

Course Guide

WebSphere Application Server V9 Administration

Course code WA590 / ZA590 ERC 1.0



November 2016 edition

Notices

This information was developed for products and services offered in the US.

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

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

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

© Copyright International Business Machines Corporation 2016.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

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

Contents

Trademarks	xii
Course description	xiii
Agenda	xv
Unit 1. WebSphere product family overview	1-1
How to check online for course material updates	1-2
Unit objectives	1-3
Topics	1-4
1.1. Platform and packaging	1-5
Platform and packaging	1-6
WebSphere software platform	1-7
WebSphere application infrastructure: The big picture	1-8
WebSphere V9 profile by product: Quick view	1-9
WebSphere Application Server V9 packaging	1-11
Product packaging comparison (1 of 2)	1-12
Product packaging comparison (2 of 2)	1-13
Supported specifications in V9	1-14
1.2. Related WebSphere products and cloud-based offerings	1-15
Related WebSphere products and cloud-based offerings	1-16
IBM WebSphere Application Server Developer Tools for Eclipse	1-17
WebSphere eXtreme Scale	1-18
WebSphere supports the entire hybrid cloud landscape	1-20
Comparison of cloud features	1-21
Pattern-based deployments on SoftLayer	1-22
IBM Bluemix and WebSphere	1-23
IBM Bluemix: Local and dedicated	1-24
WebSphere Application Server for Bluemix	1-25
WebSphere Application Server and Docker containers	1-26
WebSphere Application Server V9 standards and changes	1-27
Unit summary	1-28
Review questions	1-29
Review answers	1-30
Unit 2. WebSphere Application Server architecture: Stand-alone	2-1
Unit objectives	2-2
Topics	2-3
2.1. Architecture runtime	2-4
Architecture runtime	2-5
Version 9 packaging	2-6
WebSphere Application Server basics	2-7
WebSphere architecture runtime (1 of 10)	2-8
WebSphere architecture runtime (2 of 10)	2-10
WebSphere architecture runtime (3 of 10)	2-11
WebSphere architecture runtime (4 of 10)	2-12
WebSphere architecture runtime (5 of 10)	2-13
WebSphere architecture runtime (6 of 10)	2-14
WebSphere architecture runtime (7 of 10)	2-15
WebSphere architecture runtime (8 of 10)	2-16

WebSphere architecture runtime (9 of 10)	2-17
WebSphere architecture runtime (10 of 10)	2-18
JDBC providers	2-19
Data sources	2-20
2.2. Administration architecture	2-21
Administration architecture	2-22
WebSphere administration architecture (1 of 4)	2-23
WebSphere administration architecture (2 of 4)	2-24
WebSphere administration architecture (3 of 4)	2-25
WebSphere administration architecture (4 of 4)	2-26
2.3. Profiles	2-27
Profiles	2-28
WebSphere profile overview	2-29
WebSphere profile benefits	2-30
Managing profiles	2-31
Profile types	2-32
Unit summary	2-33
Review questions	2-34
Review answers	2-35
Exercise: Profile creation	2-36
Exercise objectives	2-37
Unit 3. WebSphere Application Server administrative console	3-1
Unit objectives	3-2
Administrative console	3-3
Starting the administrative console	3-4
Administrative console in a cell topology	3-5
Console login	3-6
Recovering prior changes	3-7
Administrative console session timeout	3-8
Administrative console panels	3-9
Administrative console areas	3-10
Administrative console banner	3-11
Administrative console navigation tree (1 of 2)	3-12
Administrative console navigation tree (2 of 2)	3-13
Administrative console help (1 of 2)	3-14
Administrative console help (2 of 2)	3-15
Administrative console preferences	3-17
Administrative console preferences, filters, and scope	3-18
Guided Activities	3-19
My tasks	3-20
Users and groups	3-21
Troubleshooting information	3-22
Tivoli Performance Viewer	3-23
Unit summary	3-24
Review questions	3-25
Review answers	3-26
Exercise: Exploring the administrative console	3-27
Exercise objectives	3-28
Unit 4. Introduction to the PlantsByWebSphere application	4-1
Unit objectives	4-2
PlantsByWebSphere application	4-3
PlantsByWebSphere sample	4-4
Home page: <a href="http://<hostname>/PlantsByWebSphere">http://<hostname>/PlantsByWebSphere	4-5
Login and Registration	4-6

My Account	4-7
Shopping	4-8
Select an item	4-9
Shopping cart	4-10
Checking out: Billing information	4-11
Checking out: Submit	4-12
Help page	4-13
View Server Info page	4-14
Admin home page	4-15
HTML documentation for PlantsByWebSphere (1 of 2)	4-16
HTML documentation for PlantsByWebSphere (2 of 2)	4-17
Unit summary	4-18
Unit 5. Application assembly.....	5-1
Unit objectives	5-2
Topics	5-3
5.1. Overview of Java EE application packaging.....	5-4
Overview of Java EE application packaging	5-5
Overview of application assembly and installation	5-6
Java EE packaging	5-7
WebSphere: Application packaging	5-8
WebSphere application contents	5-9
5.2. Application assembly and deployment tools.....	5-10
Application assembly and deployment tools	5-11
Development and assembly tools	5-12
IBM WebSphere Application Server Developer Tools for Eclipse	5-13
IBM WebSphere Application Server Developer Tools for Eclipse tasks	5-14
Features in assembly tools (1 of 2)	5-15
Features in assembly tools (2 of 2)	5-16
Assembling an enterprise application	5-17
IBM WebSphere Application Server Developer Tools for Eclipse usage	5-18
Import modules	5-19
Specify runtime environment	5-20
Generate deployment descriptor	5-21
Java EE perspective	5-22
Application deployment descriptor	5-23
Enterprise applications without application.xml	5-24
Packaging enterprise applications for deployment	5-25
Application scope resources	5-26
Data sources	5-27
5.3. Enhanced EAR.....	5-28
Enhanced EAR	5-29
Enhanced EAR	5-30
Unenhancing an EAR file	5-31
Dealing with enhanced EAR files at deployment time	5-32
5.4. Java EE modules	5-33
Java EE modules	5-34
EJB module assembly	5-35
EJB components are included in the web module	5-36
Web module assembly	5-37
Web module	5-38
Adding metadata through the menus	5-39
Application client assembly	5-40
Generating an EAR file for deployment	5-41
Export an EAR file	5-42
Final export step	5-43

5.5. Java EE annotations	5-44
Java EE annotations	5-45
Annotations in Java code	5-46
How annotations help the developer	5-47
Example of declaring a session bean with annotations	5-48
Stateless session bean as shown in the Explorer view	5-49
Using annotations to inject EJB references	5-50
Example of injecting an EJB reference	5-51
Application bindings	5-52
WebSphere default bindings for bean interfaces	5-53
Specifying Java EE metadata for deployment	5-54
Unit summary	5-55
Review questions	5-56
Review answers	5-57
Exercise: Assembling an application	5-58
Exercise objectives	5-59
Unit 6. Application installation	6-1
Unit objectives	6-2
Topics	6-3
6.1. Application installation	6-4
Application installation	6-5
Installing enterprise applications	6-6
Installation tasks	6-7
Creating a J2C authentication alias	6-8
Step 1: Enter basic data source information	6-9
Step 2: Select JDBC provider	6-10
Step 2.1: Create new JDBC provider	6-11
Step 2.2: Enter database class path information	6-12
Step 3: Enter database-specific properties for the data source	6-13
Step 4: Set up security aliases	6-14
Step 5: Summary of data source creation	6-15
Installing a new application	6-16
Example of fast path installation	6-17
Example of detailed installation	6-18
Enhanced EAR file	6-19
Removing enhancements	6-20
6.2. Application settings and interaction	6-21
Application settings and interaction	6-22
Starting an application	6-23
Application update	6-24
Other application configuration settings	6-25
Application startup behavior and auto start	6-26
View the application deployment descriptor	6-27
The application deployment descriptor	6-28
Manage modules (1 of 4)	6-29
Manage modules (2 of 4)	6-30
Manage modules (3 of 4)	6-31
Manage modules (4 of 4)	6-32
Metadata for modules (1 of 2)	6-33
Metadata for modules (2 of 2)	6-34
6.3. Monitored directory	6-35
Monitored directory	6-36
Overview of monitored directory deployment	6-37
Supported tasks	6-38
Enabling the monitored directory	6-39

Notes about the monitored directory	6-40
Drag-and-drop properties files	6-42
Properties file based configuration	6-43
Steps to use a properties file to deploy applications (1 of 2)	6-44
Steps to use a properties file to deploy applications (2 of 2)	6-45
Unit summary	6-46
Review questions	6-47
Review answers	6-48
Exercise: Installing an application	6-49
Exercise objectives	6-50
Unit 7. Problem determination	7-1
Unit objectives	7-2
Topics	7-3
7.1. Server logs and diagnostic tracing	7-4
Server logs and diagnostic tracing	7-5
Examining server log files	7-6
WebSphere Application Server logs	7-7
Server log files: Types and locations	7-9
Configuring JVM logs	7-11
Viewing runtime messages in the console (1 of 2)	7-12
Viewing runtime messages in the console (2 of 2)	7-13
HTTP plug-in logs and tracing	7-14
Diagnostic tracing	7-15
Using diagnostic tracing	7-16
Enable and configure tracing	7-17
Setting the log detail level (1 of 2)	7-18
Setting the log detail level (2 of 2)	7-19
Trace output content and format	7-20
Reading a log or trace file (1 of 2)	7-21
Reading a log or trace file (2 of 2)	7-22
High Performance Extensible Logging (HPEL)	7-23
HPEL logging and tracing configuration	7-24
Configure HPEL logging	7-25
The HPEL Log Viewer	7-26
logViewer command-line tool (1 of 2)	7-28
logViewer command-line tool (2 of 2)	7-29
What is Cross Component Trace (XCT)?	7-30
Administering XCT	7-31
XCT request IDs	7-32
Use XCT request ID information to track requests	7-33
7.2. Gathering diagnostic data	7-35
Gathering diagnostic data	7-36
Gathering JVM diagnostic data	7-37
Enable verbose garbage collection	7-38
Java memory dumps and cores	7-39
Generating a JVM thread dump (javacore)	7-40
Generating a JVM heap dump	7-41
Hung thread detection	7-42
Connection leak diagnostic messages	7-43
Checking version levels and applying APARs	7-44
7.3. Problem determination tools	7-45
Problem determination tools	7-46
Problem determination tool availability	7-47
Problem determination tools: Administrative console	7-48
Dumping the JNDI namespace	7-49

7.4. IBM Support Assistant	7-50
IBM Support Assistant	7-51
Introduction to IBM Support Assistant	7-52
IBM Support Assistant: Overview videos available	7-54
IBM Support Assistant example: Javacore analysis	7-55
IBM Support Assistant example: Thread status analysis	7-56
IBM Support Assistant example: Memory visualizer	7-57
IBM Support Assistant example: Memory diagnostic	7-58
Unit summary	7-59
Review questions	7-60
Review answers	7-61
Exercise: Problem determination	7-62
Exercise objectives	7-63
Unit 8. Introduction to wsadmin and scripting	8-1
Unit objectives	8-2
wsadmin versus administration console	8-3
Scripting benefits	8-4
wsadmin	8-5
wsadmin invocation options	8-7
Jython versus Jacl	8-10
Administrative functions that use wsadmin	8-11
Administrative objects in wsadmin	8-13
Starting wsadmin	8-14
Starting wsadmin with security enabled	8-16
wsadmin properties	8-18
Profile scripts	8-21
AdminConfig: Managing configurations	8-23
AdminApp: Managing applications	8-24
AdminControl: Managing running objects	8-25
AdminTask: Accessing administrative functions	8-26
Help within wsadmin	8-28
Administrative command help	8-30
Important points to remember when using wsadmin	8-31
Scripting: Simple script	8-32
Scripting: Looping script	8-33
Jython script library	8-34
How to use the Jython script library	8-36
Configuration repository: The issues	8-38
Properties file based configuration: A solution	8-39
Properties file configuration content	8-41
Properties file configuration commands	8-43
Command assistance	8-46
Unit summary	8-47
Review questions	8-48
Review answers	8-49
Exercise: Using wsadmin	8-50
Exercise objectives	8-51
Unit 9. WebSphere security	9-1
Unit objectives	9-2
Topics	9-3
9.1. WebSphere security basics	9-4
WebSphere security basics	9-5
Basic security end-to-end model	9-6
WebSphere Application Server security overview	9-7

WebSphere security service: Wide view	9-9
Types of security	9-10
Administrative security	9-11
Application security	9-12
Authentication and authorization: What is the difference?	9-13
Challenge mechanism: Authentication basic steps	9-14
Challenge mechanism: Basic authentication	9-15
Challenge mechanism: Form-based authentication	9-16
Defining the challenge type	9-17
9.2. WebSphere user registries	9-18
WebSphere user registries	9-19
Registries and authentication mechanisms	9-20
Defining user registries	9-21
Manual security configuration	9-22
Security wizard: Step 1	9-23
Security wizard: Step 2	9-24
Security wizard: Step 3	9-25
Security wizard: Step 4	9-26
User registry support	9-27
Federated repositories	9-28
Custom registry: Configuration	9-29
Authentication mechanism: LTPA	9-30
LTPA provides delegation	9-31
LTPA provides single sign-on (SSO)	9-32
9.3. Administrative security	9-33
Administrative security	9-34
Administrative security	9-35
Console security	9-36
More console security roles	9-37
Administrative roles	9-38
Console security: Creating users and groups	9-39
Console security: Mapping users and groups	9-40
9.4. Application security	9-41
Application security	9-42
Authorization	9-43
Security roles: Application authorization	9-44
Securing Java EE application artifacts: Part 1	9-45
Securing Java EE application artifacts: Part 2	9-46
Applying application security	9-47
Creating security constraints	9-48
Using the console to map security roles	9-49
9.5. Java 2 security	9-50
Java 2 security	9-51
Java security model	9-52
Java security overview	9-53
Enabling Java 2 security	9-54
9.6. Security auditing	9-56
Security auditing	9-57
Security auditing	9-58
Enable security auditing	9-59
Viewing audit data	9-60
Securing audit records	9-61
Unit summary	9-62
Review questions	9-63
Review answers	9-64
Exercises: Configuring WebSphere Application Server security, Configuring application security	9-65

Exercise objectives	9-66
Unit 10. Performance monitoring	10-1
Unit objectives	10-2
Topics	10-3
10.1. Performance tuning and monitoring	10-4
Performance tuning and monitoring	10-5
The need for performance monitoring and tuning	10-6
Tuning performance suggested practices	10-7
Performance terminology	10-8
Tuning parameter hot list	10-9
Solving performance problems	10-11
Measuring performance and collecting data	10-12
WebSphere performance tools (1 of 3)	10-14
WebSphere performance tools (2 of 3)	10-15
WebSphere performance tools (3 of 3)	10-16
PMI architecture	10-18
Types of performance data	10-20
PMI data collection settings	10-21
Using the administrative console to enable PMI	10-22
Start monitoring	10-23
Tivoli Performance Viewer (1 of 4)	10-25
Tivoli Performance Viewer (2 of 4)	10-26
Tivoli Performance Viewer (3 of 4)	10-27
Tivoli Performance Viewer (4 of 4)	10-28
Summary reports	10-29
Example: Servlet Summary Report	10-31
Performance servlet overview	10-32
Performance servlet output	10-33
10.2. Request metrics	10-35
Request metrics	10-36
Request metrics (RM) overview	10-37
Enabling request metrics collection	10-38
Isolating performance for specific types of requests	10-40
Example request metrics data	10-41
10.3. Performance advisors	10-42
Performance advisors	10-43
Performance advisors overview	10-44
Performance and Diagnostic Advisor (1 of 5)	10-46
Performance and Diagnostic Advisor (2 of 5)	10-47
Performance and Diagnostic Advisor (3 of 5)	10-48
Performance and Diagnostic Advisor (4 of 5)	10-50
Performance and Diagnostic Advisor (5 of 5)	10-51
Tivoli Performance Viewer advisor	10-52
Examples of performance advice	10-53
Viewing performance advice	10-55
Performance advice detail	10-56
Performance advisor suggested practices	10-57
Unit summary	10-59
Review questions	10-60
Review answers	10-61
Exercise: Using the performance monitoring tools	10-62
Exercise objectives	10-63
Unit 11. Course summary	11-1
Unit objectives	11-2

Course objectives	11-3
To learn more on the subject	11-4
Enhance your learning with IBM resources	11-5
Unit summary	11-6
Course completion	11-7
Appendix A. List of abbreviations	A-1
Appendix A. Resource guide.....	A-1
Training1
Social media links1
Support2
Middleware documentation and tips2
Services3

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®
DB™
Express®
IBM z Systems™
Rational®
WebSphere®

Bluemix®
DB2®
HACMP™
Notes®
Redbooks®
z Systems™

DataPower®
developerWorks®
IBM z™
PureApplication®
Tivoli®
z/OS®

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware and the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the “Marks”) of VMware, Inc. in the United States and/or other jurisdictions.

SoftLayer® is a trademark or registered trademark of SoftLayer, Inc., an IBM Company.

Other product and service names might be trademarks of IBM or other companies.

Course description

WebSphere Application Server V9 Administration

Duration: 2.5 days

Purpose

This course teaches you the skills that are needed to administer IBM WebSphere Application Server V9.

This release of IBM WebSphere Application Server provides enhanced support for standards (notably Java 7 EE), emerging technology, and a choice of development frameworks.

