URL rewriting, HTML files (files with .html or .htm extensions) cannot be used for links. To use URL rewriting, JSP files must be used for display purposes. A session with URL rewriting expires when the shopper logs off.

**Note:** WebSphere Commerce caching and URL rewriting cannot interoperate. With URL rewriting turned on, you need to disable the WebSphere Commerce caching component.

### Using URL rewriting session management

To specify how sessions should be managed, do the following:

1. Open Configuration Manager.
2. Select the **Instance**, then open the **Session Management** folder.
3. Select the appropriate session values.  
Enable URL rewriting. Select this check box to use URL rewriting for session management.  
Cookie session manager. Select WebSphere Application Server.
4. Click **Apply**.
5. Close Configuration Manager.
6. From the WebSphere Application Server Administration Console, stop then restart the instance.

### Writing JSP templates for URL rewriting

If you want to use URL rewriting to maintain session state, do not include links to parts of your Web application in plain HTML files. This restriction is necessary because URL encoding cannot be used in plain HTML files. To maintain state using URL rewriting, every page that the user requests during the session must have code that can be understood by the Java interpreter. If you have such plain HTML files in your Web application and portions of the site that the user might access during the session, convert them to JSP files. This will impact the application writer because, unlike maintaining sessions with cookies, maintaining sessions with URL rewriting requires that each JSP template in the application must use URL encoding for every HREF attribute on <A> tags. Session will be lost if one or more JSP templates in an application do not call the `encodeURL(String url)` or `encodeRedirectURL(String url)` methods.

## Writing links

With URL rewriting, all links that you return to the browser or redirect must have the session ID appended to them. For example, this link in a Web page:

```
<a href="store/catalog">
```

is rewritten as

```
<a href="store/catalog;$jsessionid$DA32242SSGE2">
```

When the user clicks this link, the rewritten form of the URL is sent to the server as part of the client's request. The Servlet Engine recognizes `;$jsessionid$DA32242SSGE2` as the session ID and saves it for obtaining the proper `HttpSession` object for this user.

The following example shows how Java code may be embedded within a JSP file:

```
<%  
    response.encodeURL ("/store/catalog");  
%>
```

To rewrite the URLs you are returning to the browser, call the `encodeURL()` method in your JSP template before sending the URL to the output stream. For example, if a JSP template that does not use URL rewriting has:

```
out.println("<a href=\"/store/catalog\">catalog</a>")"
```

replace it with:

```
out.println("<a href=\"");  
out.println(response.encodeURL ("/store/catalog"));  
out.println("\>catalog</a>");
```

To rewrite the URLs you are redirecting, call the `encodeRedirectURL()` method. For example, if your JSP template has:

```
response.sendRedirect (response.encodeRedirectURL ("http://myhost/store/catalog"));
```

The `encodeURL()` and `encodeRedirectURL()` methods are part of the `HttpServletResponse` object. In both cases, these calls check to see if URL rewriting is configured before encoding the URL. If it is not configured, it returns the original URL.

**Writing Forms:** To write forms for submission, call the `response.encodeURL("Logon")`; on the `ACTION` tag of the form template. For example,

```
String strLoginPost = response.encodeURL("Logon");  
<FORM NAME="Logon" METHOD="post" ACTION= <%= strLoginPost %> >  
...  
</FORM>
```

**Writing the first page:** The entry page, usually the home page, cannot contain frames. If you want to use frames in your store, you can have a non-frame page with a link to the store act as the store's entry page. However, if the store does use frames and a customer tries to access those pages with frames without going through the entry page first, their session may be lost. Customers can also lose their session if they use the **Back** button (only with frames) to return to the entry page and refresh the entry page. Refreshing the entry page gives them a new session ID. A link back to the entry page as an alternative to the **Back** button is necessary to help prevent this type of session loss.

---

## **Part 3. System administrator security tasks**

This part describes the security tasks that can typically be performed by a system administrator at your site, not necessarily the WebSphere Commerce site administrator.



---

## Chapter 7. Setting and changing passwords

Most components in WebSphere Commerce utilize user IDs and passwords that are validated by the operating system. For information on changing those passwords, refer to your operating system documentation. This chapter covers how to set and change passwords for WebSphere Commerce components that do not validate user IDs and passwords through the operating system.

---

### Quick reference to user IDs, passwords and Web addresses

Administration in the WebSphere Commerce environment requires a variety of user IDs. These user IDs along with their requisite authorities are described in the list below. For the WebSphere Commerce user IDs, the default passwords are identified.

#### Windows User ID

Your Windows user ID *must* have Administrator authority. If you are using DB2®, it requires that the user ID and password adhere to the following rules:

- They cannot be more than 8 characters in length.
- They can contain only the characters A to Z, a to z, 0 to 9, @, #, \$, and \_.
- They cannot begin with an underscore (\_).
- The user ID cannot be any of the following, in upper, lower, or mixed case: USERS, ADMINS, GUESTS, PUBLIC, LOCAL.
- The user ID cannot begin with any of the following in upper, lower, or mixed case: IBM, SQL, SYS.
- The user ID cannot be the same as any Windows service name.
- The user ID must be defined on the local machine, and belong to the Local Administrator's group.
- The user ID must have the *Act as part of the operating system* advanced user right.



You can perform the installation without the *Act as part of the operating system* advanced user right, however, the DB2 setup program will be unable to validate the account that you specify for the Administration Server. It is recommended that any user account used to install DB2 has this advanced user right.

#### Important

If your Windows user ID does *not* have Administrator authority, is more than 8 characters in length, or is not defined on the local machine, you will be notified of the problem and will not be able to proceed with the installation.

If you are using DB2, you will use this user ID as the DB2 database user name (database user logon ID).



If you need to create a user ID fitting the above criteria, you can find information on creating a Windows user ID in the Windows online help.

### iSeries user profiles 400

Two iSeries user profiles are used and referred to frequently when you install and configure WebSphere Commerce:

- A user profile which you create and use to install WebSphere Commerce and access the Configuration Manager. To install and configure WebSphere Commerce, you must use an iSeries user profile of USRCLS(\*SECOFR) or use the QSECOFR user profile. If you need to create a user profile, refer to the *WebSphere Commerce 5.4 Installation Guide* for iSeries.
- A user profile which is created by the Configuration Manager when you create a WebSphere Commerce instance. This user profile is also referred to as the "instance user profile." A user profile of USRCLS(\*USER) is created by the Configuration Manager each time you create a WebSphere Commerce instance. If you need to create a user profile, refer to the *WebSphere Commerce 5.4 Installation Guide* for iSeries.

### Configuration Manager user ID

The Configuration Manager tool's graphical interface allows you to modify the way WebSphere Commerce is configured. The default Configuration Manager user ID and password are webadmin and webibm.

Windows AIX Solaris Linux You can access Configuration Manager from your WebSphere Commerce machine, or any machine on the same network as WebSphere Commerce.

400 For iSeries, you can access Configuration Manager from any Windows machine that is on the same network as your iSeries server.

### IBM HTTP Server User ID Windows AIX Solaris Linux

If you are using IBM HTTP Server, you can access your Web server home page by opening your Web browser and typing the following Web address:  
`http://host_name`

If you have customized your Web server, you may be required to type the name of your Web server's front page after the host name.

### WebSphere Commerce Instance Administrator

The Instance Administrator user ID and password apply to the following WebSphere Commerce tools:

- WebSphere Commerce Accelerator. To access the WebSphere Commerce Accelerator from a remote machine running a Windows operating system, open your Internet Explorer Web browser, and type the following Web address:  
`https://host_name:8000/accelerator`
- WebSphere Commerce Administration Console. To access the WebSphere Commerce Administration Console from a remote machine running a Windows operating system, open your Internet Explorer Web browser, and type the following Web address:  
`https://host_name:8000/adminconsole`
- Store Services. You can access your Store Services page by opening your Web browser and typing the following Web address:  
`https://host_name:8000/storeservices`

The default Instance Administrator user ID is `wcsadmin` and the default password is `wcsadmin`.

**Note:** The `wcsadmin` user ID should never be removed, and should always have instance administrator authority.

WebSphere Commerce requires that the user ID and password adhere to the following rules:

- The password must be at least 8 characters in length.
- The password must include at least 1 numeric digit.
- The password does not contain more than 4 occurrences of a character.
- The password does not repeat the same character more than 3 times.

### Payment Manager Administrator

When you install Payment Manager , the WebSphere Commerce Administrator ID, `wcsadmin`, is automatically assigned the Payment Manager Administrator role. Follow the instructions in the *WebSphere Commerce 5.4 Installation Guide* to switch the Payment Manager Realm Class to `WCSRealm` if it has not already been done.

The Payment Manager Administrator role enables a user ID to control and administer Payment Manager .

#### Notes: 400

- Do not delete or rename the logon user ID `wcsadmin`, and do not change the preassigned Payment Manager role of `wcsadmin` as WebSphere Commerce functions related to Payment Manager integration will not work.
- If you assign a Payment Manager role to a WebSphere Commerce administrator and then later want to delete or rename the logon user ID of this administrator, you must remove the administrator's Payment Manager role before deleting or renaming the user ID.

### Important

Payment Manager has preassigned the Payment Manager Administrator role to two other administration IDs:

- nadmin
- admin


To prevent a user from inadvertently obtaining this Payment Manager Administrator role, you can:

1. Create the above administration IDs in WebSphere Commerce using the WebSphere Commerce Administration Console.
2. On the Payment Manager user interface, select **Users**.
3. Remove the Payment Manager Administrator role from these two administration IDs.