In this course, you learn how to configure and maintain IBM WebSphere Application Server V9 in a single-server environment. You learn how to deploy enterprise Java applications in a single computer configuration. In addition, you learn how to work with features of WebSphere Application Server V9, such as the wsadmin scripting interface, security, and performance monitoring.

Hands-on exercises throughout the course give you practical experience with the skills you develop in the lectures.

Audience

This course is designed for administrators who configure and manage web-based applications on WebSphere Application Server. Web administrators, application developers and deployers, security specialists, and application architects can also benefit from this course.

Prerequisites

- Basic operational skills for the Linux operating system
- Administrative skills for a web server, such as IBM HTTP Server or Apache
- Basic understanding of cloud concepts, private, public, and hybrid clouds, and traditional on-premises environments

Objectives

- Relate WebSphere Application Server to the WebSphere family of products
- Describe the features and standards in WebSphere Application Server V9
- Describe the use of WebSphere Application Server in cloud, hybrid cloud, and on-premises environments
- Describe the architectural concepts that are related to WebSphere Application Server
- Assemble and install server-side Java enterprise applications

- Use WebSphere administrative tools to configure and manage enterprise applications
- Use wsadmin scripting
- Configure WebSphere Application Server security
- View performance information about server and application components
- Troubleshoot problems by using problem determination tools and log files

Contents

- Intermediate training for administrators
- How to assemble and install applications
- Overview of security, command-line interface, and problem determination tools

Agenda

**Note**

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

- (00:15) Course introduction
- (01:00) Unit 1. WebSphere product family overview
- (01:00) Unit 2. WebSphere Application Server architecture: Stand-alone
- (00:45) Exercise 1. Profile creation (part 1)
- (00:30) Unit 3. WebSphere Application Server administrative console
- (00:45) Exercise 2. Exploring the administrative console
- (00:15) Unit 4. Introduction to the PlantsByWebSphere application
- (01:00) Unit 5. Application assembly
- (00:45) Exercise 3. Assembling an application

Day 2

- (00:45) Unit 6. Application installation
- (01:00) Exercise 4. Installing an application
- (00:45) Unit 7. Problem determination
- (01:45) Exercise 5. Problem determination
- (01:00) Unit 8. Introduction to wsadmin and scripting
- (01:00) Exercise 6. Using wsadmin

Day 3

- (01:30) Unit 9. WebSphere security
- (00:30) Exercise 7. Configuring WebSphere Application Server security
- (00:30) Exercise 8. Configuring application security
- (00:45) Unit 10. Performance monitoring
- (00:30) Exercise 9. Using the performance monitoring tools
- (00:05) Unit 11. Course summary

Unit 1. WebSphere product family overview

Estimated time

01:00

Overview

This unit describes the products in the WebSphere product family and their relationship to WebSphere Application Server V9.

How you will check your progress

- Review questions

References

WebSphere Application Server V9 documentation in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

How to check online for course material updates



Note: If your classroom does not have internet access, ask your instructor for more information.

Instructions

1. Enter this URL in your browser:
ibm.biz/CloudEduCourses
2. Find the product category for your course, and click the link to view all products and courses.
3. Find your course in the course list and then click the link.
4. The wiki page displays information for the course. If the course has a corrections document, this page is where it is found.
5. If you want to download an attachment, such as a course corrections document, click the **Attachments** tab at the bottom of the page.


Comments (0)	Versions (1)	Attachments (1)	About
--------------	--------------	------------------------	-------
6. To save the file to your computer, click the document link and follow the prompts.

Figure 1-1. How to check online for course material updates

Unit objectives

- Describe the WebSphere family of products
- Describe the relationships between various products in the WebSphere family
- Describe the WebSphere Application Server V9 offerings
- Describe how WebSphere Application Server is used in cloud, on-premises, and hybrid cloud environments
- Describe the standards that are supported in this release

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-2. Unit objectives

Topics

- Platform and packaging
- Related WebSphere products and cloud-based offerings

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-3. Topics

1.1. Platform and packaging

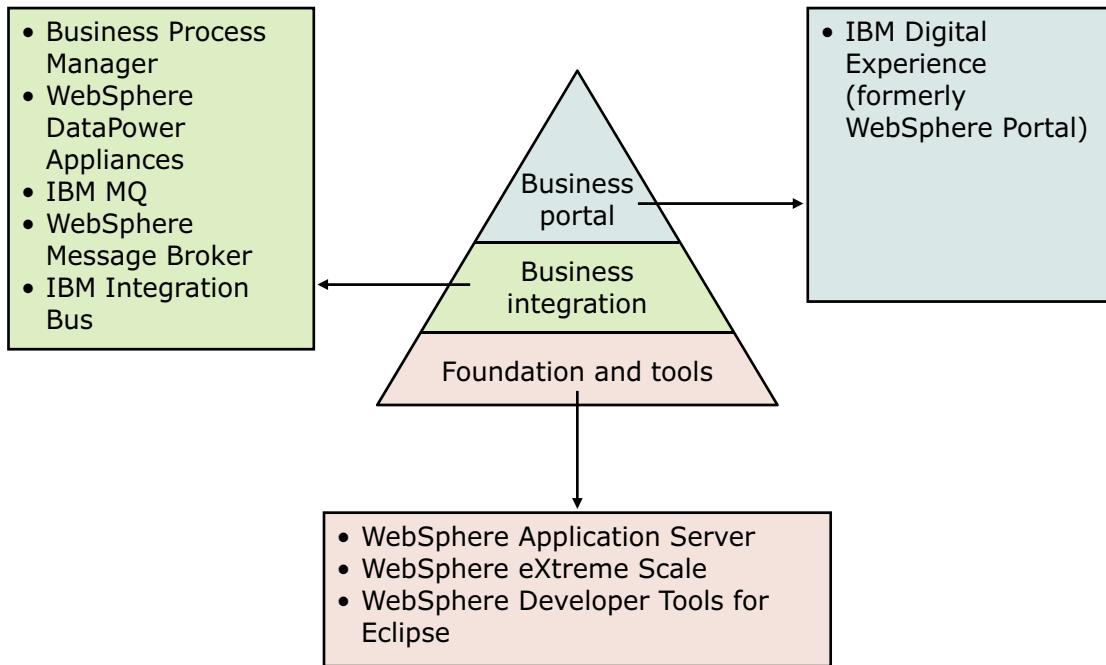
Platform and packaging

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 1-4. Platform and packaging

WebSphere software platform



WebSphere product family overview

© Copyright IBM Corporation 2016

Figure 1-5. WebSphere software platform

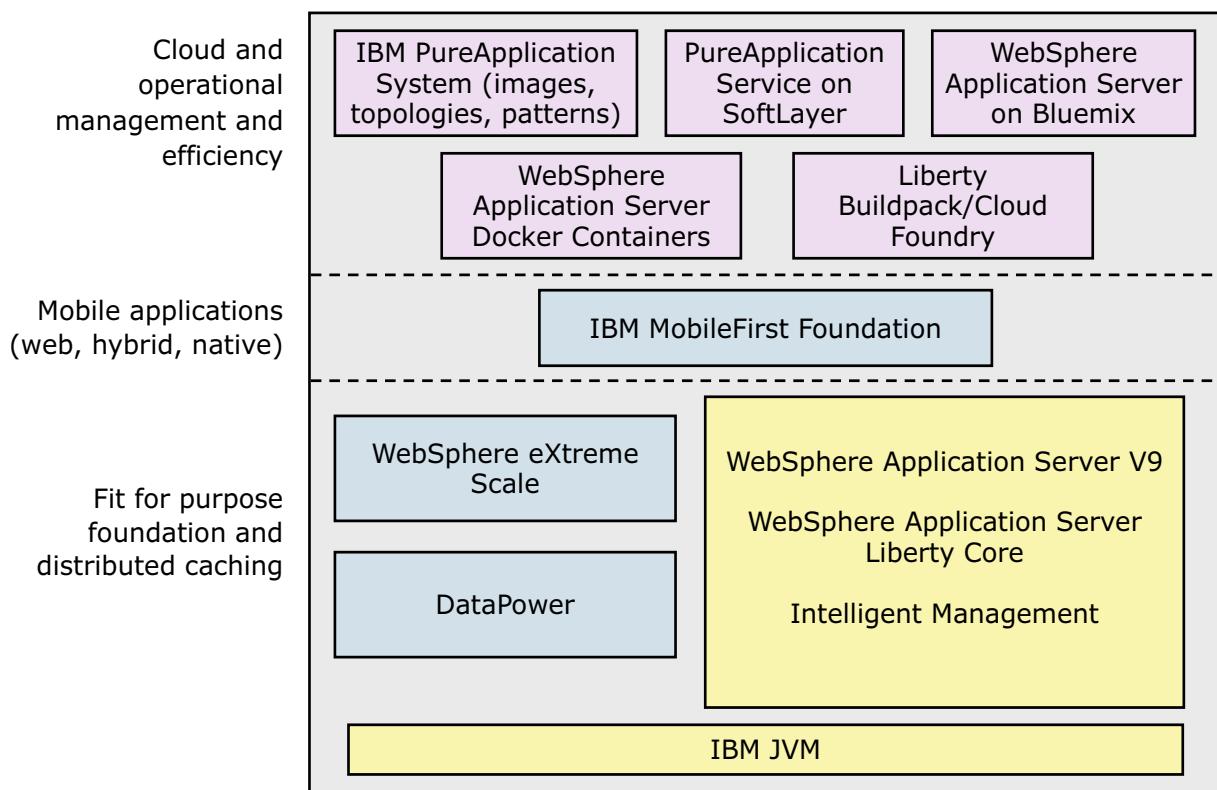
The WebSphere software platform consists of multiple products. These products can be grouped into three categories:

- Foundation and tools
- Business integration
- Business portals

As shown in the graphic, the three categories build upon each other in the order listed. WebSphere Application Server is the base product for all others that are listed in this slide. WebSphere Developer Tools for Eclipse is the primary development tool for all these products.



WebSphere application infrastructure: The big picture



WebSphere product family overview

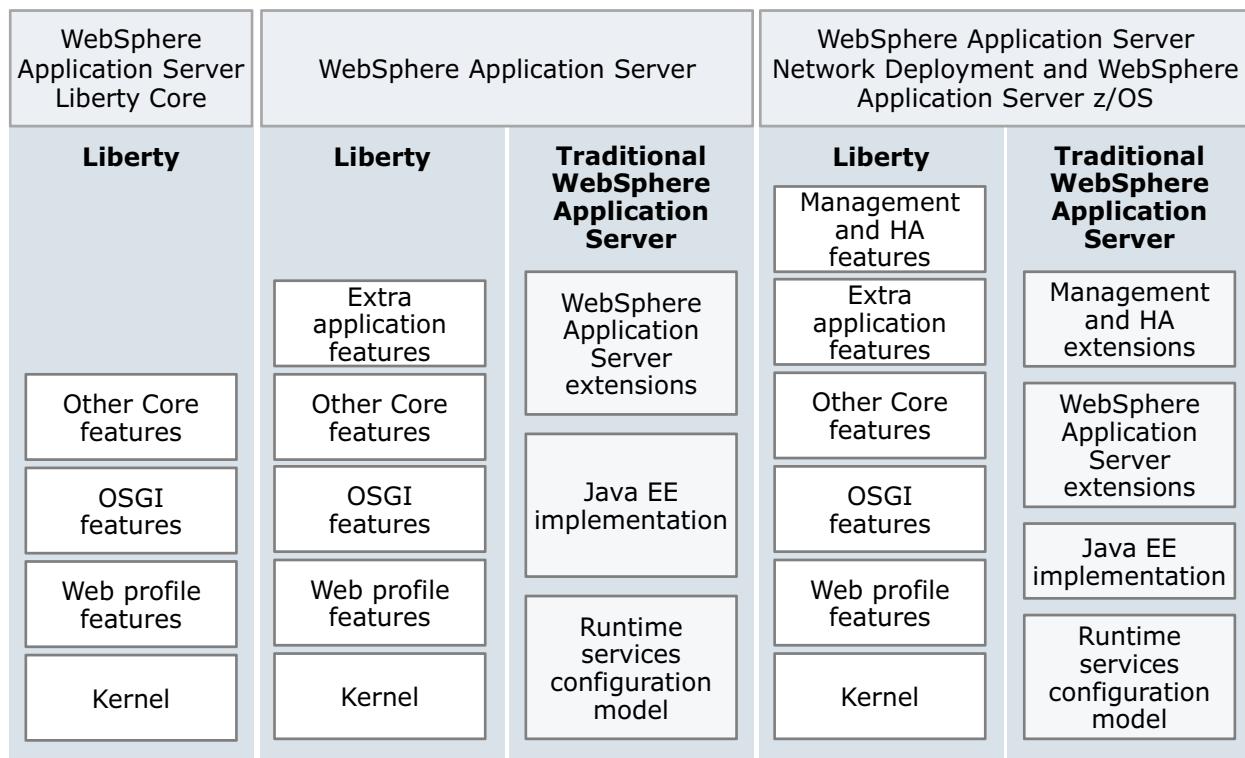
© Copyright IBM Corporation 2016

Figure 1-6. WebSphere application infrastructure: The big picture

WebSphere Application Server Version 9 delivers the core foundational requirements for the rest of the WebSphere Application Server portfolio of products. Built upon the IBM Java Virtual Machine (JVM), the Application Server provides the foundation for the WebSphere portfolio, including IBM PureApplication System, WebSphere eXtreme Scale, and the DataPower appliances.



WebSphere V9 profile by product: Quick view



WebSphere product family overview

© Copyright IBM Corporation 2016

Figure 1-7. WebSphere V9 profile by product: Quick view

IBM WebSphere Application Server V9 provides a range of flexible, secure, Java EE 7 runtime environments. It can handle everything from lightweight production projects to large enterprise deployments. You can use WebSphere Application Server on premises or across public, private, and hybrid clouds – and you can be up and running on a hosted cloud environment in minutes with IBM WebSphere Application Server on Cloud.

WebSphere Application Server V9, has three different runtimes:

1. WebSphere Application Server Liberty Core
2. WebSphere Application Server
3. WebSphere Application Server ND and WebSphere Application Server z/OS

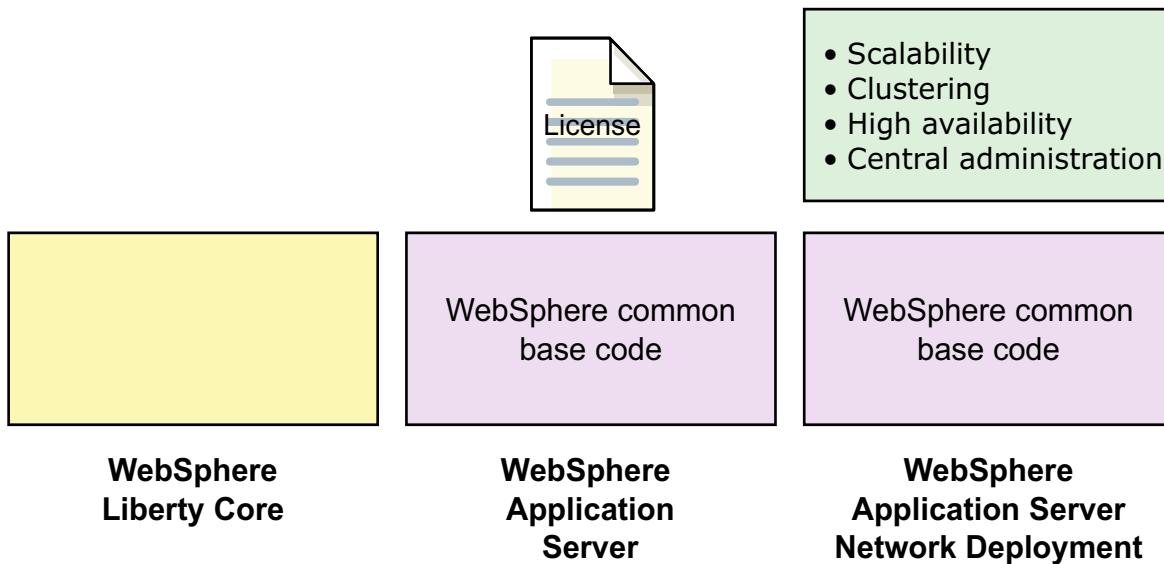
WebSphere Application Server Liberty Core is a lightweight, production runtime that is designed for rapid development and deployment of web and cloud-based applications.

WebSphere Application Server offers a fast, flexible, and secure Java application server runtime environment. It supports environments from single server and midsize configurations to large deployments that require web tier clustering over multiple application server instances.

WebSphere Application Server Network Deployment provides an advanced, flexible runtime environment for large-scale and mission-critical application deployments for open environments. It

offers near-continuous availability with advanced performance and management capabilities for mission-critical applications. WebSphere Application Server Network Deployment also provides a runtime for IBM z Systems and z/OS and enables prioritized workload management, advanced transactional integrity, horizontal and vertical scalability, and data and workload collocation.

WebSphere Application Server V9 packaging



[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-8. WebSphere Application Server V9 packaging

The image compares the basic differences in packaging between the WebSphere Liberty Core, WebSphere Application Server, and WebSphere Application Server Network Deployment editions. All three packages include the WebSphere common base code. The major difference between Liberty Core and WebSphere Application Server is the license. Liberty Core has a limited license for two processors; WebSphere Application Server has a license for unlimited processors. The Network Deployment product adds support for scalability, clustering, high availability, and central administration.