You should also be aware of the Payment Manager Instance Password, which is needed to start, stop, or delete a Payment Manager instance. It is also required to add cassettes to a Payment Manager instance. If a Payment Manager instance is created by the WebSphere Commerce Configuration Manager, the Payment Manager instance password is the same as the WebSphere Commerce instance logon password, which is also referred to as instance user profile password. If a Payment Manager instance is created from an iSeries session using the **CRTPYMMGR** command, or from the iSeries Task Page, you will be prompted to provide the password.

## Changing the Configuration Manager Password

You can change the Configuration Manager password when you launch the Configuration Manager by clicking **Modify** in the window where you enter your user ID and password.

 Alternately, to change the Configuration Manager user ID or password switch to the bin subdirectory under the WebSphere Commerce installation path and type the following in a command window:

```
config_env
java com.ibm.commerce.config.server.PasswordChecker -action [action type]
  -pwfile [password file] -userid [user ID]
  -password [userid password] [-newpassword [new userid password]]
```

where action types are Add, Check, Delete or Modify. The parameters are explained below:

### pwfile

The path to the file where the password will be stored. The default path is the bin subdirectory under the WebSphere Commerce installation path. This parameter is always required.

### userid

Enter the user ID that you want to add, check, delete or modify. This parameter is always required.



### password

Enter the password that you want to create, check, delete or modify. This parameter must be used in conjunction with the `userid` parameter. This parameter is always required.

### newpassword

Use this parameter to change the password for a particular user ID. This parameter must be used in conjunction with the `userid` and `password` parameters. This parameter is required when you specify the action type `Modify`.




---

## Setting Your IBM HTTP Server Administrator Password

    To set your IBM HTTP Server administrator password,

1. Switch to the IBM HTTP Server installation directory on your machine.
2. Type the following command:

```
 htpasswd -b conf\admin.passwd user password
```


```
   htpasswd -b conf/admin.passwd user passwordwhere user and password are the user ID and password that you want to have administrative authority for IBM HTTP Server.
```

You have now successfully set your IBM HTTP Server administration password.

---

## Changing Your SSL Key File Password

    If you are using IBM HTTP Server, follow the steps below to change your SSL key file password.

1.  Click **Start Menu** → **Programs** → **IBM HTTP Server** → **Key Management Utility**.
2. From the **Key Database File** menu, select **Open**.
3. Switch to the `ssl` subdirectory under the IBM HTTP Server installation path on your machine. Your key file (which has the file extension `.kdb`) should be in this folder. If not, create a new key file by following the instructions outlined in Chapter 8, “Enabling SSL for production with IBM HTTP Server,” on page 69.
4. From the **Key Database File** menu, select **Change Password**. The Change Password window appears.
5. Enter your new password, and enable **Stash the password to a file**.
6. Click **OK**. Your password has been changed.

You have now successfully changed your SSL key file administration password.




---

## Generating WebSphere Commerce encrypted passwords

    WebSphere Commerce allows you to generate encrypted passwords. To generate encrypted passwords, do the following:

1. Go to the `bin` subdirectory under the WebSphere Commerce installation directory.
2. Run the following script from a command line:

```
 wcs_password.bat password SALT merchant_key
```

```
   ./wcs_password.sh password SALT merchant_keywhere
```

- *password* is the plain text password.

- *SALT* is a random string that is used in the generation of a password. This is found in the *SALT* column of the *USERREG* database table for the particular user whose password is being updated.
- *merchant\_key* is the merchant key that was entered during instance creation.

**400** For iSeries, to change the encrypted password for shoppers, use the *CHGWCSPWD* command. See the F1 online help for the details of running this command.

---

## Generating Payment Manager encrypted passwords

WebSphere Commerce allows you to generate encrypted passwords for Payment Manager. To generate encrypted passwords, do the following:

1. Go to the *bin* subdirectory under the WebSphere Commerce installation directory.
2. Run the following script from a command line:

```
Windows wcs_pmpassword.bat password SALT
AIX Solaris Linux ./wcs_pmpassword.sh password SALT
```

where:

- *password* is the plain text password.
- *SALT* is a random string that is used in the generation of a password. This is found in the *SALT* column of the *USERREG* database table for the particular user whose password is being updated.

**400** For iSeries, to generate encrypted password for Payment Manager, use the *CRTWCSPMPW* command. See the F1 online help for the details of running this command.

---

## Chapter 8. Enabling SSL for production with IBM HTTP Server

**400** This section does not apply to the iSeries platform. For iSeries information, see “Enabling SSL on the IBM HTTP Server (iSeries)” on page 73.

After you create your WebSphere Commerce instance with IBM HTTP Server, Secure Sockets Layer (SSL) is enabled for testing purposes. Before you open your site to shoppers, you must enable SSL for production by following the steps in this chapter.

---

### About security

IBM HTTP Server provides a secure environment for your business transactions by using encryption technology. Encryption is the scrambling of information transactions on the Internet so that they cannot be read until they are unscrambled by the receiver. The sender uses an algorithmic pattern or key to scramble (encrypt) a transaction, and the receiver uses a decryption key. These keys are used by the Secure Sockets Layer (SSL) protocol.

Your Web server uses an authentication process to verify the identity of the person with whom you are conducting business (that is, to make sure they are whom they say they are). This involves obtaining a certificate signed by a trusted third party called a certification authority (CA). For IBM HTTP Server users, the CA may be Equifax<sup>®</sup> or VeriSign<sup>®</sup> Inc. Other CAs are available as well.

To create a production key file, complete the following steps:

1. Configure a security key file for production.
2. Request a secure certificate from a certifying authority.
3. Set your production key file as the current key file.
4. Receive the certificate and test the production key file.

These steps are described in detail below.

#### Notes:

1. If you are already using a production key file signed by a certifying authority, you may be able to skip these steps. Read this chapter to make this determination.
2. As you perform these steps, your browser may display security messages. Review the information in each message carefully and decide how to proceed.

---

### Configuring a security key file for production

To configure a security key file for production, do the following on your Web server machine:

1. Stop the IBM HTTP Server.
2. Change your directory to the conf subdirectory under the IBM HTTP Server installation directory on your machine.
3. Create a backup copy of httpd.conf.
4. Open httpd.conf in a text editor.
5. Ensure that the following lines are uncommented for port 443:

- **Windows**
    - a. #LoadModule ibm\_ssl\_module modules/IBModuleSSL128.dll
    - b. #Listen 443
    - c. #<VirtualHost host.some\_domain.com:443> (You must also substitute your fully qualified host name in this line.)
    - d. #SSLEnable
    - e. #</VirtualHost>
    - f. Keyfile "drive:/WebSphere/HTTPServer/ssl/keyfile.kdb"
  - **AIX Solaris Linux**
    - a. **AIX Solaris** #LoadModule ibm\_ssl\_module libexec/mod\_ibm\_ssl\_128.so  
**Linux** #LoadModule ibm\_ssl\_module
    - b. #AddModule mod\_ibm\_ssl.c
    - c. #Listen 443
    - d. #<VirtualHost host.some\_domain.com:443> (You must also substitute your fully qualified host name in this line.)
    - e. #SSLEnable
    - f. #</VirtualHost>
    - g. #SSLDisable
    - h. Keyfile "keyfile"  
where *keyfile* is one of:
      - **AIX** /usr/HTTPServer/ssl/keyfile.kdb
      - **Solaris** /opt/IBMHTTPD/ssl/keyfile.kdb
      - **Linux** /opt/IBMHTTPServer/ssl/keyfile.kdb
    - i. #SSLV2Timeout 100
    - j. #SSLV3Timeout 1000
6. Ensure that the following lines are uncommented for port 8000:
    - a. #Listen 8000
    - b. #<VirtualHost host.some\_domain.com:8000>. You must also substitute your fully qualified host name in this line.
    - c. #SSLEnable
    - d. #</VirtualHost>

**Note:** It is recommended that your firewall software blocks external access to the port you have configured for WebSphere Commerce Tools (port 8000 by default). Consult the documentation for the firewall software you are using at your site for information on how you do this.

7. Save your changes.
8. To ensure that your httpd.conf file does not contain syntax errors:
  - **Windows** Change to the IBM HTTP Server installation directory on your machine and run the following command:  
apache -t
  - **AIX Solaris Linux** Change to the bin subdirectory under the IBM HTTP Server installation directory on your machine and run the following command:  
./httpd -t
9. Start the IBM HTTP Server.

---

## Request a secure certificate from a certifying authority

To validate the security key file that you just created in the previous step, you need a certificate from a certifying authority (CA) such as Equifax or VeriSign. The certificate contains the server's public key, the Distinguished Name associated with the server's certificate, and the serial number and expiration date of the certificate.

If you want to use a different CA, contact it directly for information on the procedure to follow.

### Equifax users

To request a secure server certificate from Equifax, refer to the following Web address and follow the instructions provided:

<http://www.equifax.com>

You should receive the secure server certificate through E-mail from Equifax in 2 to 4 business days.

### VeriSign users

To request a secure server certificate from VeriSign, refer to the following URL and follow the instructions provided:

<http://www.verisign.com>

**AIX** Although you are using the procedures for IBM HTTP Server, follow the link for **Internet Connection Secure Server (ICSS)**. Follow the instructions provided. When you receive your certificate, create the production key file as described in the previous section, if you have not already done so.

**Solaris** Even though you are using the procedures for IBM HTTP Server, follow the link for **Internet Connection Secure Server (ICSS)**. The subsequent page indicates that the procedures apply to the OS/2<sup>®</sup> and AIX platforms. These instructions also apply for Solaris software.

Follow the instructions provided. Once you submit your request, your certificate should arrive within three to five working days. When you receive it, create the production key file as described in the previous section, if you have not already done so.

---

## Receive and set your production key file as the current key file

After the certificate arrives from the CA, you must make the Web server use your production key file. Perform the following steps:

1. Copy the *certificatename.kdb*, *certificatename.rdb*, and *certificatename.sth* files you received from the certificate authority into the `ssl` subdirectory under the IBM HTTP Server installation path on your machine, where *certificatename* is the certificate name you supplied with your certificate request.

2. Stop IBM HTTP Server.

3. **AIX** **Solaris** Export `JAVA_HOME` by running the following commands:

```
DISPLAY=host_name:0.0
export DISPLAY
JAVA_HOME=java_home
export JAVA_HOME
```

where *host\_name* is the fully qualified host name of the machine you are currently using and *java\_home* is:

- **AIX** /usr/java130
- **Solaris** /opt/WebSphere/AppServer/java131

4. Open the Key Management Utility (ikeyman).
5. Open the *certificatename.kdb* file, and enter your password when prompted.
6. Select **Personal Certificates**, and click **Receive**.
7. Click **Browse**.
8. Select the folder where you have stored the files you received from the certificate authority. Select the *certificatename.txt* file and click **OK**.
9. The **Personal Certificates** list box should now list either VeriSign *certificatename* certificate or Equifax *certificatename* certificate.
10. Exit the Key Management Utility.
11. Change directory to the conf subdirectory under the IBM HTTP Server installation path on your machine.
12. Create a backup copy of httpd.conf.
13. Open httpd.conf in a text editor.
14. Ensure that the lines listed in step 5 on page 69 are not commented.
15. Search for Keyfile "*keyfile\_path\_name/keyfile.kdb*" directive, and change the path name to point to the file created in the above steps.
16. Restart the IBM HTTP Server.

---

## Test the production key file

To test the production key, do the following:

1. Go the following URL with your browser:

`https://host_name`

**Notes:**

- a. If you have customized your Web server, you may need to type the name of the Web server's front page after the host name.
- b. Be sure to type https, *not* http.

If your key is defined correctly, you will see several messages about your new certificate.

2. On the **New Site Certificate** panel, if you want to accept this certificate, select the **Accept this certificate forever (until it expires)** radio button.
3. From your Web browser, restore your caching and proxy (or socks) server settings to their original states.

You have now enabled SSL on your server.

---

## SSL Consideration for Payment Manager





By default, the communication between WebSphere Commerce and Payment Manager is through SSL. However, if you launch the Payment Manager user interface directly as follows:

`http://host_name/webapp/PaymentManager/PaymentServerUI/Start`

then you are invoking Payment Manager using non-SSL communication. To ensure that the communication is through SSL, you should either use

`https://host_name/webapp/PaymentManager/PaymentServerUI/Start`

or rename the indexSSL.html file to index.html in the following directory:

-  `WAS_HOME\installedApps\IBM_PaymentManager.ear\PaymentManager.war`
-    `WAS_HOME/installedApps/IBM_PaymentManager.ear/PaymentManager.war`

This way, you can continue to use the `http://host_name/webapp/PaymentManager/` directory, and the renamed index.html will redirect to https (SSL).

---

## Enabling SSL on the IBM HTTP Server (iSeries)

 This section applies to the iSeries platform.

SSL is a security protocol. SSL ensures that data transferred between a client and a server remains private. It allows the client to authenticate the identity of the server and the server to authenticate the identity of the client.

Digital certificates are electronic documents that authenticate the servers and clients involved in secured transactions over the Internet. The issuer of digital certificates is called a certificate authority (CA). The iSeries system can perform the role of CA in an Intranet environment issuing server and client certificates, and run as an authenticated server with server certificates issued either by an iSeries CA or an Internet CA like VeriSign®. As a Web server, the IBM HTTP Server for iSeries can also be configured to request client certificates for authentication of SSL-enabled clients.

For detailed information on how to enable SSL on the IBM HTTP Server for iSeries, refer to the iSeries Information Center at the following Web address:  
<http://publib.boulder.ibm.com/html/as400/infocenter.html>

Once you are at the site, select your operating system version and your language, and then click **Go**. Search for the topic "Securing applications with SSL" for guidance on how to enable SSL.

## Using SSL with Payment Manager

If you create the system certificate store after creating your WebSphere Commerce instance, you must grant both the Payment Manager instance and the WebSphere Commerce instance access to the system certificate store. For example, the following commands will grant the Payment Manager instance the required access on a V5R1 system:

```
CHGAUT OBJ('/QIBM/UserData/ICSS/Cert/Server') USER(QPYMSVR) DTAAUT(*RX)
CHGAUT OBJ('/QIBM/UserData/ICSS/Cert/Server/DEFAULT.KDB') USER(QPYMSVR) DTAAUT(*R)
```

and the following commands will grant the WebSphere Commerce the required access on a V5R1 system:

```
CHGAUT OBJ('/QIBM/UserData/ICSS/Cert/Server') USER(QEJBSVR) DTAAUT(*RX)
CHGAUT OBJ('/QIBM/UserData/ICSS/Cert/Server/DEFAULT.KDB') USER(QEJBSVR) DTAAUT(*R)
```

If you choose to use a remote Payment Manager instance, you must configure both the WebSphere Commerce instance and the Payment Manager instance to trust the remote certificate authority that issues the digital certificate. To establish a trust relationship between the two remote applications, refer to the following high-level procedure:

1. On the WebSphere Commerce machine, use the Digital Certificate Manager to export the server's certificate authority.

2. Transfer the certificate file to the Payment Manager machine.
3. On the Payment Manager machine, use the Digital Certificate Manager to import the WebSphere Commerce server's certificate authority.
4. Configure the Payment Manager application server to trust the imported WebSphere Commerce server's certificate authority.
5. On the Payment Manager machine, use the Digital Certificate Manager to export the server's certificate authority.
6. Transfer the certificate file to the WebSphere Commerce machine.
7. On the WebSphere Commerce machine, use the Digital Certificate Manager to import the Payment Manager server's certificate authority.
8. Configure the WebSphere Commerce application server to trust the imported Payment Manager server's certificate authority.

For detailed information refer to the following Web address, and look for **Hints and Tips.**:

<http://www.ibm.com/software/webservers/commerce/servers/lit-tech-os400.html>

---

## + Enabling the SSL Accelerator option

+ When using an SSL Accelerator (also known as an SSL Terminator) with  
 + WebSphere Commerce, you can use the SSL Accelerator option  
 + (SSLAcceleratorOption) to configure WebSphere Commerce to correctly receive  
 + requests that require redirects. An SSL Accelerator (or SSL Terminator) is hardware  
 + that strips off HTTPS (secure sockets layer) encryption at or before the Web server  
 + tier in a multi-tier setup.

+ To enable the SSL Accelerator option, the following attributes have to be added to  
 + the *WC\_instance.xml* file in the WebServer node:

```
+ <WebServer ...
+ :
+ SSLAcceleratorOption="Enabled"
+ InSSLPort="443"
+ InNonSSLPort="80"
+ OutSSLPort="443"
+ OutNonSSLPort="80"
+ :
+ ... >
```

+ Where:

### + InSSLPort

+ The port configured for WebSphere Commerce to receive the SSL data -  
 + WebSphere Commerce will treat this as SSL data even if the scheme says  
 + http. The default is port 443.

### + InNonSSLPort

+ The port configured for WebSphere Commerce to receive the non SSL data  
 + - WebSphere Commerce will treat any data received in this port as non-SSL  
 + data. The default port is 80.

### + OutSSLPort

+ The port that WebSphere Commerce will use to send out SSL data on a  
 + redirect. The default port is 443.

### + OutNonSSLPort

+ The port that WebSphere Commerce will use to send out non-SSL data on  
 + a redirect. The default port is 80.



---

## Chapter 9. Enabling SSL for IBM SecureWay Directory Server (LDAP)

The following are the steps to configure SSL security for IBM SecureWay Directory Server and WebSphere Commerce.

---

### Set up SecureWay

To set up the IBM SecureWay Directory Server:

1. Install IBM SecureWay Directory Server according to the SecureWay Directory Server product installation instructions. Ensure that you install the GSKit component.
2. After the installation completes, invoke the IBM Key Manager (*drive:\Program Files\IBM\GSK4\bin\gsk4ikm.exe* on Windows).
3. Create a new CMS Key database file. Make sure the **stash password to file** is selected (for example, *ldap\_key.kdb*)
4. Create a self-signed certificate
5. Extract the certificate as Base64-encoded ASCII data type.
6. Create a new SSLight key database class (for example, *keyring.class*).
7. In the **Singer Certificates** section, add the certificate file created in step 5.
8. Open a browser to the following address: <http://hostname/ldap>
9. Click **Security** → **SSL** → **Settings** and make the following changes:
  - SSL status: SSL on or SSL only
  - Authentication method: Server Authentication
  - Secure port: 636
  - Key database path and file name:
    - AIX Solaris Linux `/Keys/ldap_key.kdb`
    - Windows `drive:\Keys\ldap_key.kdb`
  - Key label: *your\_label* (The label of the certificate)
10. Click **Update** and restart SecureWay.