Product packaging comparison (1 of 2)

	WebSphere Application Server	WebSphere Application Server Network Deployment
Core Application Server	Stand-alone node	Multiple distributed nodes, with centralized administration
IBM HTTP Server web server plug-ins	✓	✓
Application client, update installer, installation factory, migration tools, IBM Support Assistant	✓	✓
Data Direct JDBC drivers	✓	✓
Liberty profile	✓	✓
Intelligent Management Pack		✓

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-9. Product packaging comparison (1 of 2)

This chart summarizes the WebSphere Application Server and Network Deployment packages for production use, and provides a side-by-side comparison of some of the important features. Detailed comparison information for the versions can be found at:

<http://www.ibm.com/software/webservers/appserv/wasfamily/compare.html>

Both versions include the core application server, IBM HTTP Server, web server plug-ins, application client, and Data Direct JDBC drivers.

The WebSphere Application Server package provides for deployment of a stand-alone node. In addition to a stand-alone node, the Network Deployment edition supports a managed or clustered multi-node environment with a central point of administration.

The Network Deployment edition also includes the Edge Components, IBM Tivoli Directory Server, and Tivoli Access Manager Server.

Product packaging comparison (2 of 2)

	WebSphere Application Server	WebSphere Application Server Network Deployment
Batch processing	✓	✓
Development and deployment tools	IBM WebSphere Application Server Developer Tools for Eclipse	IBM WebSphere Application Server Developer Tools for Eclipse
Edge components		✓
IBM Tivoli Directory Server (LDAP server)		✓
Tivoli Access Manager Server		✓

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-10. Product packaging comparison (2 of 2)

This chart summarizes the WebSphere Application Server and Network Deployment packages for production use, and provides a side-by-side comparison of some of the important features. Detailed comparison information for the versions can be found at:

<http://www.ibm.com/software/webservers/appserv/wasfamily/compare.html>

Both versions include the core application server, IBM HTTP Server, web server plug-ins, application client, and Data Direct JDBC drivers.

The WebSphere Application Server package provides for deployment of a stand-alone node. In addition to a stand-alone node, the Network Deployment edition supports a managed or clustered multi-node environment with a central point of administration.

The Network Deployment edition also includes the Edge Components, IBM Tivoli Directory Server, and Tivoli Access Manager Server.

Supported specifications in V9

Specification or API		
Java EE 7 certified	Java SE 8	Servlet 3.1
JSP 2.3	Portlet 2.0	SIP 1.1, 1.0
EJB 3.2	JDBC 4.1	JPA 2.1
JAXB 2.2	JAX-RS 2.0	JAX-RPC 1.1
JAXP 1.2	JAXR 1.0	JAX-WS 2.2
SAAJ 1.2, 1.3	JSF 2.2	SOAP 1.1, 1.2
StAX 1.0	UDDI 3.0	XML schema 1.0
JCA 1.7	WS-AT 1.0, 1.1	WS-BA 1.0, 1.1
WS-COOR 1.0, 1.1	WSDL 1.1	JSR 109 1.2
SDO 1.0	JMS 2.0	JavaMail 1.5
Java 2 Security	JAAS 2.0	JACC 1.5
JCE 1.0	CertPath 1.1	WS-Security 1.1
JAF 1.1	WS-I Attachments 1.0	WSIF
JNDI 1.0	JTA 1.1	And more

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-11. Supported specifications in V9

This chart lists some of the specifications and APIs supported in WebSphere Application Server V9. For a more exhaustive list of supported specifications and APIs, see the WebSphere documentation in the IBM Knowledge Center.

1.2. Related WebSphere products and cloud-based offerings

Related WebSphere products and cloud-based offerings

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 1-12. Related WebSphere products and cloud-based offerings

IBM WebSphere Application Server Developer Tools for Eclipse

- Efficient development and innovative features of WebSphere Application Server V9 available at no charge
- Provided by plug-ins from the Eclipse Marketplace that can be installed into an existing Eclipse environment to support development for WebSphere Application Server
- Lightweight development environment for the developer desktop
- Reduce testing effort and develop with confidence by using a runtime environment that is identical to the production runtime environment your applications eventually run on
- Available at:
<http://www.ibm.com/developerworks/downloads/ws/wasdevelopers/index.html>

WebSphere product family overview

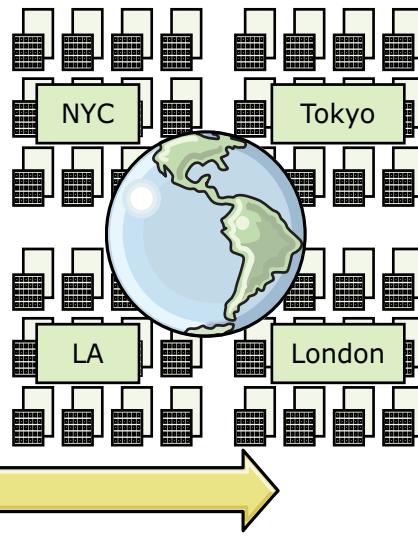
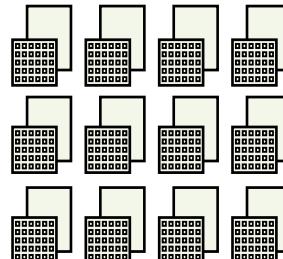
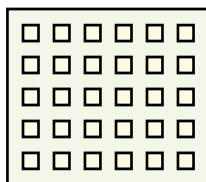
© Copyright IBM Corporation 2016

Figure 1-13. IBM WebSphere Application Server Developer Tools for Eclipse

WebSphere Application Server Version 9 delivers a no-charge offering to enable quick and easy developer access to the application server that is built on the same code base as the other WebSphere Application Server offerings.

WebSphere eXtreme Scale

A flexible framework for realizing high performance, scalable and data-intensive applications



- Can be used as a powerful cache
 - Scales from simple, in-process topologies to powerful, distributed topologies
- Can be used as a form of in-memory database
 - Manages application states
 - Scales to thousands of servers
 - Sometimes referred to as Distributed Application State Management
- Can be used as a platform for building powerful data grid applications

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-14. WebSphere eXtreme Scale

Applications and organizations can evolve as business needs and requirements change, which can require more scalability or the use of some of the other advanced features that eXtreme Scale provides.

What is IBM WebSphere eXtreme Scale? IBM WebSphere eXtreme Scale is a flexible framework for realizing high-performance, scalable, and data-intensive applications. It is a single map addressable space of virtualized JVM heap spaces. The first use of WebSphere eXtreme Scale is a power cache for your applications. WebSphere eXtreme Scale can move that collection of application caches into a grid that is highly available, elastic, and self-healing. The second use of WebSphere eXtreme Scale is a form of in-memory database. The third use of WebSphere eXtreme Scale is a form of redundancy across data centers.

WebSphere eXtreme Scale dynamically caches, partitions, replicates, and manages application data and business logic across multiple servers. WebSphere eXtreme Scale does massive volumes of transaction processing with high efficiency and linear scalability, and provides qualities of service such as transactional integrity, high availability, and predictable response times. WebSphere eXtreme Scale can be used in different ways. It can be used as a powerful cache, as a form of an in-memory database processing space to manage application state, or as a platform for building powerful extreme transaction processing (XTP) applications.

Some editions of WebSphere Application Server come with an entitlement to use WebSphere eXtreme Scale.

WebSphere supports the entire hybrid cloud landscape



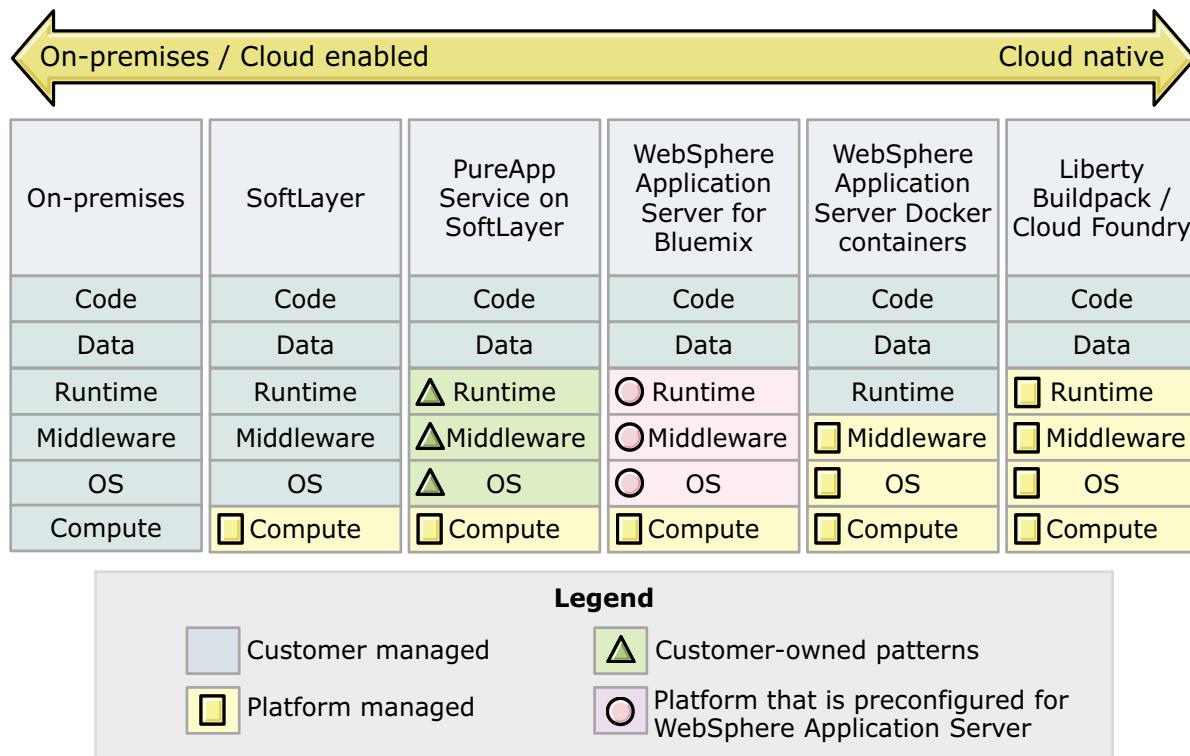
[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-15. WebSphere supports the entire hybrid cloud landscape

WebSphere Application Server is supported across the entire hybrid cloud landscape. WebSphere Application Server is packaged and delivered as a traditional on-premises solution. It can also be delivered as other solutions, ranging from a service on Bluemix to a Docker file available on GitHub.

Comparison of cloud features



WebSphere product family overview

© Copyright IBM Corporation 2016

Figure 1-16. Comparison of cloud features

WebSphere Application Server cloud features vary widely depending upon whether the solution is on-premises or completely native on the cloud.

The customer completely manages an on-premises solution, from the compute nodes to the application code. With a SoftLayer solution, IBM provides the compute node and the customer manages the remainder of the stack.

PureApplication Service on SoftLayer provides a mix of customer managed assets, customer owned patterns, and platform managed compute nodes. In this solution, WebSphere Application Server can be deployed as a virtual system pattern.

IBM's Bluemix platform provides the WebSphere Application Server for Bluemix service. With this service, you can set up a preconfigured WebSphere Application Server Liberty, Traditional Network Deployment, or Traditional WebSphere Java EE instance in a hosted cloud environment on Bluemix. You manage the code and data, while IBM manages the Bluemix platform and compute nodes.

WebSphere Application Server Docker containers and Liberty Buildpack are both managed by their respective platforms, while you manage the code and data.

Pattern-based deployments on SoftLayer

- Dedicated and isolated hybrid-enabled cloud environment that is built to run enterprise-grade applications
- Using the expertise of patterns, deployments of complex enterprise workloads become fast, repeatable, and reliable
- Supports middleware pattern deployments that can run on the IBM Cloud:
 - WebSphere Application Server
 - WebSphere Application Server Network Deployment
 - Liberty Core
 - WebSphere eXtreme Scale

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-17. Pattern-based deployments on SoftLayer

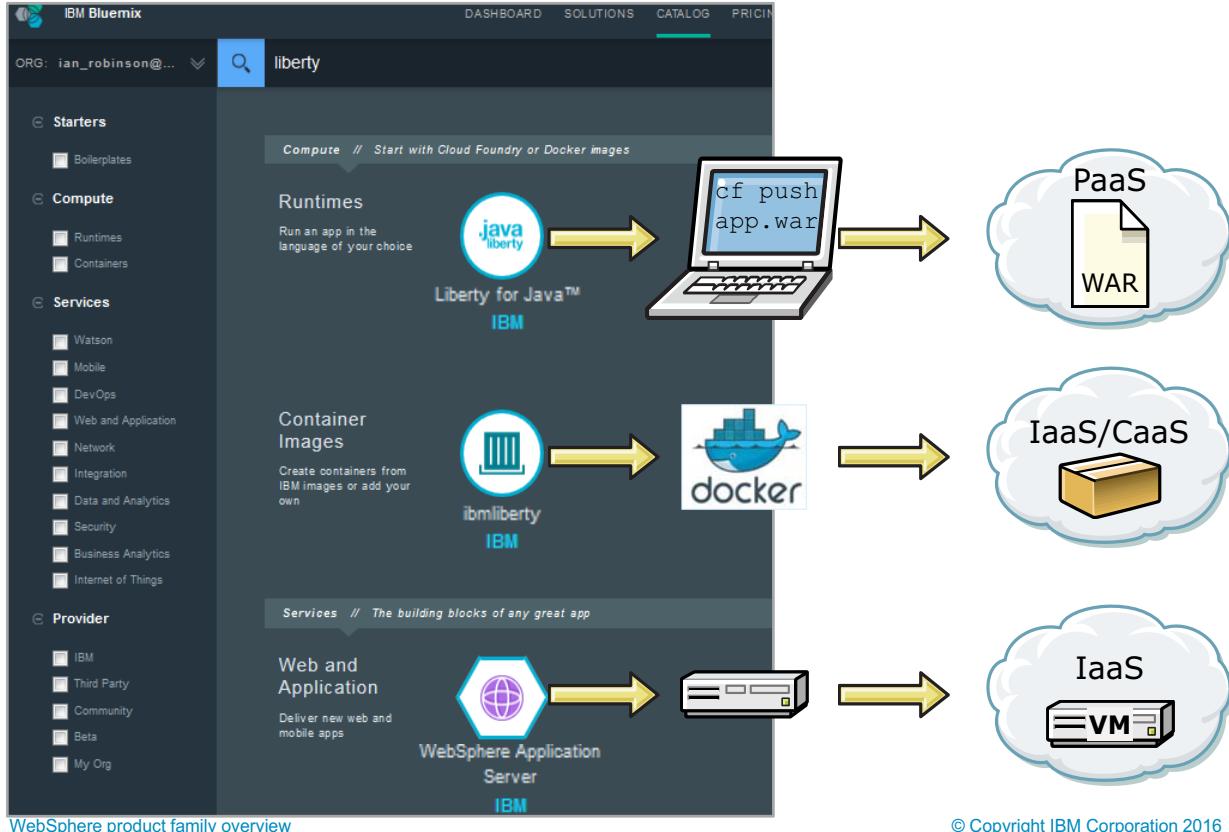
IBM PureApplication Service on SoftLayer is a dedicated and isolated hybrid-enabled cloud environment that is built to run enterprise-grade applications. PureApplication Service helps you extend your applications to hybrid cloud simply and quickly. Using the expertise of patterns, deployments of complex enterprise workloads become fast, repeatable, and reliable.

PureApplication System supports IBM middleware product pattern deployments, including:

- WebSphere Application Server Liberty Core
- WebSphere Application Server
- WebSphere Application Server Network Deployment
- WebSphere eXtreme Scale
- DataPower appliances



IBM Bluemix and WebSphere



WebSphere product family overview

© Copyright IBM Corporation 2016

Figure 1-18. IBM Bluemix and WebSphere

WebSphere is the Java runtime for the IBM Cloud. For next generation “born on the cloud” apps:

- Simple apps that are pushed into Bluemix’s Instant Runtimes are wrapped in a Liberty Java container.
- Java apps that run in Docker containers are running with the specific set of Liberty features that they need.
- Apps that evolved from an older design that predates cloud-native thinking are best serviced through the WebSphere Application Server for Bluemix service. In this case, Bluemix provides the IaaS layer for WebSphere.

IBM Bluemix: Local and dedicated



Dedicated



Local

- Your own cloud with physically isolated hardware
- Use either VPN or Direct Link technology to securely connect to your enterprise
- Available in over 25 IBM Cloud data centers around the world

- Delivered as-a-service on your hardware or on a pre-integrated converged infrastructure
- Use your existing mainframes, SOA, processes, and data on a cloud platform
- Relay technology keeps your Bluemix Local deployment on the same update and release cycle as the public platform

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-19. IBM Bluemix: Local and dedicated

Bluemix Dedicated provides physically isolated hardware in an IBM data center. Single-tenant and provisioned on a combination of bare metal and virtual machines, your Bluemix environment is created just for you. With the syndicated catalog, you can power your apps and services with a combination of dedicated compute and services, along with services from the public Bluemix catalog.

IBM not only delivers Bluemix and the platform services layer as-a-service (PaaS), but it can also operate the underlying infrastructure layer as-a-service (IaaS) as Bluemix Local. This combination enables companies to have a full-stack cloud experience. Bluemix Local is delivered behind the firewall and integrated with IBM's market-leading security intelligence and edge protection technology that some companies use to manage billions of security incidents a day. Additionally, Bluemix platform patches and security fixes are delivered swiftly through Relay.