---

### WebSphere Commerce

To set up WebSphere Commerce to work with SecureWay Directory Server you need to modify the *instance.xml* file:

```
java.naming.security.ssl.keyring = keyring
'keyring' is the name of the SSLight key database class (keyring.class)
This class file should put in the class path in WAS.
```

```
java.naming.security.ssl.authentication = ibm
'ibm' is the password specified when create the SSLight key database class.
```

```
java.naming.security.protocol = ssl
LdapPort = 636
<MemberSubSystem name="Member SubSystem"
    ProfileDataStorage="LDAP"
    AuthenticationMode="LDAP">
  <Directory LdapAdminDN="cn=root"
    LdapAuthenticationMode="SIMPLE"
    LdapTimeout="0"
```

```
LdapVersion="3"  
EntryFileName="WC_Install_Dir/xml/ldap/ldapentry.xml"  
LdapPort="636"  
SingleSignOn="0"  
LdapAdminPW="EaDPFd9VAf0"  
LdapHost="yazhuang.torolab.ibm.com"  
MigrateUsersFromWCSdb="ON"  
JNDIEnvPropName1="java.naming.security.ssl.keyring"  
JNDIEnvPropValue1="keyring"  
JNDIEnvPropName2="java.naming.security.ssl.authentication"  
JNDIEnvPropValue2="ibm"  
JNDIEnvPropName3="java.naming.security.protocol"  
JNDIEnvPropValue3="ssl"  
display="false"  
LdapType="SECUREWAY" />  
</Membersubsystem>
```

Restart WebSphere Commerce.

---

## Chapter 10. Single sign-on

This chapter outlines how to set up single sign-on for WebSphere Commerce.

---

### Prerequisites

To enable single sign-on, you must meet the following requirements:

- There must be an existing LDAP server installed and configured. To configure an LDAP server see the *IBM WebSphere Commerce Version 5.4 Additional Software Guide*.
- WebSphere Commerce must be installed and configured to use LDAP.
- WebSphere Application Server security must be enabled. To enable WebSphere Application Server security see Chapter 5, "Enabling WebSphere Application Server security," on page 49.

---

### Enabling single sign-on

#### Limitations

There are several key limitations of single sign-on when it is used with WebSphere Commerce. These limitations are:

- The LTPA cookies may flow across different web server ports.
- You may need to modify the `ldapentry.xml` file and add the object class `ePerson`. That is as an attribute of `ldapocs` element.
- You need to modify the `instance.xml` and ensure that migration is "on" for user in the LDAP component.
- The machines participating in the single sign-on configuration must have their system clocks synchronized.
- Single sign-on is only supported between applications that can read and issue the WebSphere Application Server Light Weight Third Party Authentication (LTPA) token.

To enable single sign-on you must do the following:

1. Enable single sign-on within the WebSphere Application Server. For more information, search for "single sign-on" in the WebSphere Application Server InfoCenter available at:

<http://www.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/index.html>

Select **Single Sign-On: WebSphere Application Server** and complete the following sections:

- **Configuring SSO for WebSphere Application Server.**
  - **Modify WebSphere Application Server security settings.**

**Note:** The step that details how to fill in the LDAP fields can be safely ignored.

- **Export the LTPA keys to a file.**

2. On your WebSphere Commerce machine, start the WebSphere Commerce Configuration Manager.

3. To configure the **Member Subsystem** node, do the following:
  - a. Expand **WebSphere Commerce** → *host\_name* → **Instance List** → *instance\_name* → **Instance Properties** → **Member Subsystem**.
  - b. In the **Authentication Mode** drop-down menu, select **LDAP**.
  - c. Enable the **Single sign-on** checkbox.
  - d. In the **Host** field, enter the fully qualified host name of your LDAP server.
  - e. Enter the administrator's distinguished name in the **Administrator Distinguished Name** field. This should be the same name that was used on your LDAP server.
  - f. In the **Administrator Password** field, enter the administrator's password. This should be the same password that was used on your LDAP server. Confirm the password in the **Confirm Password** field.
  - g. Complete each of the remaining fields.
  - h. Click **Apply**, then click **OK**.
4. Restart the WebSphere Application Server.

---

## **Part 4. WebSphere Commerce developer security tasks**

This part describes the security tasks that have to do with WebSphere Commerce programming. These tasks are typically performed by WebSphere Commerce programmers.



---

## Chapter 11. Access control

---

### Understanding access control

The access control model of a WebSphere Commerce application has three primary concepts: users, actions and resources. Users are the people that use the system. Resources are the entities that are maintained in or by the application. For example, resources may be products, documents, or orders. User profiles that represent people are also resources. Actions are the activities that users can perform on the resources. Access control is the component of the e-commerce application that determines whether a given user can perform a given action on a given resource.

In a WebSphere Commerce application, there are two main levels of access control. The first level of access control is performed by the WebSphere Application Server. In this respect, WebSphere Commerce uses WebSphere Application Server to protect enterprise beans and servlets. The second level of access control is the fine-grained access control system of WebSphere Commerce.

The WebSphere Commerce access control framework uses access control policies to determine if a given user is permitted to perform a given action on a given resource. This access control framework provides fine-grained access control. It works in conjunction with, but does not replace the access control provided by the WebSphere Application Server.

### Overview of resource protection in WebSphere Application Server

The following WebSphere Commerce resources are protected under access control by WebSphere Application Server:

- Entity beans  
These beans model objects in an e-commerce application. They are distributed objects that can be accessed by remote clients.
- JSP templates  
WebSphere Commerce uses JSP templates for display pages. Each JSP template can contain one or more data beans that retrieve data from entity beans. Clients can request JSP pages by composing a URL request.
- Controller and view commands  
Clients can request controller and view commands by composing URL requests. In addition, one display page may contain a link to another by using the JSP file name or the view name, as registered in the VIEWREG table.

The WebSphere Commerce server is typically configured to use the following Web paths:

- `/webapp/wcs/stores/servlet/*`  
This is used for requests to the request servlet.
- `/webapp/wcs/stores/*.jsp`  
This is used for requests to the JSP servlet.

The following diagram shows the route that requests could potentially follow to access WebSphere Commerce resources, for the preceding Web path configuration.

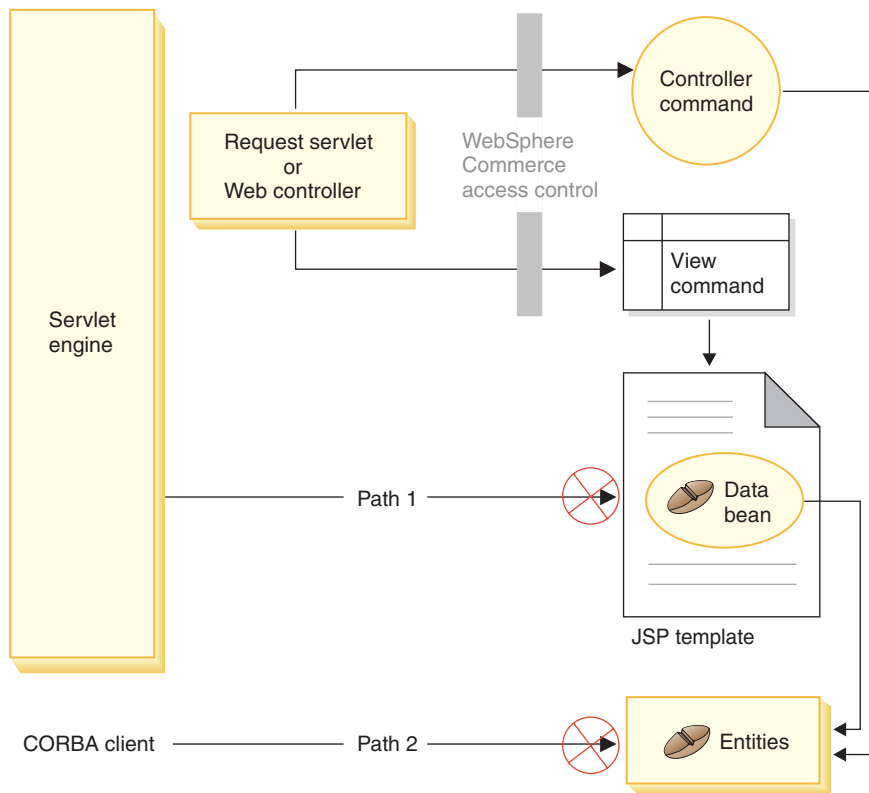


Figure 3.

All the legitimate requests should be directed to the request servlet, which then directs them to the Web controller. The Web controller implements access control for controller commands and views. The Web paths shown above do, however, make it possible for malicious users to directly access JSP templates (path 1) and entity beans (path 2). In order to prevent these malicious attacks from being successful, they must be rejected at run time.

Direct access to the JSP templates and entity beans can be prevented by using one of the following approaches:

#### WebSphere Application Server security

WebSphere Application Server provides a security feature. Using this approach, all enterprise bean methods and JSP templates are configured to be invoked by the System Identity only. To access these WebSphere Commerce resources, a URL request must be routed to the request servlet that sets the System Identity to the current thread, before passing it to the Web controller. The Web controller then ensures that the caller has the required authorization before passing the request to the corresponding controller command or view. Any attempts to directly access JSP templates and entity beans (that is, without using the Web controller) are rejected by the WebSphere Application Server security component.

For information about configuring WebSphere Application Server to secure WebSphere Commerce resources, refer to the *WebSphere Commerce Installation Guide*. For information about security within WebSphere Application Server, refer to the System Administration topic in the WebSphere Application Server documentation.

For information about configuring WebSphere Application Server security for methods in customized enterprise beans, refer to the sections



"Assembling new enterprise beans into an enterprise application" and "Assembling modified enterprise beans into an enterprise application" in the *WebSphere Commerce 5.4 Programmer's Guide*.