WebSphere Application Server for Bluemix



- Move applications in just minutes
- Deploy the same traditional WebSphere Application Server, WebSphere Application Server Network Deployment, WebSphere Liberty Server, or Liberty Collective in the cloud as in on-premises data center
- Reuse existing WebSphere skills and tools, such as the WebSphere Integrated Management Console, in the cloud as in your data center on existing WebSphere installations

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-20. WebSphere Application Server for Bluemix

WebSphere Application Server for Bluemix is a service that facilitates quick setup on a preconfigured WebSphere Application Server Liberty, Traditional Network Deployment, or Traditional WebSphere Java EE instance in a hosted cloud environment on Bluemix.

You are given a familiar WebSphere administration experience and have full access to the underlying operating system. You can reuse your existing scripts and tweak the system as necessary to make it work with your own, or third-party, frameworks. The Admin Center and Admin Consoles are provided to administer your WebSphere Application Server Liberty, ND, or Traditional service, just like your on-premises WebSphere configurations.

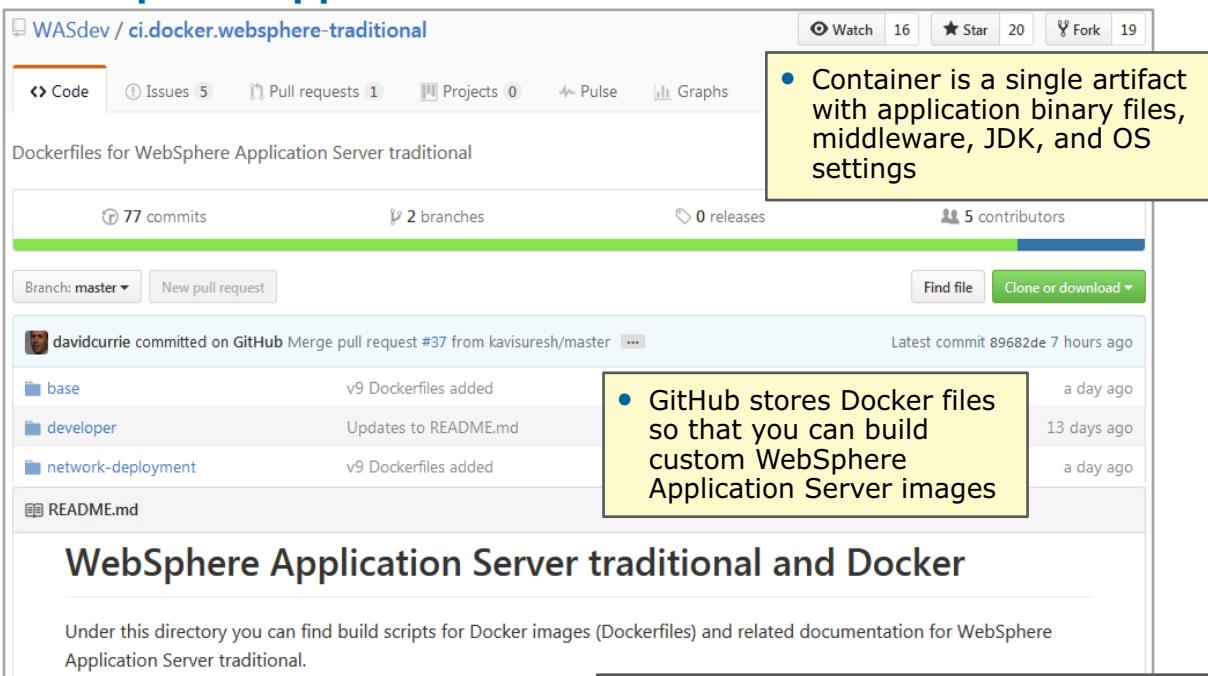
The service is provided through Bluemix as a plan.

The WebSphere Application Server for Bluemix Network Deployment Plan consists of a WebSphere Application Server Network Deployment cell environment with two or more virtual machines. The first virtual machine contains the Deployment Manager and IBM HTTP Server, and the remaining virtual machines contain custom nodes (node agents) that are federated to the Deployment Manager.

The WebSphere Application Server for Bluemix Liberty Core Plan includes the use of a Liberty Collective. The Liberty Collective is an administrative domain for a group of Liberty profiles (servers) and consists of two or more virtual machines.

IBM Training 

WebSphere Application Server and Docker containers



Container is a single artifact with application binary files, middleware, JDK, and OS settings

GitHub stores Docker files so that you can build custom WebSphere Application Server images

WebSphere Application Server traditional and Docker

Under this directory you can find build scripts for Docker images (Dockerfiles) and related documentation for WebSphere Application Server traditional.

- WebSphere Application Server for Developers traditional
- WebSphere Application Server traditional
 - Base
 - Network Deployment

Deploy containers to container services

Decrease development and deployment time and cost

WebSphere product family overview

© Copyright IBM Corporation 2016

Figure 1-21. WebSphere Application Server and Docker containers

Docker containers bundle application binary files, middleware, the JDK, and operating system settings into a single artifact, providing deployment consistency and portability from development to test to production. IBM makes several WebSphere Liberty containers available on the Docker hub, including a Java EE 6 and Java EE 7 web profile and full Java EE 7, each building on a common Liberty kernel layer. You can then build a fully customized Java EE Docker container by starting with just the Liberty kernel and the specific list of features in a `server.xml` configuration.

You can also use Docker files that are available on GitHub to build custom images that run WebSphere Application Server with customized license entitlements. WebSphere Application Server Developer Tools V9 adds support so that you can code and test your applications locally or remotely in Docker while you develop in their Eclipse environment. You can choose to deploy these containers to the IBM Container Service, Docker Datacenter, and other container services.

WebSphere Application Server V9 standards and changes

- Release 9 is Java Enterprise Edition 7 certified
- Java Platform, Standard Edition 8 is the default for traditional WebSphere Application Server V9
- Updated system requirements and operating support matrix
- Many components are updated to use more modern components: IBM HTTP Server, Admin Console, open source libraries
- Security hardening for Admin Console
- IBM HTTP Server based to Apache 2.4 in Release 9
- Default Jython is version 2.7 (older versions available but deprecated)
- Only Java 8 64-bit (no 32-bit) JDK
- Installation requires IBM Installation Manager 1.8.5 or higher
- IBM Knowledge Center includes release details in the document *What is new in WebSphere Application Server traditional*

WebSphere product family overview

© Copyright IBM Corporation 2016

Figure 1-22. WebSphere Application Server V9 standards and changes

Regarding changes in operating support, Ubuntu KVM 14.04 and 16.04 are added as supported operating systems. Windows 7 is dropped. The minimum supported version of Windows is 8.1.

Express is no longer used. Express is now replaced with Liberty.

For more information, see the following websites:

- Documentation for changes in the release:
https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/com.ibm.websphere.base.doc/ae/welc_newinrelease.html
- WebSphere Application Server product offerings for supported operating systems:
https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/com.ibm.websphere.installation.base.doc/ae/cins_offerings.html

Unit summary

- Describe the WebSphere family of products
- Describe the relationships between various products in the WebSphere family
- Describe the WebSphere Application Server V9 offerings
- Describe how WebSphere Application Server is used in cloud, on-premises, and hybrid cloud environments
- Describe the standards that are supported in this release

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-23. Unit summary

Review questions

1. What is the main difference between the WebSphere Application Server and WebSphere Application Server ND editions?
2. If you want to deploy a single application with low transaction volume, which WebSphere Application Server edition do you need?
3. True or False: WebSphere Application Server for Bluemix is provided in WAR format, which includes WebSphere Application Server and the operating system.
4. True or False: Support is included for mixed WebSphere Application Server releases in a cell.



[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-24. Review questions

Write your answers here:

- 1.
- 2.
- 3.
- 4.

Review answers

1. What is the main difference between the WebSphere Application Server and WebSphere Application Server ND editions?
 The answer is: WebSphere Application Server provides for the deployment of a stand-alone node. WebSphere Application Server ND provides for a managed or clustered multi-node environment with a central point of administration.
2. If you want to deploy a single application with low transaction volume, which WebSphere Application Server edition do you need?
 The answer is: You can use any edition to deploy a single application with low transaction volume.
3. True or False: WebSphere Application Server for Bluemix is provided in WAR format, which includes WebSphere Application Server and the operating system.
 The answer is False. WebSphere Application Server for Bluemix can “spin up” preconfigured WebSphere Application Server Liberty, Traditional Network Deployment, or Traditional WebSphere Java EE instances.
4. True or False: Support is included for mixed WebSphere Application Server releases in a cell.
 The answer is True.

[WebSphere product family overview](#)

© Copyright IBM Corporation 2016

Figure 1-25. Review answers



Unit 2. WebSphere Application Server architecture: Stand-alone

Estimated time

01:00

Overview

This unit provides an architectural overview of WebSphere Application Server V9 at run time.

How you will check your progress

- Review questions

References

WebSphere Application Server V9 documentation in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Describe a typical e-business application architecture
- Describe the architectural differences between WebSphere Application Server packages
- Describe what is running in a WebSphere Application Server node
- Describe the architectural implications of the web server plug-in
- Describe the use of Java Database Connectivity (JDBC) providers and data sources

Topics

- Architecture runtime
- Architecture administration
- Profiles

WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-2. Topics

2.1. Architecture runtime

This topic provides information about architecture runtime.

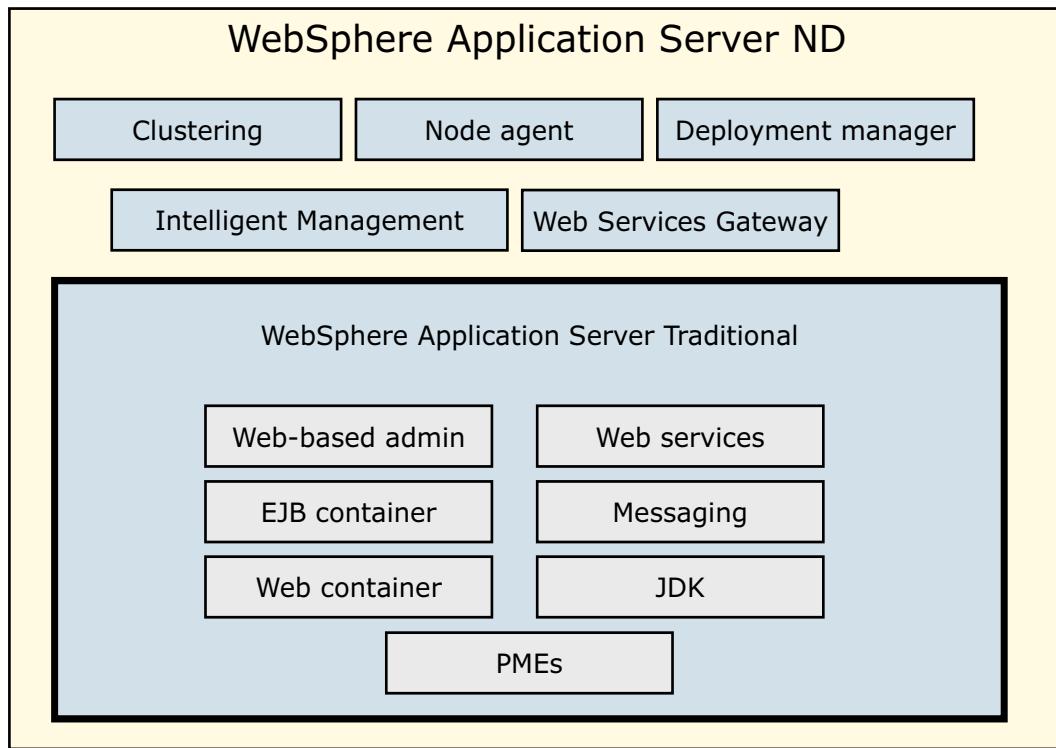
Architecture runtime

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-3. Architecture runtime

Version 9 packaging



WebSphere Application Server architecture: Stand-alone

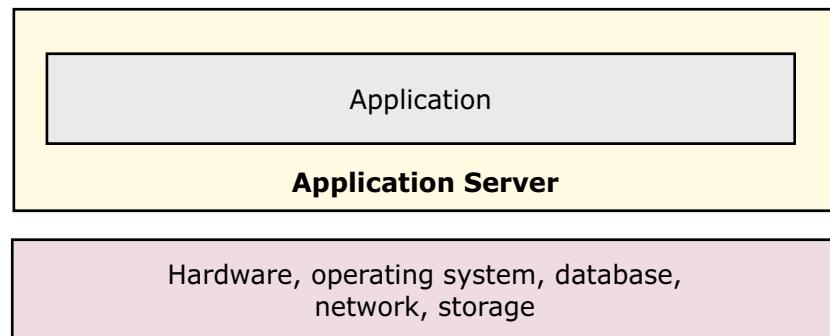
© Copyright IBM Corporation 2016

Figure 2-4. Version 9 packaging

In this unit, the focus is on the single-server version of WebSphere Application Server traditional, shown against the blue background in this diagram.

WebSphere Application Server basics

- WebSphere Application Server
 - Is a platform on which Java based business applications run
 - Is an implementation of the Java Platform, Enterprise Edition (Java EE) specification
 - Provides services (database connectivity, threading, workload management) that the business applications can use



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-5. WebSphere Application Server basics

This diagram illustrates the differences between the application, the application server, and the hardware and operating system layers. WebSphere Application Server is a platform on which Java-based business applications run and is an implementation of the Java Platform, Enterprise Edition specification. It provides services (database connectivity, threading, workload management) that the business applications can use.

WebSphere architecture runtime (1 of 10)



Legend	→ HTTP or HTTPS→ RMI/IOP	→ SOAP	- - → JDBC
---------------	-----------------	----------------	--------	------------

WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-6. WebSphere architecture runtime (1 of 10)

This diagram is the first of a series of ten diagrams that build in complexity over the next ten pages. The diagrams illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

The main element is the application server, a Java process that encapsulates many services, including the containers, where business logic runs. If you are familiar with Java EE, you recognize the web container and the EJB container. The web container runs servlets and JavaServer Pages (JSPs), both of which are Java classes that generate markup that a web browser can view. Traffic into and out of the web container travels through the embedded IBM HTTP Server. While servlets and JSPs can act independently, they most commonly make calls to EJBs to run business logic or access data. EJBs, which run in the EJB container, are easily reusable Java classes. They most commonly communicate with a relational database or other external source of application data, either returning that data to the web container or changing the data on behalf of the servlet or JSP.

The JMS messaging engine is built into the application server. This messaging engine is pure Java. JMS destinations, which are known as queues and topics, provide asynchronous messaging services to the code that runs inside the containers. JMS is covered in more depth later in this course.

As you see in more detail later on, the web services engine provides the ability for application components to be exposed as web services, which can be accessed by using SOAP.

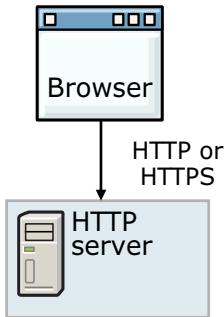
Several other services are run within the application server, including the dynamic cache, data replication, security, and others. You look at these topics later in the course.

In addition, some important components exist outside of the application server process.

WebSphere Application Server also provides a plug-in for HTTP servers that determines the HTTP traffic that WebSphere intends to handle, and routes the requests to the appropriate server. The plug-in is also a critical player in workload management of HTTP requests, as it can distribute the load to multiple application servers, and steer traffic away from servers that are not available. It also reads its configuration from a special XML file.

1 of 10: The browser is the main interaction mechanism for users.

WebSphere architecture runtime (2 of 10)



Legend	→ HTTP or HTTPS→ RMI/IOP	→ SOAP	- - → JDBC
--------	-----------------	----------------	--------	------------

WebSphere Application Server architecture: Stand-alone

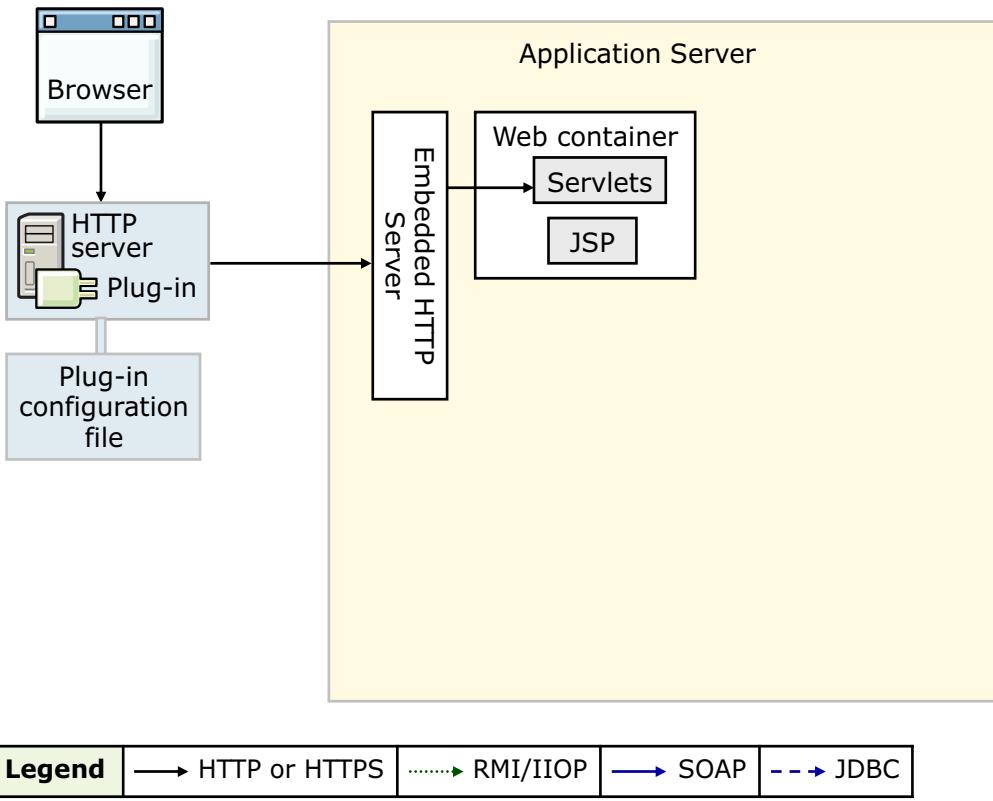
© Copyright IBM Corporation 2016

Figure 2-7. WebSphere architecture runtime (2 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

2 of 10: A browser communicates with a web server (HTTP server).

WebSphere architecture runtime (3 of 10)



Legend	→ HTTP or HTTPS→ RMI/IOP	→ SOAP	- - → JDBC
---------------	-----------------	----------------	--------	------------

WebSphere Application Server architecture: Stand-alone

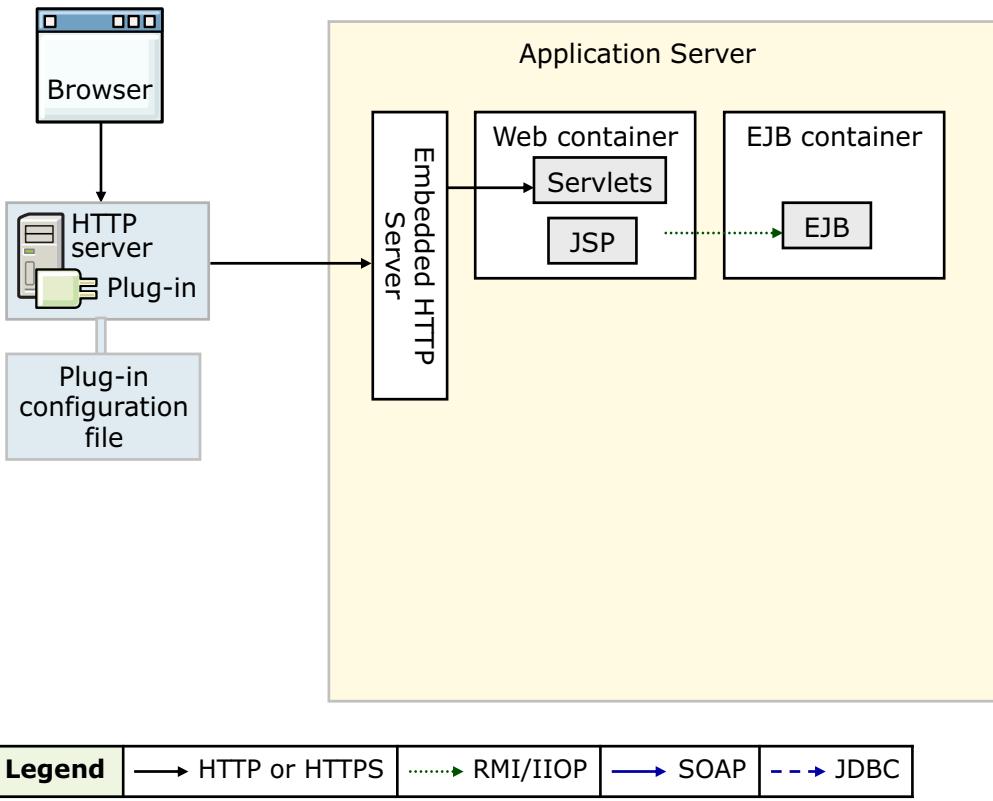
© Copyright IBM Corporation 2016

Figure 2-8. WebSphere architecture runtime (3 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

3 of 10: The way the request gets into the WebSphere Application Server is from the HTTP server plug-in that is loaded with the HTTP server. This request is forwarded to the embedded HTTP server within the application server. The embedded server forwards the request into the web container to either a servlet or a JSP.

WebSphere architecture runtime (4 of 10)



Legend	→ HTTP or HTTPS→ RMI/IOP	→ SOAP	- - → JDBC
---------------	-----------------	----------------	--------	------------

WebSphere Application Server architecture: Stand-alone

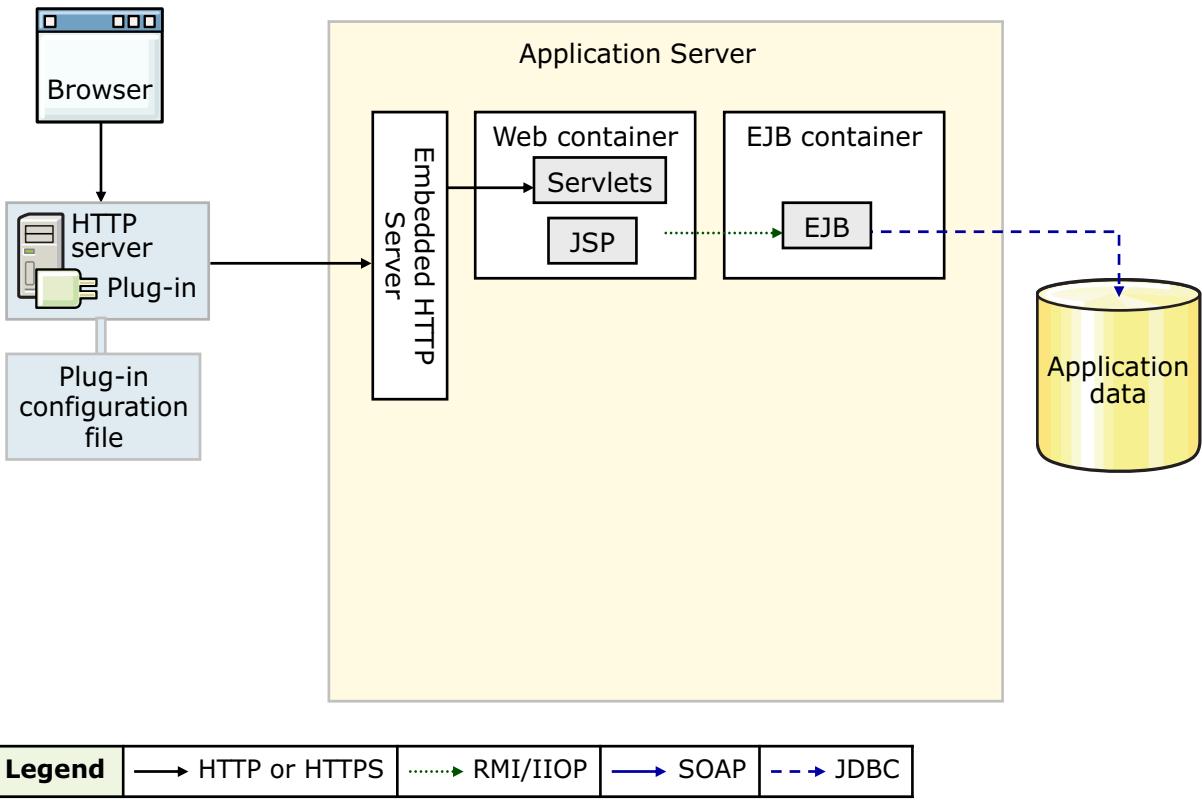
© Copyright IBM Corporation 2016

Figure 2-9. WebSphere architecture runtime (4 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

4 of 10: If these servlets or JSPs access distributed business logic or a database, the Java EE way to accomplish it is through EJBs within the EJB container.

WebSphere architecture runtime (5 of 10)



WebSphere Application Server architecture: Stand-alone

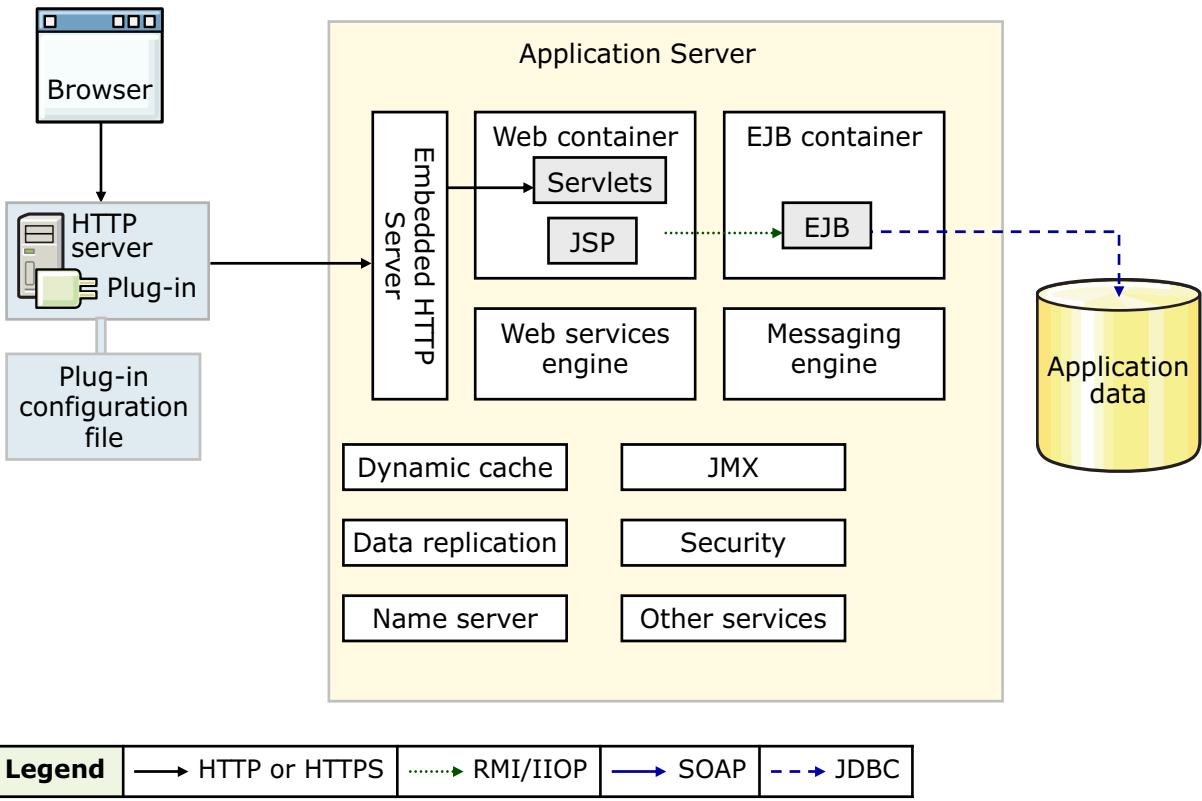
© Copyright IBM Corporation 2016

Figure 2-10. WebSphere architecture runtime (5 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

5 of 10: EJBs (entity in this case) can communicate with the database to store, retrieve, query, and delete data. JDBC is one way that this communication can occur.

WebSphere architecture runtime (6 of 10)



Legend	→ HTTP or HTTPS→ RMI/IOP	→ SOAP	- - → JDBC
---------------	-----------------	----------------	--------	------------

WebSphere Application Server architecture: Stand-alone

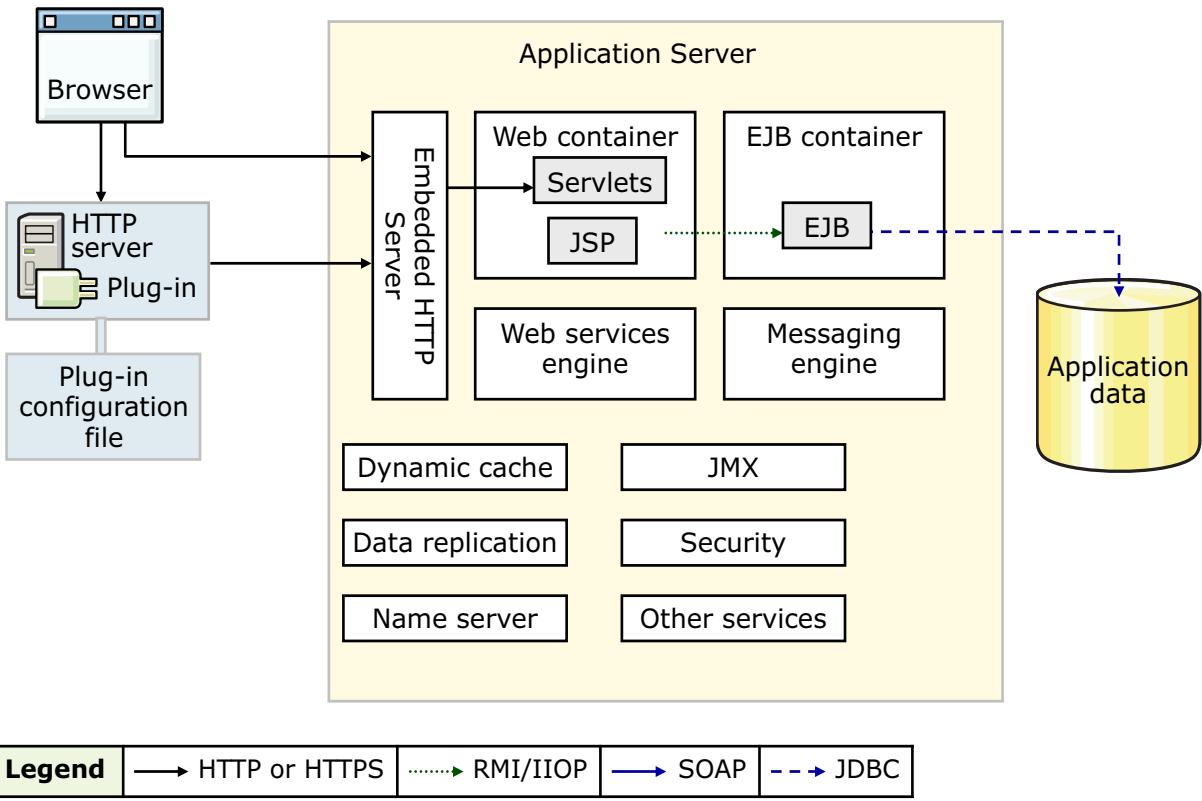
© Copyright IBM Corporation 2016

Figure 2-11. WebSphere architecture runtime (6 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

6 of 10: Many other services are provided within WebSphere Application Server. Some of those services are depicted here.

WebSphere architecture runtime (7 of 10)



WebSphere Application Server architecture: Stand-alone

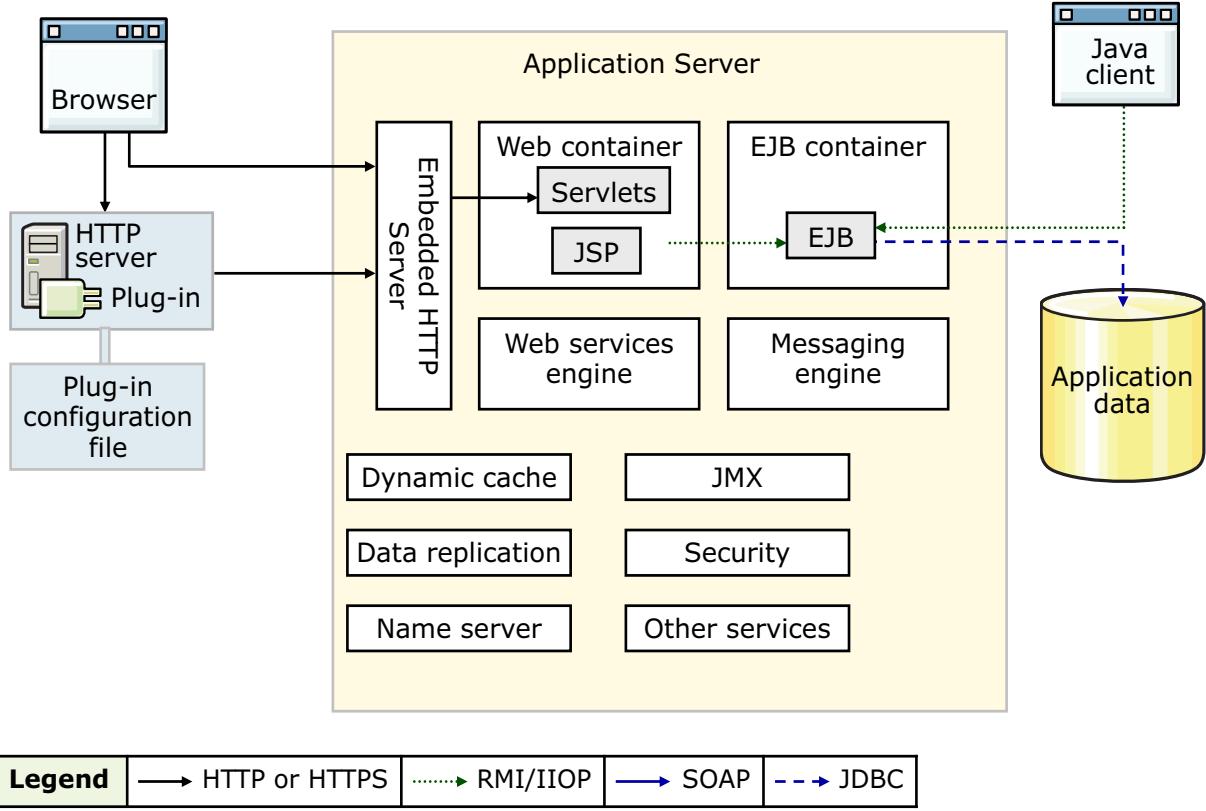
© Copyright IBM Corporation 2016

Figure 2-12. WebSphere architecture runtime (7 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

7 of 10: The browser can communicate directly with the embedded HTTP server (bypassing the external web server); use this direct communication only for testing and development purposes. In this way, you access your application servers in many of the lab exercises.