### Firewall protection

When a WebSphere Commerce Server runs behind the firewall, Internet clients are not able to directly access the entity beans. Using this approach, protection for JSP templates is provided by the data bean that is included in the page. The data bean is activated by the data bean manager. The data bean manager detects if the JSP template was forwarded by a view command. If it was not forwarded by a view command an exception is thrown and the request for the JSP template is rejected.

## Introduction to WebSphere Commerce access control policies

The WebSphere Commerce access control model is based upon the enforcement of access control policies. Access control policies allow access control rules to be externalized from business logic code, thereby removing the need to hard code access control statements into code. For example, you do not need to include code similar to the following:

```
if (user.isAdministrator())  
    then {}
```

Access control policies are enforced by the access control policy manager. In general, when a user attempts to access a protected resource, the access control policy manager first determines what access control policies are applicable for that protected resource, and then, based upon the applicable access control policies, it determines if the user is allowed to access the requested resources.

An access control policy is a 4-tuple policy that is stored in the ACPOLICY table. Each access control policy takes the following form:

```
AccessControlPolicy [UserGroup, ActionGroup, ResourceGroup, Relationship]
```

The elements in the 4-tuple access control policy specify that a user belonging to a specific user group is permitted to perform actions in the specified action group on resources belonging to the specified resource group, as long as the user satisfies the conditions specified in the relationship or relationship group, with respect to the resource in question. For example, [AllUsers, UpdateDoc, doc, creator] specifies that all users can update a document, if they are the creator of the document.

The user group is a specific type of member group that is defined in the MBRGRP database table. A user group must be associated with member group type of -2. The value of -2 represents an access group and is defined in the MBRGRPTYPE table. The association between the user group and member group type is stored in the MBRGRPUSG table.

The membership of a user into a particular user group may be stated explicitly or implicitly. An explicit specification occurs if the MBRGRPMBR table states that the user belongs to a particular member group. An implicit specification occurs if the user satisfies a condition (for example, all users that fulfill the role of Product Manager) that is stated in the MBRGRPCOND table. There may also be combined conditions (for example, all users that fulfill the role of Product Manager and that have been in the role for at least 6 months) or explicit exclusions.

Most conditions to include a user in a user group are based upon the user fulfilling a particular role. For example, there could be an access control policy that allows all users that fulfill the Product Manager role, to perform catalog

management operations. In this case, any user that has been assigned the Product Manager role in the MBRROLE table is then implicitly included in the user group.

For more details about the member group subsystem, refer to the WebSphere Commerce online help.

The ActionGroup element comes from the ACACTGRP table. An action group refers to an explicitly specified group of actions. The listing of actions is stored in the ACACTION table and the relationship of each action to its action group (or groups) is stored in the ACACTACTGP table. An example of an action group is the "OrderWriteCommands" action group. This action group includes the following actions that are used to update orders:

- com.ibm.commerce.order.commands.OrderDeleteCmd
- com.ibm.commerce.order.commands.OrderCancelCmd
- com.ibm.commerce.order.commands.OrderProfileUpateCmd
- com.ibm.commerce.order.commands.OrderUnlockCmd
- com.ibm.commerce.order.commands.OrderScheduleCmd
- com.ibm.commerce.order.commands.ScheduledOrderCancelCmd
- com.ibm.commerce.order.commands.ScheduledOrderProcessCmd
- com.ibm.commerce.order.commands.OrderItemAddCmd
- com.ibm.commerce.order.commands.OrderItemDeleteCmd
- com.ibm.commerce.order.commands.OrderItemUpdateCmd
- com.ibm.commerce.order.commands.PayResetPMCcmd

A resource group is a mechanism to group together particular types of resources. Membership of a resource in a resource group can be specified in one of two ways:

- Using the conditions column in the ACRESGRP table
- Using the ACRESGPRES table

In most cases, it is sufficient to use the ACRESGPRES table for associating resources to resource groups. Using this method, resources are defined in the ACRESGRY table using their Java class name. Then, these resources are associated with the appropriate resource groups (ACRESGRP table) using the ACRESGPRES association table. In cases where the Java class name alone is not sufficient to define the members of a resource group (for example, if you need to further restrict the objects of this class based on an attribute of the resource), the resource group can be defined entirely using the conditions column of the ACRESGRP table. Note that in order to perform this grouping of resources based on an attribute, the resource must also implement the Groupable interface.

The following diagram shows an example resource grouping specification. In this example resource group 10023 includes all the resources that are associated with it in the ACRESGPRES table. Resource group 10070 is defined using the conditions field column in the ACRESGRP table. This resource group includes instances of the Order remote interface, that also have status = "Z" (specifying a shared requisition list).

**Note:** Details about the XML information for the Conditions column of the ACRESGRP table are found in the *WebSphere Commerce Access Control Guide*.

ACRESGRP

AcResGrp_Id	GrpName	Conditions
10023	AccountRepresentatives CmdResourceGroup	null
10070	SharedRequisitionList ResourceGroup	<pre> &lt;profile&gt; &lt;andListCondition&gt; &lt;simpleCondition&gt; &lt;variable name="Status"/&gt; &lt;operator name="="/&gt; &lt;value data="Z"/&gt; &lt;/simpleCondition&gt; &lt;simpleCondition&gt; &lt;variable name="classname"/&gt; &lt;operator name="="/&gt; &lt;value data="com.ibm.commerce.order. objects.Order"/&gt; &lt;/simpleCondition&gt; &lt;/andListCondition&gt; &lt;/profile&gt; </pre>

ACRESRPES

AcResGrp_Id	AcResCgry_Id
10023	10246
10023	10247
10023	10248
10023	10249
10023	10250

ACRESCGRY

AcResCgry_Id	ResClassname
10246	com.ibm.commerce.contract. commands.ContractCreateCmd
10247	com.ibm.commerce.contract. commands.ContractCreateCmd
10248	com.ibm.commerce.contract. commands.ContractCreateCmd
10249	com.ibm.commerce.contract. commands.ContractCreateCmd
10250	com.ibm.commerce.contract. commands.ContractCreateCmd

Figure 4.



The MEMBER\_ID column of the AACTGRP, ACRESGRP, and ACRELGRP tables should have a value of -2001 (Root Organization).

The access control policy can optionally include either a Relationship or RelationshipGroup element as its fourth element.

If your access control policy uses a Relationship element, this comes from the ACRELATION table. If, on the other hand, it includes a RelationshipGroup element, that comes from the ACRELGRP table. Note that neither need be included, but if you include one, you cannot include the other. A RelationshipGroup specification from the ACRELGRP table takes precedence over the Relationship information from the ACRELATION table.

The ACRELATION table specifies the types of relationships that exist between users and resources. Some examples of types of relationships include creator, submitter, and owner. An example of the use of the relationship element is to use it to ensure that the creator of an order can always update the order.

The ACRELGRP table specifies the types of relationship groups that can be associated with particular resources. A relationship group is a grouping of one or more relationship chains. A relationship chain is a series of one more relationships. An example of a relationship group is to specify that a user must be the creator of the resource and also belong to the buying organizational entity that is referenced in the resource.

The relationship group (or relationship) specification is an optional part of the access control policy. It is commonly used if you have created your own commands and these commands are not restricted to certain roles. In these cases, you might want to enforce a relationship between the user and the resource. Typically, if commands are to be restricted to certain roles, it is accomplished through the UserGroup element of the access control policy rather than by using the Relationship element.

Another important concept related to access control policies is the concept of an access control policy *owner*. An access control policy owner is the organizational entity that owns the access control policy. Knowing the owner of an access control policy is important because an access control policy can only be applied to resources that are owned by the access control policy owner.

For each resource in question, the access control policy manager applies access control policies that are owned by the owning organizational entity or by its ancestor organizational entities in the member hierarchy, until either a policy is found that grants permission or until all policies have been checked and none grant permission.

Consider the following diagram showing a member hierarchy.

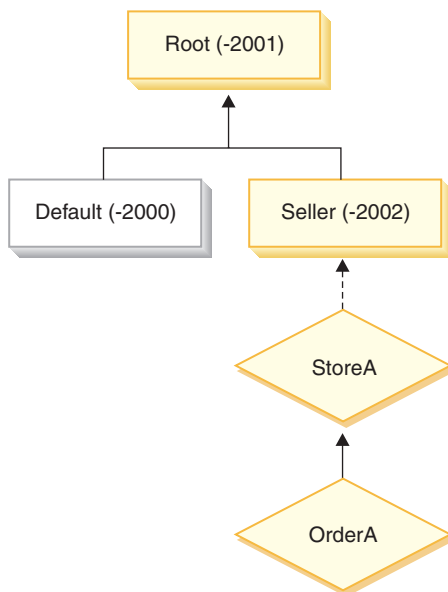


Figure 5.

For the resource “OrderA”, any access control policy that is owned by the Seller or Root organization can be applied. If the access control policy manager finds one policy owned by either of these organizations that grants the user permission (based upon the four elements in the access control policy) it immediately stops searching through the access control policies. However, if it does not find any

access control policies owned by those organizations that grant the user permission to perform the action on the protected resources, then access is denied.

## Relationship groups

A relationship group allows you to specify multiple relationships. A relationship can be directly between a user and the resource in question, or it can be a chain of relationships that indirectly relate the user to the resource.

**Note:** For the following sections related to relationship groups, it is important to recognize that the only organizations available in WebSphere Commerce Professional Edition are the RootOrganization, the DefaultOrganization, and the SellerOrganization. Examples that refer to other organizations only apply to WebSphere Commerce Business Edition.