WebSphere architecture runtime (8 of 10)



WebSphere Application Server architecture: Stand-alone

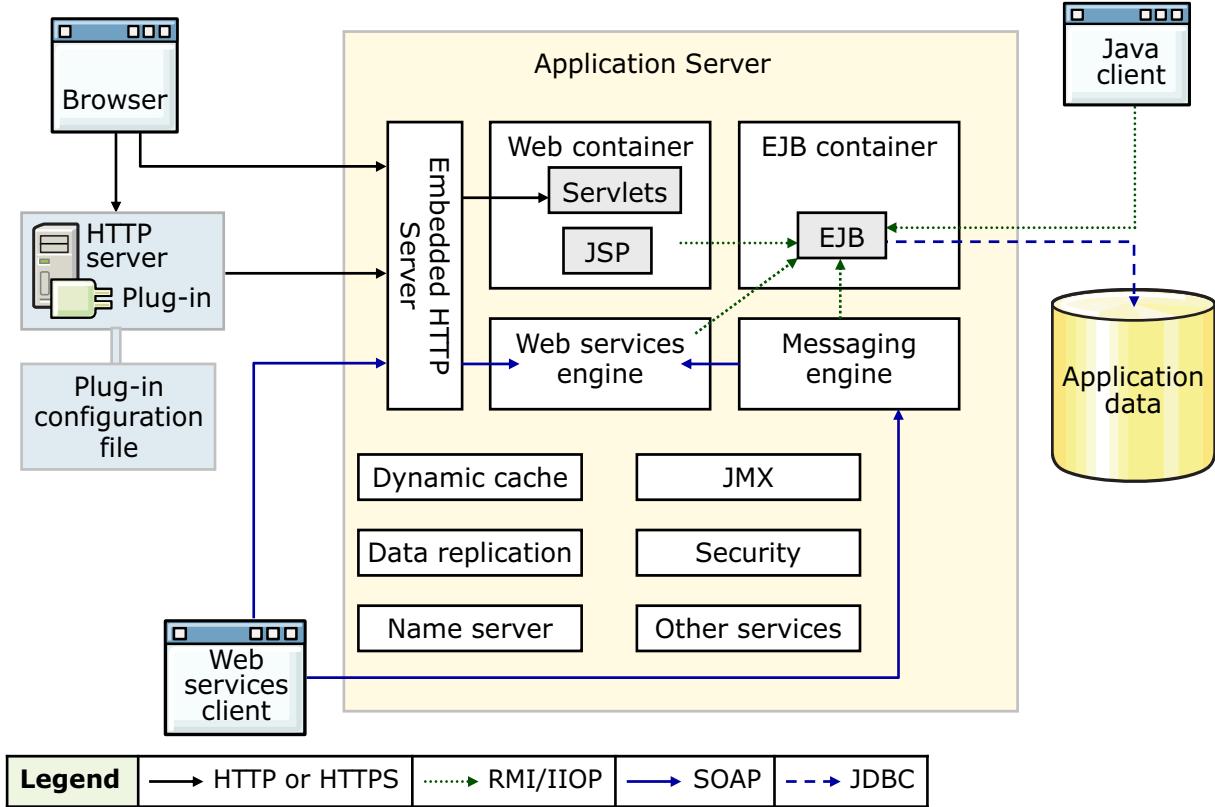
© Copyright IBM Corporation 2016

Figure 2-13. WebSphere architecture runtime (8 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

8 of 10: Browsers are not the only clients; a pure Java client can access EJBs directly through RMI/IOP.

WebSphere architecture runtime (9 of 10)



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

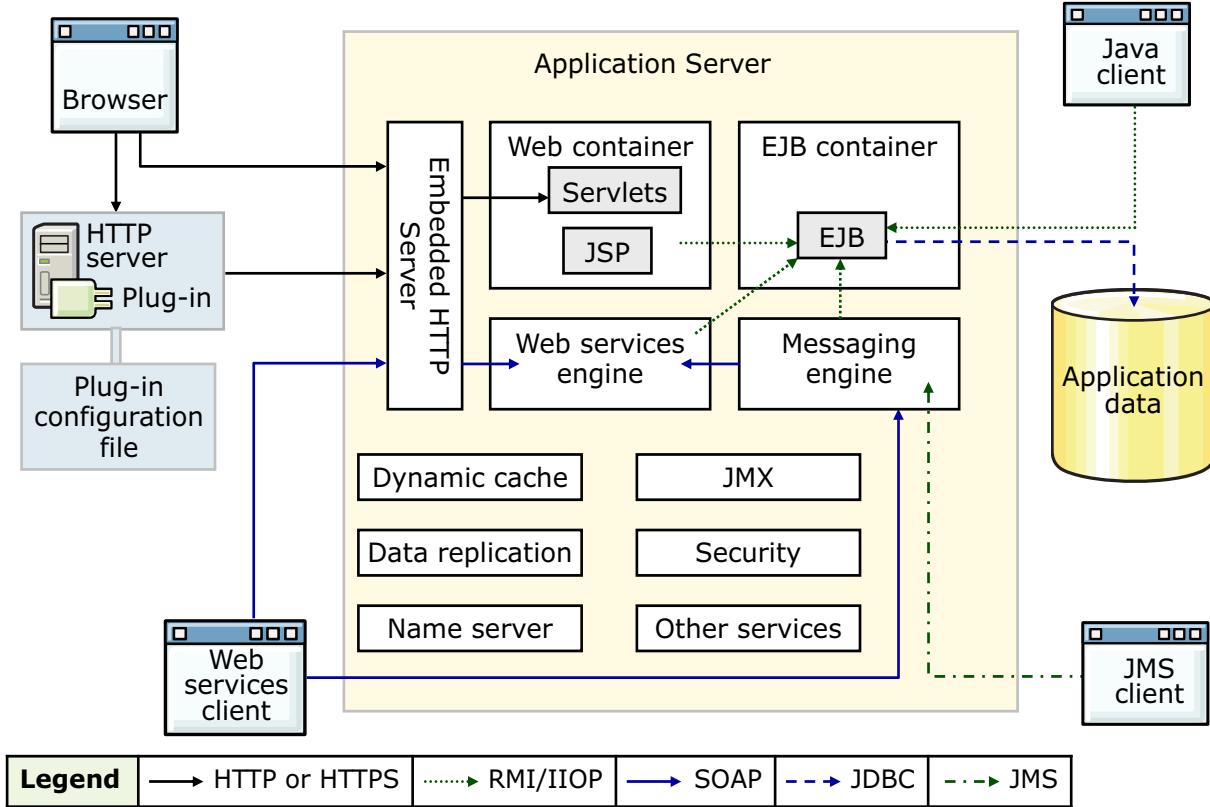
Figure 2-14. WebSphere architecture runtime (9 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

9 of 10: Web services clients can also access your application server. This communication occurs in two ways:

- Through SOAP over HTTP and passing through the embedded HTTP server
- Through SOAP over JMS communicating directly to the messaging engine within the application server

WebSphere architecture runtime (10 of 10)



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

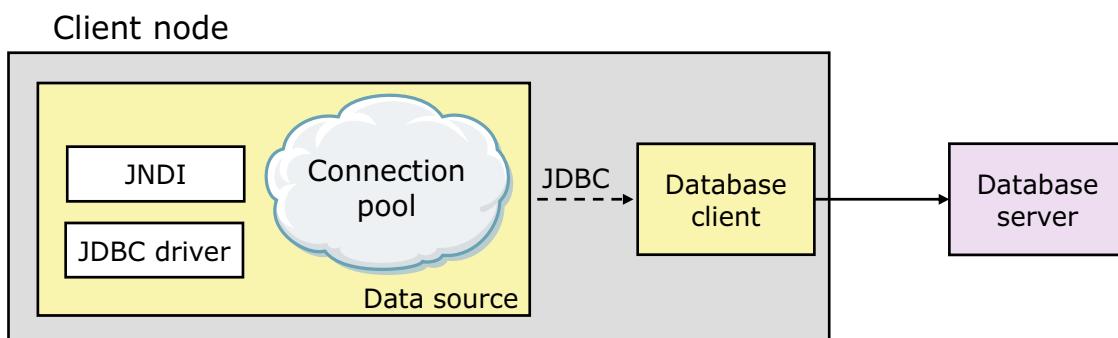
Figure 2-15. WebSphere architecture runtime (10 of 10)

This diagram is part of a series of diagrams that illustrate the basic architecture of WebSphere Application Server, including several of the larger components.

10 of 10: Finally, you can use a JMS client to communicate directly with the messaging engine.

JDBC providers

- Provide the JDBC driver implementation for database access
 - Type 2 JDBC drivers (thick): Require the database client software on the client node to connect to the database server
 - Type 3 JDBC drivers (net protocol): Require server-side code to map net protocol to native database
 - Type 4 JDBC drivers (native protocol): Connect directly to the database by using its native protocol
- XA drivers support transaction recovery



[WebSphere Application Server architecture: Stand-alone](#)

© Copyright IBM Corporation 2016

Figure 2-16. JDBC providers

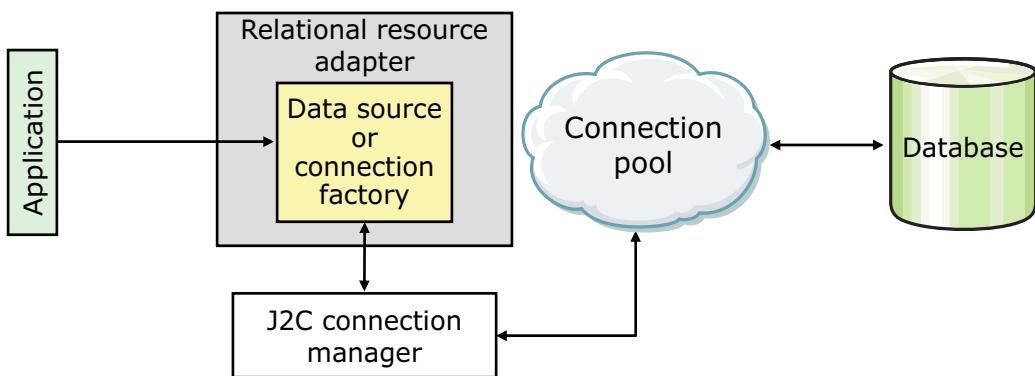
To access a database from an application server, a JDBC provider is necessary. The JDBC provider allows access to the database through a JDBC driver implementation for database access. Three different JDBC driver implementations are available with the application server.

- Type 2 JDBC drivers, sometimes known as “thick”, require the database client software on the client node to connect to the database server.
- Type 3 JDBC drivers (net protocol) require server-side code to map net protocol to the native database.
- Type 4 JDBC drivers (native protocol) connect directly to the database by using its native protocol.

XA drivers support transaction recovery.

Data sources

- Data sources can improve performance and portability for database access
 - Standard and XA data sources
- Two parts provide connection pooling:
 - J2C connection manager
 - Relational resource adapter
- Connection factories are similar to data sources
 - Typically connect to external resources that are not databases
- WebSphere Application Server Version 4.0 data sources are deprecated in Version 9



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-17. Data sources

Rather than having the JDBC drivers communicate directly with the database, the communication is abstracted into a data source.

2.2. Administration architecture

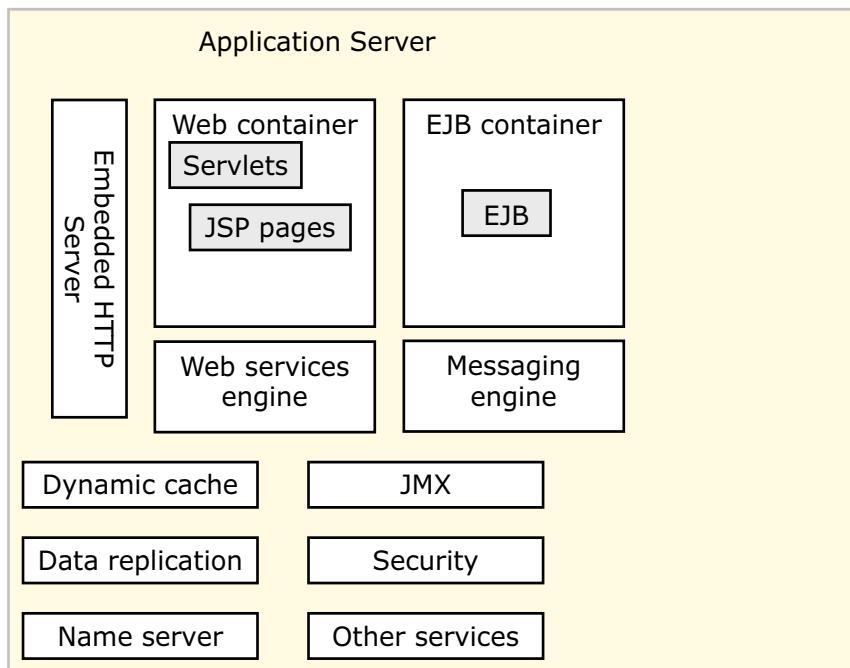
This topic provides information about architecture administration.

Administration architecture

© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-18. Administration architecture

WebSphere administration architecture (1 of 4)



[WebSphere Application Server architecture: Stand-alone](#)

© Copyright IBM Corporation 2016

Figure 2-19. WebSphere administration architecture (1 of 4)

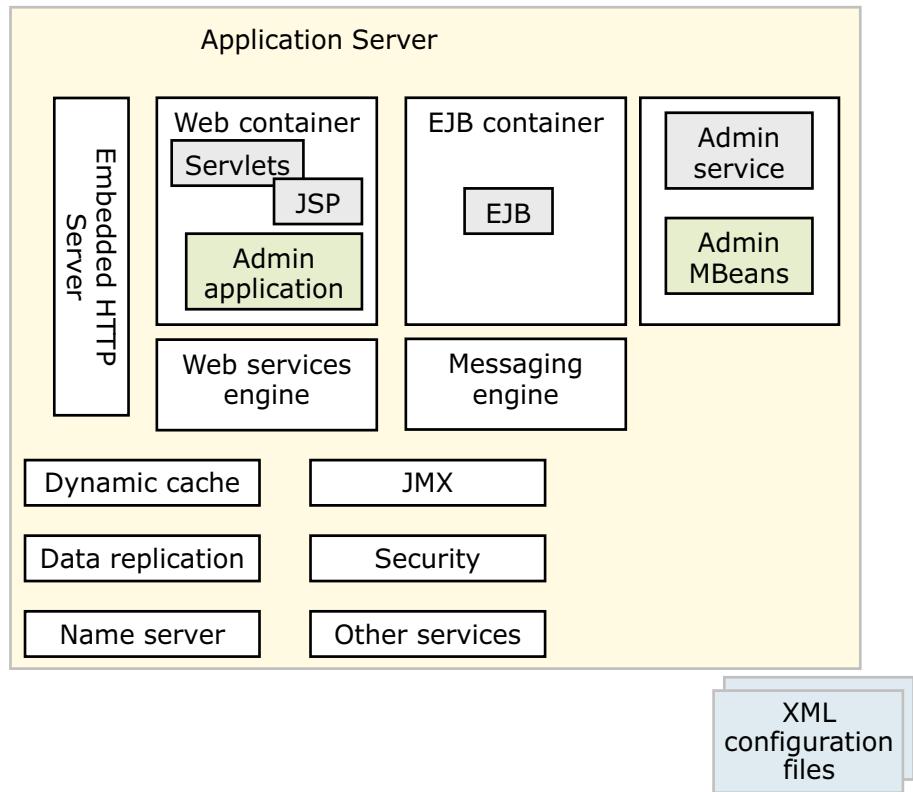
This diagram is the first of a series of four diagrams that build in complexity over the next four pages. Earlier, you saw the runtime depiction of a WebSphere Application Server. This diagram series illustrates the basic architecture of administering WebSphere Application Server.

Two main tools are used to administer WebSphere Application Server: the administrative console and the wsadmin command-line tool.

The configuration of the server is stored in a set of XML files, often referred to as the configuration repository. These files define the server itself, and the resources and services that it provides.

1 of 4: This diagram is a standard application server, as previously mentioned.

WebSphere administration architecture (2 of 4)



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-20. WebSphere administration architecture (2 of 4)

2 of 4: One of the services available within the application server is the administrative service. This service allows for configuration of the application server. The files necessary for configuration are stored outside of the actual application server in a set of XML configuration files. An application that runs within the web container provides users the ability to administer the application server through a web application: the administrative console.