**Comparing relationships to relationship groups:** Access control policies can specify that a user must fulfill a particular relationship with respect to the resource being accessed, or they can specify that a user must fulfill the conditions specified in a relationship group.

In most cases, specifying a relationship should satisfy the access control requirements for your application. If, however, the policy is such that you must specify a relationship that is not directly between the user and the resource, but that is actually a series of relationships between the user and the resource, you must then use a relationship group.

For example, if you must specify an association between a user and a buying organization where the relationship requires that the user is playing a particular role for that organization or that the user is a member of the buying organization, then you must use a relationship group and a chain of relationships.

If you merely need to enforce an association that is directly between the user and the resource in question, you can use a simple relationship. For example, this would be the case if you need to enforce that the user must be the creator of the resource.

If you combine multiple simple relationships, for example, the user must be the creator *or* the submitter, then this becomes a chain of relationships and you must use a relationship group. This combination of simple relationships may occur when using either WebSphere Commerce Professional Edition or WebSphere Commerce Business Edition.

**General information about relationship groups:** A relationship chain is a series of one more relationships. The length of a relationship chain is determined by the number of relationships that it contains. This can be determined by examining the number of `<parameter name="aName" value="aValue" />` elements in the XML representation of the relationship chain.

Only the last `<parameter name="Relationship" value="aValue" />` element must be handled by the `fulfills()` method of the resource. The rest are handled internally by the access control policy manager.

When a relationship chain has a length of 2, the first `<parameter name="aName" value="aValue" />` element is between a user and an organizational entity. The last `<parameter name="aName" value="aValue" />` element is between an organizational entity and the resource.

If you need to define relationship groups, you must do so by defining the relationship group information in an XML file. You can modify the defaultAccessControlPolicies.xml file, or create your own XML file. For more information about creating these XML-based information, refer to the *WebSphere Commerce Access Control Guide*.

The following sections show examples of different types of relationship groups.

*Relationship groups composed of a single relationship chain:* **Business** As part of an access control policy, you may be required to enforce that a user must belong to the organizational entity that is the BuyingOrganizationalEntity of the resource. This requires the creation of a relationship group that is composed of one relationship chain that has a length two. The relationship chain is said to be of length "two" because it consists of two separate relationships. The first relationship is between the user and its parent organizational entity. The user is the "child" in that relationship. For the second relationship, the access control policy manager checks if the parent organizational entity fulfills the BuyingOrganizationalEntity relationship with the resource. In other words, it returns "true" if it is the buying organizational entity of the resource.

The following XML snip is taken from the defaultAccessControlPolicies.xml file and shows how to define this type of relationship group:

```
<RelationGroup Name="MemberOf->BuyerOrganizationalEntity"
  OwnerID="RootOrganization">
  <RelationCondition><![CDATA[
    <profile>
      <openCondition name="RELATIONSHIP_CHAIN">
        <parameter name="HIERARCHY" value="child"/>
        <parameter name="RELATIONSHIP" value="BuyingOrganizationalEntity"/>
      </openCondition>
    </profile>
  ]]></RelationCondition>
</RelationGroup>
```

**Business** Another example would be to enforce that the user must have the role of Account Representative for the organizational entity that is the buying organizational entity of the resource in question. Again, this uses a relationship group that is composed of one relationship chain of length two. The first part of the chain will find all of the organizational entities for which the user has the Account Representative role. Then for this set of organizational entities, the access control policy manager checks if at least one of them fulfills the BuyingOrganizationalEntity relationship with the resource. In other words, it returns true if one of them is the buying organizational entity of the resource.

The following XML snip is taken from the defaultAccessControlPolicies.xml file and shows how to define this type of relationship group:

```
<RelationGroup Name="AccountRep->BuyerOrganizationalEntity"
  OwnerID="RootOrganization">
  <RelationCondition><![CDATA[
    <profile>
      <openCondition name="RELATIONSHIP_CHAIN">
        <parameter name="ROLE" value="Account Representative"/>
        <parameter name="RELATIONSHIP" value="BuyingOrganizationalEntity"/>
      </openCondition>
    </profile>
  ]]></RelationCondition>
</RelationGroup>
```

*Relationship groups composed of multiple relationship chains:* It is possible to compose a relationship group so that it contains multiple relationship chains. When doing so, you must specify whether the user must satisfy all of the relationship chains, meaning it is an *AND* scenario, or the user must satisfy at least one of the relationship chains, meaning it is an *OR* scenario.

**Business** To demonstrate this type of relationship, the following XML snip is used to enforce that a user must be the creator of the resource and the user must also belong to the `BuyingOrganizationalEntity` specified in the resource. The first chain, that specifies the user must be the creator of the resource is of length one. The second chain that specifies that the user must belong to the `BuyingOrganizationalEntity` specified in the resource is of length two.

```
<RelationGroup Name="Creator_And_MemberOf->BuyerOrganizationalEntity"
  OwnerID="RootOrganization">
  <RelationCondition><![CDATA[
  <profile>
    <andListCondition>
      <openCondition name="RELATIONSHIP_CHAIN">
        <parameter name="RELATIONSHIP" value="creator" />
      </openCondition>
      <openCondition name="RELATIONSHIP_CHAIN">
        <parameter name="HIERARCHY" value="child"/>
        <parameter name="RELATIONSHIP" value="BuyingOrganizationalEntity"/>
      </openCondition>
    </andListCondition>
  </profile>
  ]]></RelationCondition>
</RelationGroup>
```

If, instead of the *AND* scenario, you require the user to satisfy either of the two relationship chains, the `<andListCondition>` tag should be changed to the `<orListCondition>` tag.

**Professional Business** To demonstrate a relationship group that can be used in WebSphere Commerce Professional Edition (as well as WebSphere Commerce Business Edition), consider a relationship group that is used to enforce that the user must be either the creator or the submitter of the resource. This is shown in the following XML snip.

```
<RelationGroup Name="Creator_Or_Submitter"
  OwnerID="RootOrganization">
  <RelationCondition><![CDATA [
  <profile>
    <orListCondition>
      <openCondition name="RELATIONSHIP_CHAIN">
        <parameter name="RELATIONSHIP" value="creator"/>
      </openCondition>
      <openCondition name="RELATIONSHIP_CHAIN">
        <parameter name="RELATIONSHIP" value="submitter"/>
      </openCondition>
    </orListCondition>
  </profile>
  ]]></RelationCondition>
</RelationGroup>
```

## Types of access control

There are two types of access control, both of which are policy-based: command-level access control and resource-level access control.

Command-level (also known as “role-based”) access control uses a broad type of policy. You can specify that all users of a particular role can execute certain types

of commands. For example, you can specify that users with the Account Representative role can execute any command in the AccountRepresentativesCmdResourceGroup resource group. Or, as depicted in the following diagram, another example policy is to specify that all store administrators can perform any action specified in the ExecuteCommandAction Group on any resource that is specified by the StoreAdminCmdResourceGrp.

**Note:** The XML information for the Conditions column of the MBRGRPCOND table is generated when you use the Administration Console to set up your access groups. For information about using the Administration Console to set up access groups, refer to the WebSphere Commerce online help.

ACPOLICY

PolicyName	Member_Id	MbrGrp_Id	AcActGrp_id	AcResGrp_Id	AcRelGrp_Id
StoreAdministrators ExecuteStoreAdmin CmdResourceGroup	-2001	-8	10052	10018	null

MBRGRP

MbrGrp_Id	MbrGrpName
-8	StoreAdministrators

MBRGRPCOND

MbrGrp_Id	Conditions
-8	<pre>&lt;profile&gt; &lt;simpleCondition&gt;   &lt;variable name="role"/&gt;   &lt;operator name="="/&gt;   &lt;value data="Store Administrator"/&gt; &lt;/simpleCondition&gt; &lt;/profile&gt;</pre>

ACACTGRP

AcActGrp_Id	GroupName
10052	ExecuteCommandActionGroup

ACRESGRP

AcResGrp_Id	GrpName
10018	StoreAdminCmdResourceGroup

Figure 6.

A command-level access control policy always has the ExecuteCommandActionGroup as the action group for controller commands. For views, the resource group is always ViewCommandResourceGroup.

All controller commands must be protected by command-level access control. In addition, any view that can be called directly, or that can be launched by a redirect from another command (in contrast to being launched by forwarding to the view) must be protected by command-level access control.



Command-level access control does not consider the resource that the command would act upon. It merely determines if the user is allowed to execute the particular command. If the user is allowed to execute the command, a subsequent resource-level access control policy could be applied to determine if the user can access the resource in question.

Consider when a store administrator attempts to perform an administrative task. The first level of access control checking would be to determine if this user is allowed to execute the particular store administration command. Once it has been determined that the user is in fact permitted to do this (because store administrators are allowed to execute commands in the `storeAdminCmds` group), a resource-level access control policy may be invoked. This policy may state that store administrators are only permitted to perform administrative tasks for stores that are owned by the organization for which the user is a store administrator.

To summarize, in command-level access control the “resource” is the command itself and the “action” is merely to execute the command (in other words, to instantiate the command object). The access control check determines if the user is permitted to execute the command. By contrast, in resource-level access control the “resource” is any protectable resource that the command or bean accesses and the “action” is the command itself.

## **Access control interactions**

This section presents the interaction diagram showing how access control works in the WebSphere Commerce access control policy framework.