WebSphere administration architecture (3 of 4)

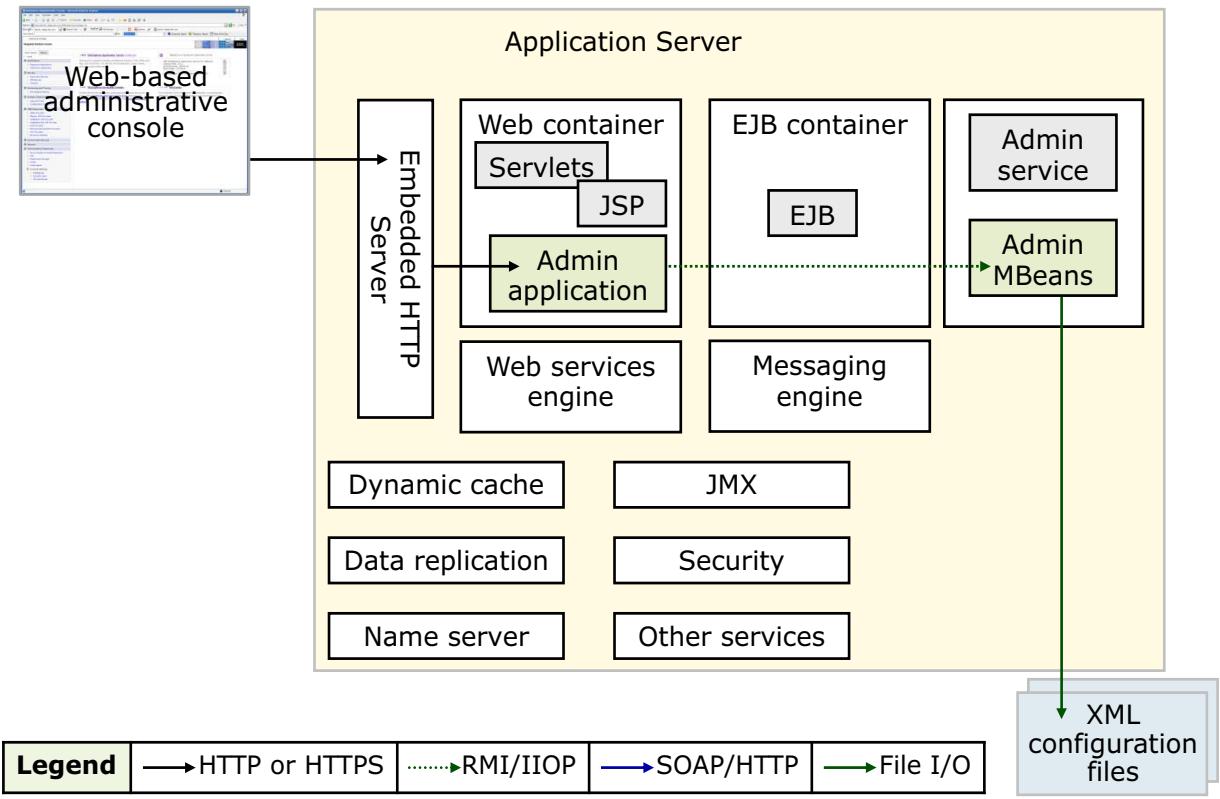


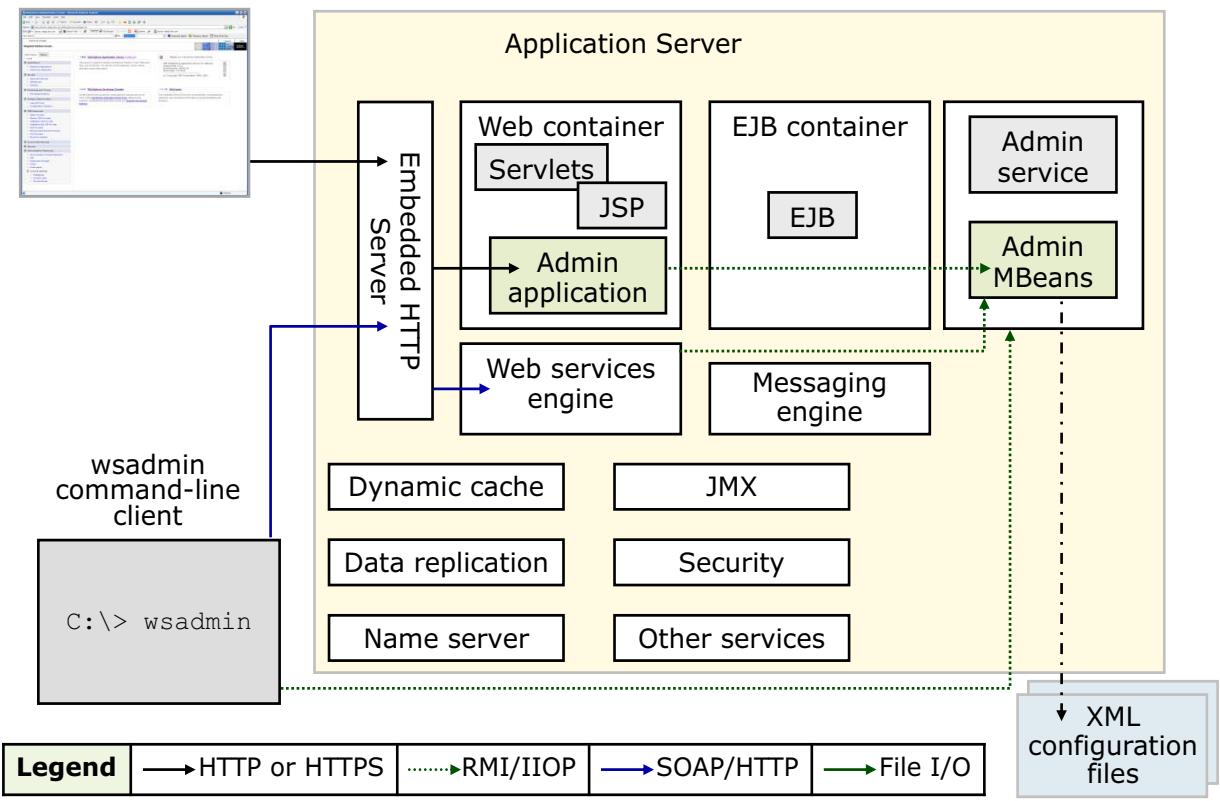
Figure 2-21. WebSphere administration architecture (3 of 4)

© Copyright IBM Corporation 2016

3 of 4: This diagram illustrates communication from the browser to the XML configuration files.



WebSphere administration architecture (4 of 4)



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-22. WebSphere administration architecture (4 of 4)

4 of 4: The `wsadmin` command-line client is used to administer the application server through SOAP, by communicating with the embedded HTTP server, or by using RMI (the default) to communicate directly with the administrative service.

2.3. Profiles

This topic provides information about profiles.

Profiles

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-23. Profiles

WebSphere profile overview

Profiles are sets of files that represent a WebSphere Application Server configuration

WebSphere Application Server files are split into two categories:

- Product files
 - Set of shared read-only static files or product binary files
 - Shared among any instances of the WebSphere Application Server product
- Profiles (configuration files)
 - Set of user-customizable data files
 - Files include WebSphere configuration, installed applications, resource adapters, properties, and log files

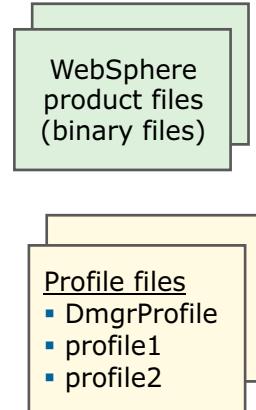
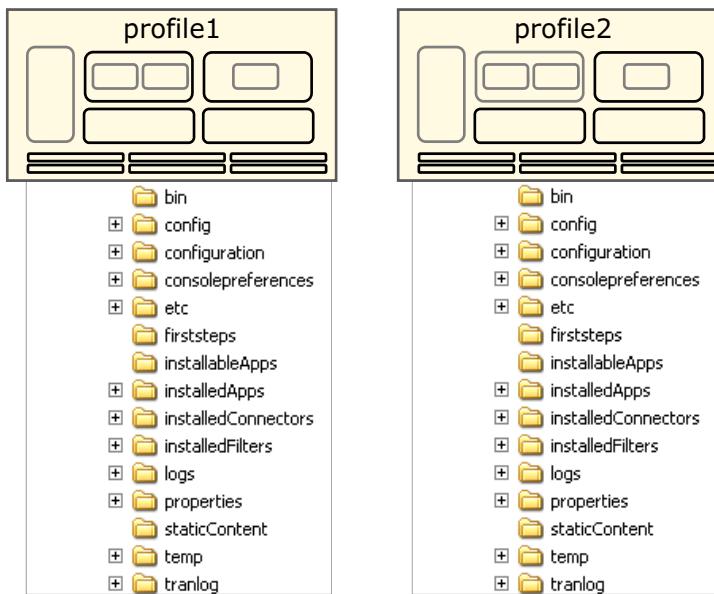


Figure 2-24. WebSphere profile overview

Profiles are the configuration mechanism that you can use to run more than one application server on a single installation of WebSphere product files. For a stand-alone server, the dmgr profile would not exist yet.

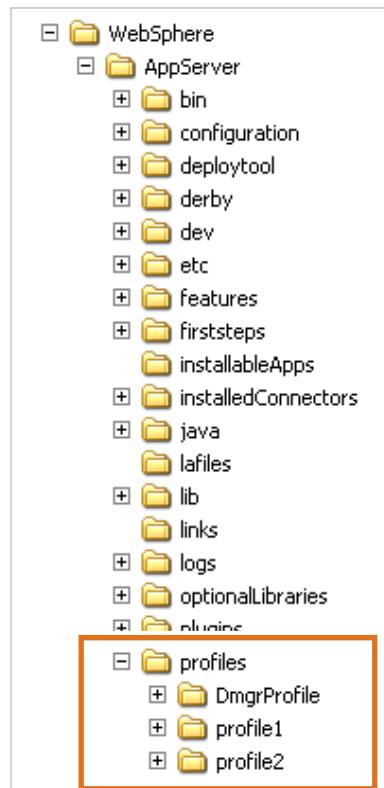
WebSphere profile benefits

- Benefits of profiles:
 - Each profile uses the same product files
 - Simpler than multiple WebSphere installations
 - Less disk space
 - Simplifies application of product updates



WebSphere Application Server architecture: Stand-alone

Figure 2-25. WebSphere profile benefits



© Copyright IBM Corporation 2016

Notice that under the WebSphere installation directory (`<was_root>`) each profile has subdirectories. In the example that is shown, two application servers are each configured according to the files that exist within their own profile directory.



Managing profiles

localuser@washost: /opt/IBM/WebSphere/AppServer/profiles/profile1/bin\$./manageprofiles.sh -help

Function:
Creates, lists, alters or deletes profiles

Syntax:
`manageprofiles -<mode> -<argument> <argument parameter> ...`

The available modes are:

- create
- augment
- delete
- unaugment
- unaugmentAll
- deleteAll
- listProfiles
- listAugments
- backupProfile
- restoreProfile
- getName
- getPath
- validateRegistry
- validateAndUpdateRegistry
- getDefaultName
- setDefaultName
- response
- help

1

2

Profiles are managed through one of the tools provided:

1. **Profile Management Tool**
 - Accessed through the WebSphere Customization Toolbox
 - Gathers user input and starts the `manageprofiles` command-line tool to create the profiles
2. **manageprofiles script**
 - Command-line interface for profile management functions

WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-26. Managing profiles

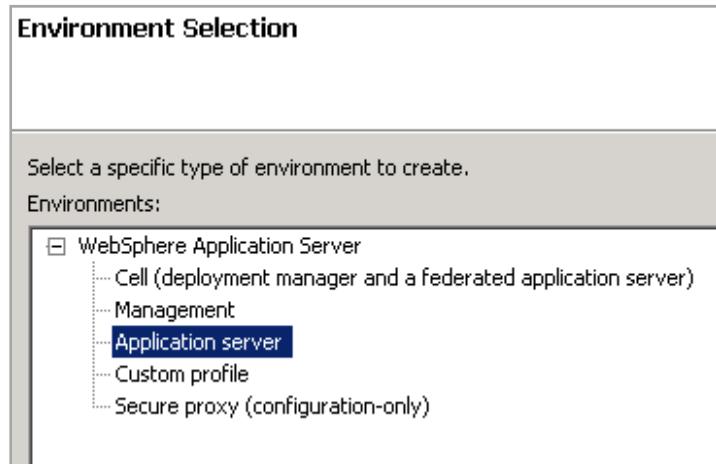
These two tools are available for creating and managing profiles. Profiles are managed through one of the tools provided:

1. The Profile Management Tool (PMT) wizard is an Eclipse-based GUI tool for creating profiles. The wizard gathers user input and starts the `manageprofiles` command-line tool to create the profiles.
2. The `manageprofiles` script is run from a command-line interface for profile management functions.



Profile types

- Cell
 - Deployment manager with a federated application server
- Management
 - Administrative agent
 - Deployment manager
 - Job manager
- Application server
 - Stand-alone
- Custom profile
 - Federated node
(no application server)
- Secure proxy



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-27. Profile types

There are numerous profile types:

- Cell
 - Deployment manager with a federated application server
- Management
 - Administrative agent
 - Deployment manager
 - Job manager
- Application server
 - Stand-alone
- Custom profile
 - Federated node
 - (No application server)
 - Secure proxy

Unit summary

- Describe a typical e-business application architecture
- Describe the architectural differences between WebSphere Application Server packages
- Describe what is running in a WebSphere Application Server node
- Describe the architectural implications of the web server plug-in
- Describe the use of Java Database Connectivity (JDBC) providers and data sources

WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-28. Unit summary

Review questions

1. Which one of the following components provides an environment for running servlets?
 - A. Client module
 - B. Web container
 - C. EJB module

2. Which type of JDBC driver is considered a “thick” driver?
 - A. Type 2
 - B. Type 3
 - C. Type 4

3. Which of the following are components contained within the JVM of the application?
 - A. HTTP Server plug-in
 - B. Embedded HTTP Server
 - C. DB2 database



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-29. Review questions

Write your answers here:

- 1.

- 2.

- 3.

Review answers

1. Which one of the following components provides an environment for running servlets?
 - A. Client module
 - B. Web container
 - C. EJB module

The answer is B.
2. Which type of JDBC driver is considered a “thick” driver?
 - A. Type 2
 - B. Type 3
 - C. Type 4

The answer is A.
3. Which of the following are components contained within the JVM of the application?
 - A. HTTP Server plug-in
 - B. Embedded HTTP Server
 - C. DB2 database

The answer is B.



WebSphere Application Server architecture: Stand-alone

© Copyright IBM Corporation 2016

Figure 2-30. Review answers

Exercise: Profile creation

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-31. Exercise: Profile creation

Exercise objectives

After completing this exercise, you should be able to:

- Use the Profile Management Tool to create a profile
- Verify the profile creation

Unit 3. WebSphere Application Server administrative console

Estimated time

00:30

Overview

This unit describes the features in the administrative console for WebSphere Application Server.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server Network Deployment V9 documentation in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Describe how to access the administrative console
- Describe the administrative console in a cell topology
- Describe the administrative console interface
- Describe the use of the following administrative tools:
 - Help
 - Preferences
 - Filters
 - Guided Activities
 - Troubleshooting
- Describe user and group administrative roles
- Describe the Tivoli Performance Viewer

IBM Training IBM

Administrative console

View: All tasks

- [Welcome](#)
- [Guided Activities](#)
- [Servers](#)
- [Applications](#)
- [Services](#)
- [Resources](#)
- [Security](#)
- [Environment](#)
- [System administration](#)
- [Users and Groups](#)
- [Monitoring and Tuning](#)
- [Troubleshooting](#)

Suite Name	Version
WebSphere Application Server	9.0.0.0

- Web browser-based tool that manages WebSphere Application Server
- Supports a full range of product administrative activities

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

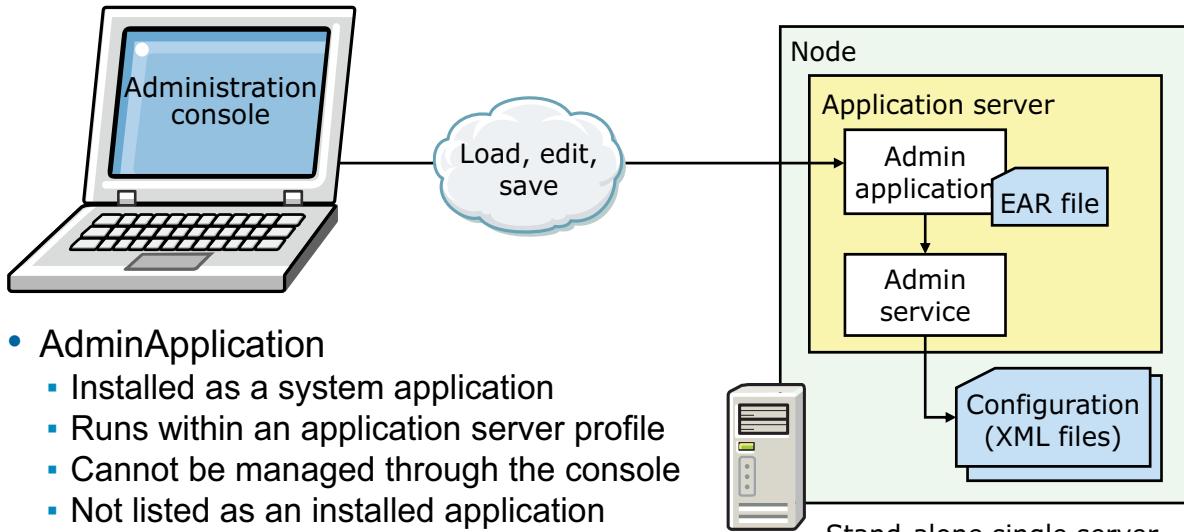
Figure 3-2. Administrative console

The administrative console is also known as the Integrated Solutions Console.