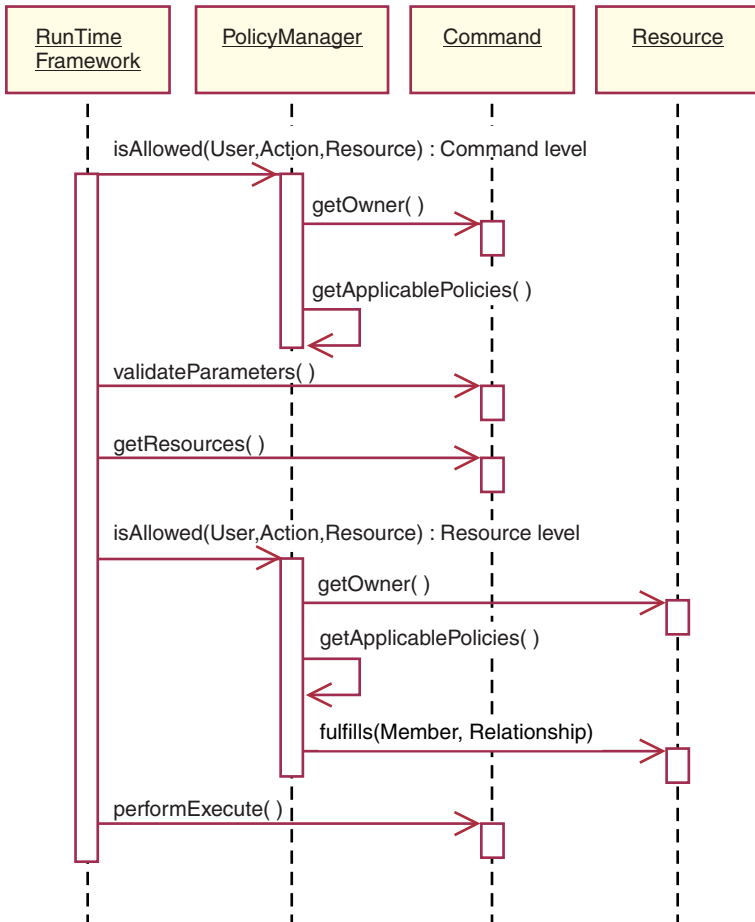


Figure 7.

The preceding diagram shows actions that are performed by the access control *policy manager*. The access control policy manager is the access control component that determines whether or not the current user is allowed to execute the specified action on the specified resource. It determines this by searching through the policies owned by the resource's owner and its ancestor organizations. If at least one policy grants access, then permission is granted.

The following list describes the actions from the preceding interaction diagram. They are ordered from the top of the diagram to the bottom.

1. `isAllowed()`  
The run-time components determine if the user has command level access for either the controller command or view.
2. `getOwner()`  
The access control policy manager determines the owner of the command-level resource. The default implementation returns the member identifier (`memberId`) of the owner of the store (`storeId`) that is in the command context. If there is no store identifier in the command context, then the root organization (`-2001`) is returned.
3. `getApplicablePolicies()`  
The access control policy manager finds and processes the applicable policies, based on the specified user, action and resource.
4. `validateParameters()`  
Initial parameter checking and resolving.

5. `getResources()`  
Returns an access vector that is a vector of resource-action pairs.  
If nothing is returned, resource-level access control checking is not performed. If there are resources that should be protected an access vector (consisting of resource-action pairs) should be returned.  
Each *resource* is an instance of a protectable object (an object that implements the `com.ibm.commerce.security.Protectable` interface). In many cases, the resource is an access bean.  
An access bean may not implement the `com.ibm.commerce.security.Protectable` interface, however, the access control check can still occur as long as the corresponding enterprise bean is protected, according to the information included in “Implementing access control in enterprise beans” on page 95.  
The *action* is a string representing the operation to be performed on the resource. In most cases, the action is the interface name of the command.
6. `isAllowed()`  
The run-time components determine if the user has resource level access to all of the resource-action pairs specified by `getResources()`.
7. `getOwner()`  
The resource returns the `memberId` of its owner. This determines which policies apply. Only policies that are owned by the resource owner and its ancestor organizations apply.
8. `getApplicablePolicies()`  
The access control policy manager searches for applicable policies and then applies them. If at least one policy per resource-action pair that grants the user permission to access the resource is found, then access is granted, otherwise access it is denied.
9. `fulfills()`  
If an applicable policy has a relationship group specified, a check is done on the resource to see if the member satisfies the specified relationship or relationships, with respect to the resource.
10. `performExecute()`  
The business logic of the command.

## Protectable interface

A key factor for having a resource protected by the WebSphere Commerce access control policies, is that the resource must implement the `com.ibm.commerce.security.Protectable` interface. This interface is most commonly used with enterprise beans and data beans, but only those particular beans that require protection need to implement the interface.

With the `Protectable` interface, a resource must provide two key methods: `getOwner()`, and `fulfills(Long member, String relationship)`.

Access control policies are owned by organizations or organizational entities. The `getOwner` method returns the `memberId` of the owner of the protectable resource. After the access control policy manager determines the owner of the resource, it also gets the `memberId` of each of the ancestors for the owner in the member hierarchy. All access control policies that belong to the owner from the original `getOwner` request as well as all access control policies that belong to any of the owner’s ancestors are then applied.

Access control policies that apply to the specified owner, as well as access control policies that apply to any of the owner's higher level ancestors in the membership hierarchy, are applied.

The `fulfills` method only returns true if the given member satisfies the required relationship with respect to the resource. Typically the member is a single user, however it can also be an organization. It would be an organization if you are using a relationship group in the access control policy.

## Groupable interface

The application of an access control policy is specific to a group of resources. Resource groupings can be made based upon attributes such as the class name, the state of an order or the `storeId` value.

If a resource is going to be grouped by an attribute other than its class name for the purpose of applying access control policies, it must implement the `com.ibm.commerce.grouping.Groupable` interface.

The following code snippet represents the `Groupable` interface:

```
Groupable interface {
    Object getGroupingAttributeValue (String attributeName, GroupContext context)
}
```

For example, to implement a policy that only applies to orders that are in the pending state (`status = P (pending)`), the remote interface of the `Order` entity bean implements the `Groupable` interface and the value for `attributeName` is set to `"status"`.

Usage of the `Groupable` interface is rare.

## Finding more information about access control

For more information about the WebSphere Commerce access control model, refer to the *WebSphere Commerce Access Control Guide*. This guide provides a detailed overview of access control and describes how to use the Administration Console to create or modify policies, action groups, and resource groups.

---

## Implementing access control

This section describes how to implement access control in customized code.

### Identifying protectable resources

In general, enterprise beans and data beans are resources that you may want to protect. However, not all enterprise beans and data beans should be protected. Within the existing WebSphere Commerce application, resources that require protection already implement the protectable interface. The question of what to protect comes into play when you create new enterprise beans and data beans. Deciding which resources to protect depends upon your application.

If a command returns an enterprise bean in the `getResources` method, then the enterprise bean must be protected because the access control policy manager will call the `getOwner` method on the enterprise bean. The `fulfills` method will also be called if a relationship is specified in the corresponding resource-level access control policy.

If you were to implement the protectable interface (and therefore, put the resource under protection) for all of your own enterprise beans and data beans, your application could require many policies. As the number of policies increases, performance may degrade and policy management becomes more challenging.

A theoretical distinction is made between primary resources and dependent resource. A *primary resource* can exist upon its own. A *dependent resource* exists only when its related primary resource exists. For example, in the out-of-the-box WebSphere Commerce application code, the Order entity bean is a protectable resource, but the OrderItem entity bean is not. The reason for this is that the existence of an OrderItem depends upon an Order -- the Order is the primary resource and the OrderItem is a dependent resource. If a user should have access to an Order, it should also have access to the items in the order.

Similarly, the User entity bean is a protectable resource, but the Address entity bean is not. In this case, the existence of the address depends on the user, so anything that has access to the user, should also have access to the address.

Primary resources should be protected, but dependent resources often do not require protection. If a user is allowed to access a primary resource, it makes sense that, by default, the user should also be allowed to access its dependent resources.

## Implementing access control in enterprise beans

If you create new enterprise beans that require protection by access control policies, you must do the following:

1. Create a new enterprise bean, ensuring that it extends from `com.ibm.commerce.base.objects.ECEntityBean`.
2. Ensure that the remote interface of the bean extends the `com.ibm.commerce.security.Protectable` interface.
3. If resources with which the bean interacts are grouped by an attribute other than the resource's Java class name, the remote interface of the bean must also extend the `com.ibm.commerce.grouping.Groupable` interface.
4. The enterprise bean class contains default implementations for the following methods:
  - `getOwner`
  - `fulfills`
  - `getGroupingAttributeValue`

Override any methods that you need. At a minimum, you must override the `getOwner` method.

The default implementations of these methods are shown in the following code snippets.

```
*****
public Long getOwner() throws Exception, java.rmi.RemoteException
{
    return null;
}
*****
*****
public boolean fulfills(Long member, String relationship)
    throws Exception, java.rmi.RemoteException
{
    return false;
}
*****
```

```

*****
public Object getGroupingAttributeValue(String attributeName,
    GroupingContext context) throws Exception, java.rmi.RemoteException
{
    return null;
}
*****

```

The following are sample implementations of these methods based on the OrderBean bean:

```

*****
public Long getOwner() throws Exception, java.rmi.RemoteException
{
    com.ibm.commerce.common.objects.StoreEntityAccessBean storeEntAB = new
    com.ibm.commerce.common.objects.StoreEntityAccessBean();
    storeEntAB.setInitKey_storeEntityId(getStoreEntityId().toString());
    return storeEntAB.getMemberIdInEJBType();
}
*****
public boolean fulfills(Long member, String relationship)
    throws Exception, java.rmi.RemoteException
{
    if (relationship.equalsIgnoreCase("creator"))
    {
        return member.equals(getMemberId());
    }
    else if (relationship.equalsIgnoreCase (
        com.ibm.commerce.base.helpers.EJBConstants.
        SAME_ORGANIZATIONAL_ENTITY_AS_CREATOR_RELATION)) {
        com.ibm.commerce.user.objects.UserAccessBean creator = new
        com.ibm.commerce.user.objects.UserAccessBean();
        creator.setInitKey_MemberId(getMemberId().toString());
        com.ibm.commerce.user.objects.UserAccessBean ab = new
        com.ibm.commerce.user.objects.UserAccessBean();
        ab.setInitKey_MemberId(member.toString());
        if (ab.getParentMemberId().equals(creator.getParentMemberId()))
            return true;
    }
    return false;
}
*****
public Object getGroupingAttributeValue(String attributeName,
    GroupingContext context) throws Exception
{
    if (attributeName.equalsIgnoreCase("Status"))
        return getStatus();
    return null;
}
*****

```

5. Create (or recreate) the enterprise bean's access bean and generated code.

## Implementing access control in data beans

If a data bean is to be protected, it can either be directly, or indirectly protected by access control policies. If a data bean is directly protected, then there exists an access control policy that applies to that particular data bean. If a data bean is indirectly protected, it delegates protection to another data bean, for which an access control policy exists.

If you create a new data bean that is to be directly protected by an access control policy, the data bean must do the following:

1. Implement the `com.ibm.commerce.security.Protectable` interface. As such, the bean must provide an implementation of the `getOwner()` and `fulfills(Long member, String relationship)` methods. These should be implemented on the remote interface of the bean.

When a data bean implements the `Protectable` interface, the data bean manager calls the `isAllowed` method to determine if the user has the appropriate access control privileges, according to the current access control policy. The `isAllowed` method is described by the following code snippet:

```
isAllowed(Context, "Display", protectable_databean);
```

2. If resources that the bean interacts with are grouped by an attribute other than the resource's Java class name, the bean must implement the `com.ibm.commerce.grouping.Groupable` interface.
3. Implement the `com.ibm.commerce.security.Delegator` interface. This interface is described by the following code snippet:

```
Interface Delegator {  
    Protectable getDelegate();  
}
```

**Note:** In order to be directly protected, the `getDelegate` method should return the data bean itself (that is, the data bean delegates to itself for the purpose of access control).

The distinction between which data beans should be protected directly versus which should be protected indirectly is similar to the distinction between primary and dependent resources. If the data bean object can exist on its own, it should be directly protected. If the existence of data bean depends upon the existence of another data bean, then it should delegate to the other data bean for protection.

An example of a data bean that would be directly protected is the `Order` data bean. An example of a data bean that would be indirectly protected is the `OrderItem` data bean.

If you create a new data bean that is to be indirectly protected by an access control policy, the data bean must do the following:

1. Implement the `com.ibm.commerce.security.Delegator` interface. This interface is described by the following code snippet:

```
Interface Delegator {  
    Protectable getDelegate();  
}
```

**Note:** The data bean returned by `getDelegate` must implement the `Protectable` interface.

If a data bean does not implement the `Delegator` interface, it is populated without the protection of access control policies.

## Implementing access control in controller commands

When creating a new controller command, the implementation class for the new command should extend the `com.ibm.commerce.commands.ControllerCommandImpl` class and its interface should extend the `com.ibm.commerce.command.ControllerCommand` interface.

For command level policies for controller commands, the interface name of the command is specified as a resource. In order for a resource to be protected, it must implement the `Protectable` interface. According to the WebSphere Commerce

programming model, this is accomplished by having the command's interface extend from `com.ibm.commerce.command.ControllerCommand` interface, and the command's implementation extend from `com.ibm.commerce.commands.ControllerCommandImpl`. The `ControllerCommand` interface extends `com.ibm.commerce.command.AccCommand` interface, which in turn extends `Protectable`. The `AccCommand` interface is the minimum interface that a command should implement in order to be protected by command level access control.

If the command accesses resources that should be protected, create a private instance variable of type `AccessVector` to hold the resources. Then override the `getResources` method since the default implementation of this method is to return a null value and therefore, no resource checking occurs.

In the new `getResources` method, you should return an array of resources or of resource-action pairs upon which the command can act. When an action is not explicitly specified, the action defaults to the interface name of the command being executed.

Additionally, it is recommended that the method determines if it must instantiate the resource or if it can use the existing instance variable holding the reference to the resource. Checking to see if the resource object already exists can help to improve system performance. You can then use the same `getResources` method, if required, in the `performExecute` method of the new controller command.

The following is an example of the `getResources` method:

```
private AccessVector resources = null;

public AccessVector getResources() throws ECEException {
    if (resources == null) {
        OrderAccessBean orderAB = new OrderAccessBean();
        orderAB.setInitKey_orderId(getOrderId().toString());
        resources = new AccessVector(orderAB);
    }
    return resources;
}
```

As an example, consider the `OrderItemUpdate` command. The `getResources` method of this command returns the `Order` and `User` protectable objects. Since the action is not specified, it defaults to the interface for the `OrderItemUpdate` command.

Multiple resources may be returned by the `getResources` method. When this occurs, a policy that gives the user access to all of the specified resources must be found if the action is to be carried out. If a user had access to two out of three resources, the action may not proceed (three out of three would be required).

If you need to perform additional parameter checking or resolving of parameters in the controller command, you can use the `validateParameters()` method. This is optional.

### **Additional resource level checking**

It is not always possible to determine all of the resources that need to be protected, at the time the `getResources` method of the controller command is called.

If necessary, a task command can also implement a `getResources` method to return a list of resources, upon which the command can act.



Another way to invoke resource level checking is to make direct calls to the access control policy manager, using the `checkIsAllowed(Object resource, String action)` method. This method is available to any class that extends from the `com.ibm.commerce.command.AbstractEactableCommand` class. For example, the following classes extend from the `AbstractEactableCommand` class:

- `com.ibm.commerce.command.ControllerCommandImpl`
- `com.ibm.commerce.command.DataBeanCommandImpl`

The `checkIsAllowed` method is also available to classes that extend the `com.ibm.commerce.command.AbstractECCCommand` class. For example, the following class extends from the `AbstractECCCommand` class:

- `com.ibm.commerce.command.TaskCommandImpl`

The following shows the signature of the `checkIsAllowed` method:

```
void checkIsAllowed(Object resource, String action)
    throws ECEException
```

This method throws an `ECAApplicationException` if the current user is not allowed to perform the specified action on the specified resource. If access is granted, then the method simply returns.

### Access control for “create” commands

Since the `getResources` method is called before the `performExecute` method in a command, a different approach must be taken for access control for resources that are not yet created. For example, if you have a `WidgetAddCmd`, the `getResources` method cannot return the resource that is about to be created. In this case, the `getResources` method should return the creator of the resources. For example, a command is created by a command factory, an order is created within a store, and a user is created within an organization.

### Default implementations for command-level access control

For command-level access control, the default implementation of the `getOwner()` method returns the `memberId` of the store owner, if the `storeId` is specified. If the `storeId` is not specified, the `memberId` of the root organization is returned (`memberId = -2001`).

The default implementation of the `getResources()` method returns `null`.

The default implementation of the `validateParameters()` does nothing.

## Implementing access control policies in views

Resource-level access control for views is performed by the data bean manager. The data bean manager is invoked in the following cases:

1. When the JSP template includes the `<useBean>` tag and the data bean is not in the attribute list.
2. When the JSP template includes the following activate method:

```
DataBeanManager.activate(xyzDatabean, request);
```

**Note:** Any data bean that is to be protected (either directly or indirectly) must implement the `Delegator` interface. Any data bean that is to be directly protected will delegate to itself, and thus must also implement the `Protectable` interface. Data beans that are indirectly protected should delegate to a data bean that implements the `Protectable` interface.

While it is not recommended, a bypass of the access control checks occurs in the following cases:

1. If the JSP template makes direct calls to access beans, rather than using data beans.
2. If the JSP template invokes the data bean's populate() method directly.

If the results of a controller command are to be forwarded to a view (using the ForwardViewCommand), then command-level access control is not performed on the views. Furthermore, if the controller command puts the populated data beans (that are used in the view) on the attribute list of the response property and then forwards to a view, the JSP template can access the data without going through the data bean manager. This does require that the <useBean> tags are used in the JSP template. This can be a way to make a JSP template more efficient, since it can bypass any redundant resource-level access control checks on resources (data beans) to which the user has already been granted access via the controller command.

---

## Part 5. Appendixes



---

## 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.

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

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 information 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 obligation 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 Canada Ltd.  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
Canada

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 document 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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.

Credit card images, trademarks and trade names provided in this product should be used only by merchants authorized by the credit card mark's owner to accept payment via that credit card.

---

## Trademarks

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

400	AIX	AS/400
DB2	IBM	iSeries
OS/2	SecureWay	WebSphere

Domino is a registered trademark of Lotus Development Corporation and/or IBM Corporation in the United States, other countries, or both.

Netscape is a registered trademark of Netscape Communications Corporation in the United States, other countries, or both.

Solaris, Solaris Operating Environment, Java, JavaBeans, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc.

VeriSign and the VeriSign logo are trademarks and service marks or registered trademarks and service marks of VeriSign, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

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









Printed in USA