The console is accessed by using any supported web browser and entering the web address:
`http://<host_name>:9060/ibm/console`

Using the browser back button with the console can produce unexpected results and is not supported. Use the controls and links that are provided in the console to navigate between pages and applications.

Starting the administrative console



- AdminApplication
 - Installed as a system application
 - Runs within an application server profile
 - Cannot be managed through the console
 - Not listed as an installed application
 - WebSphere Security protects it
- Accessed through
<http://localhost:9060/ibm/console>
 - 9060 is the default port

[WebSphere Application Server administrative console](#)

© Copyright IBM Corporation 2016

Figure 3-3. Starting the administrative console

The administrative console is implemented as an application (AdminApplication). It is deployed as a system application during product installation and within an application server. The AdminApplication cannot be managed through the console, and is not listed as an installed application. WebSphere administrative security is used to control which users can log in to the console.



Administrative console in a cell topology

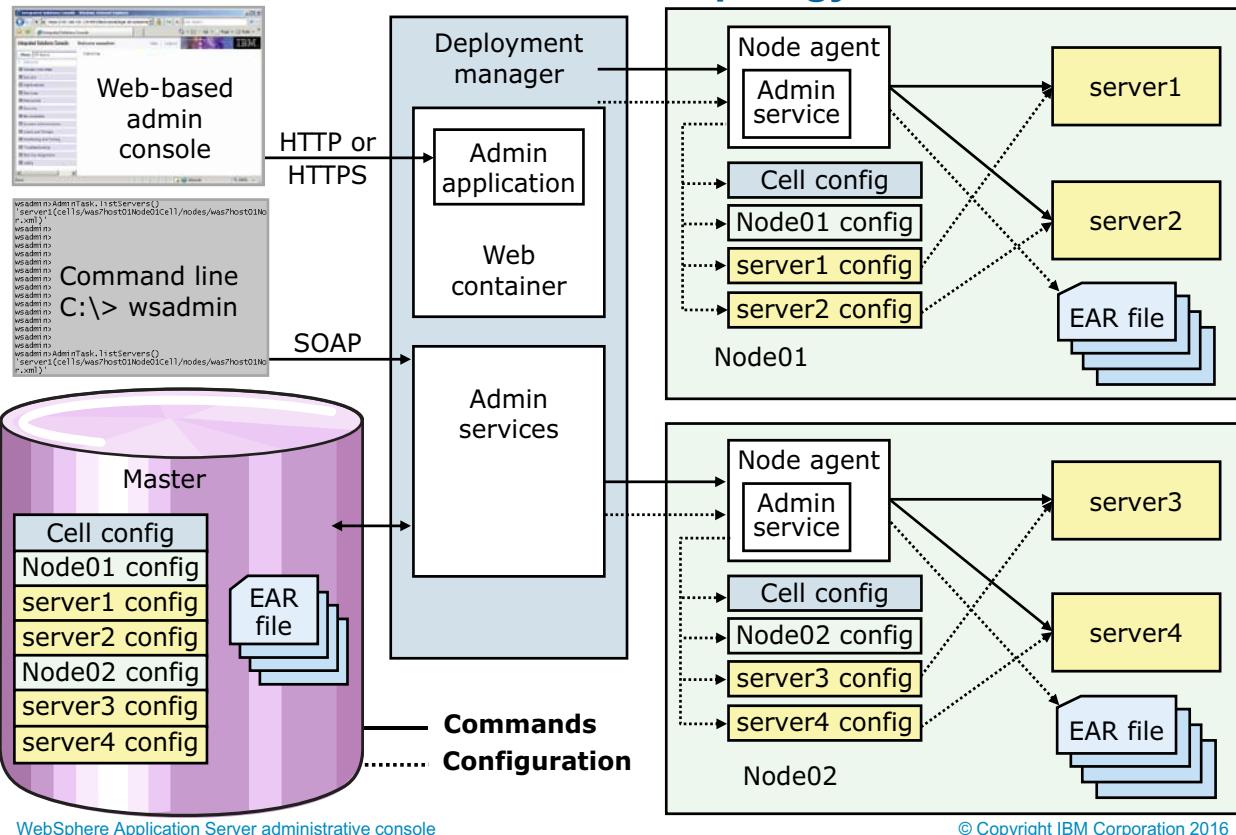


Figure 3-4. Administrative console in a cell topology

This diagram depicts two separate views of WebSphere administration.

The flow of administration commands is shown as solid lines, and the flow of administration configuration files is shown as dotted lines.

For both of these flows, the diagram proceeds from left to right.



Console login

1 Log on to the console

- **User ID**
 - A string that identifies the user
 - Is used to track changes that the user makes
 - User ID must be **unique**
- **Password**
 - If security is set, specify a password



2 User ID conflict

- Shown when another user is logged in with the same user ID



WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-5. Console login

To access the console, enter your user ID and password and then click **Log in**. The password is required only if security is enabled.

If the user ID that you provide is already logged in at a different location, you are prompted to choose between logging out from the other location or returning to the login page. If you log out the user from the other location, you might be prompted to recover unsaved changes that the user made.

Recovering prior changes



- If your prior session times out, you can recover prior changes
- Two options are available:
 - Work with the default administrative configuration
 - Work with the administrative configuration from the prior session

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-6. Recovering prior changes

After you are logged in, be sure to use the **Logout** link in the console toolbar when you are finished working with the console to prevent unauthorized access. If no activity takes place during this login session for an extended period, the session expires, and you must log in again to access the console. The administrator can change the session timeout. The default is set to 30 minutes.

Administrative console session timeout

```

timeout.jacl - Notepad
File Edit Format View Help
set dep [$AdminConfig getid /Deployment:isclite/]
set appDep [$AdminConfig list ApplicationDeployment $dep]
set sesMgmt [$AdminConfig list SessionManager $appDep]

# check if existing sesMgmt there or not, if not then create a new one, if exist then modify it
if {$sesMgmt == ""} {
    # get applicationConfig to create new SessionManager
    set appConfig [$AdminConfig list ApplicationConfig $appDep]
    if {$appConfig == ""} {
        # create a new one
        set appConfig [$AdminConfig create ApplicationConfig $appDep {}]
        # then create a new SessionManager using new Application Config just created
        set sesMgmt [$AdminConfig create SessionManager $appConfig {}]
    } else {
        # create new SessionManager using the existing ApplicationConfig
        set sesMgmt [$AdminConfig create SessionManager $appConfig {}]
    }
}

# get tuningParams config id
set tuningParams [$AdminConfig showAttribute $sesMgmt tuningParams]
if {$tuningParams == ""} {
    # create a new tuningParams
    $AdminConfig create TuningParams $sesMgmt {invalidationTimeout 60}
} else {
    #modify the existing one
    $AdminConfig modify $tuningParams {{invalidationTimeout 60}}
}

# saving the configuration changes
$AdminConfig save

```

- Issue **wsadmin -f <path to timeout script>/timeout.jacl**

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-7. Administrative console session timeout

This slide shows a Jacl script that modifies the administrative console timeout duration. You are going to work with a similar script in one of the upcoming lab exercises.



Administrative console panels

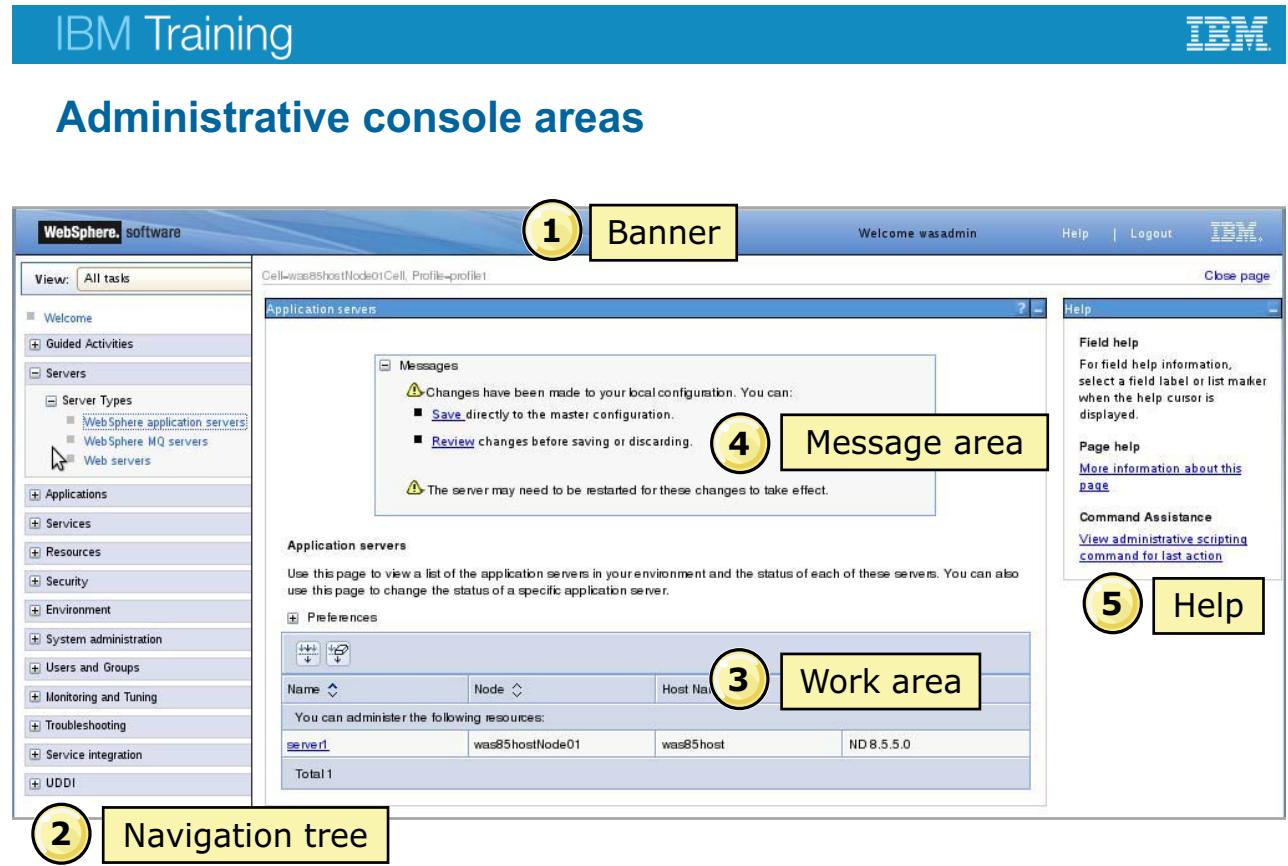
The screenshot displays three distinct types of administrative console panels:

- Collection pages:** The top panel, titled "Enterprise Applications", shows a list of installed applications. It includes a toolbar with buttons for Start, Stop, Install, Uninstall, Update, Rollout Update, Remove File, Export, Export DDL, and Export File. A yellow callout labeled "1 Collection pages" points to the title bar.
- Detail pages:** The middle panel, titled "Enterprise Applications > DefaultApplication", provides configuration options for a specific application. It features sections for "General Properties" (Name set to "DefaultApplication", Application reference validation set to "Issue warnings"), "Modules" (links to Manage Modules and Display module build Ids), and "Web Module Properties" (links to Session management, Context Root For Web Modules, and JSP and JSF options). A yellow callout labeled "2 Detail pages" points to the title bar.
- Wizard pages:** The bottom panel, titled "DefaultApplication", is a wizard page for managing JDBC connections. It lists three resources: DefaultApplication, ivtApp, and query. A yellow callout labeled "3 Wizard pages" points to the title bar. A bulleted list within this callout states: "JDBC wizard is an example".

WebSphere Application Server administrative console © Copyright IBM Corporation 2016

Figure 3-8. Administrative console panels

This slide shows the three basic types of pages in the administrative console. These common patterns provide a consistency of format within the administrative console.



WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-9. Administrative console areas

The purpose of this slide is to show the layout of the administrative console. The details are not the important message here. It is not critical that you can read the text on this slide.

The next several slides show each of the sections in detail.



Administrative console banner

Item	Description
Welcome	<ul style="list-style-type: none"> The administrative console home page Contains links to information sources
Logout	<ul style="list-style-type: none"> Logs you out of the administrative console session Shows the Login page after successfully logging out If changes were made and not saved, the Save page is shown
Help	<ul style="list-style-type: none"> Opens a new web browser with detailed online help for the administrative console Note: This Help is not the IBM Knowledge Center

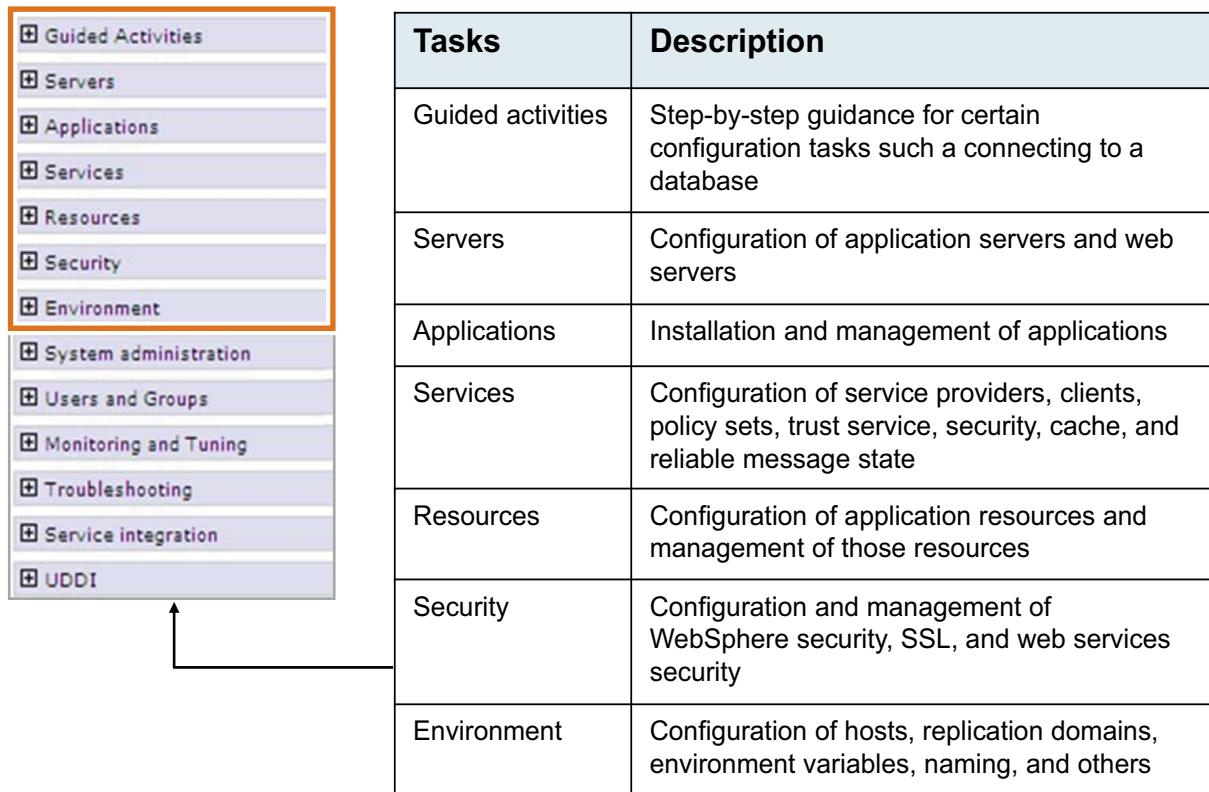
WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-10. Administrative console banner

After you are logged in, be sure to use the **Logout** link in the console toolbar when you are finished working with the console to prevent unauthorized access.

Administrative console navigation tree (1 of 2)



WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-11. Administrative console navigation tree (1 of 2)

This slide briefly describes some of the tasks that are listed in the navigation tree. Expanding any of these tasks reveals more subtasks.

Administrative console navigation tree (2 of 2)

Tasks	Description
System administration	Configuration and management of components, users, and preferences
Users and Groups	Configuration of users and groups
Monitoring and Tuning	Configuration of the Performance Monitoring Infrastructure and Tivoli Performance Viewer
Troubleshooting	Tracking and verification of configuration errors and problems
Service integration	Configuration for service integration buses, messaging engines, and messages destinations
UDDI	Configuration of UDDI nodes

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-12. Administrative console navigation tree (2 of 2)

This slide briefly describes the remaining tasks that are listed in the navigation tree. Some of these tasks are explored in more detail later in this unit.



Administrative console help (1 of 2)

1 **Console help**

- Click **Help** from console banner
- Select from list of references

About the help system

With the help system, users can view, browse, and search online information. The help system is built upon open source software developed by the Eclipse Project (www.eclipse.org). It can display content that has been packaged as an Eclipse documentation plug-in.

The help system uses an embedded web application server to handle content requests in the system. This embedded web server uses a random port to avoid port conflicts between applications.

Navigating by keyboard

Use the following key combinations to navigate the help system by keyboard:

- In the Topic pane, to go to the next link, press Tab.
- To expand and collapse a node in the navigation tree, press the Right and Left arrows. In JAWS 6.0, you also need to use the Alt key (for example, Alt+Left arrow).
- To move to the next topic node, press the Down arrow or Tab. In JAWS 6.0, use Ctrl+Shift+Down arrow.

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-13. Administrative console help (1 of 2)

All of the help panels that you can access from the administrative console are also accessible from the WebSphere Application Server documentation in IBM Knowledge Center.

IBM Training IBM

Administrative console help (2 of 2)

2

Page help

- Click **More information about this page** from help workspace

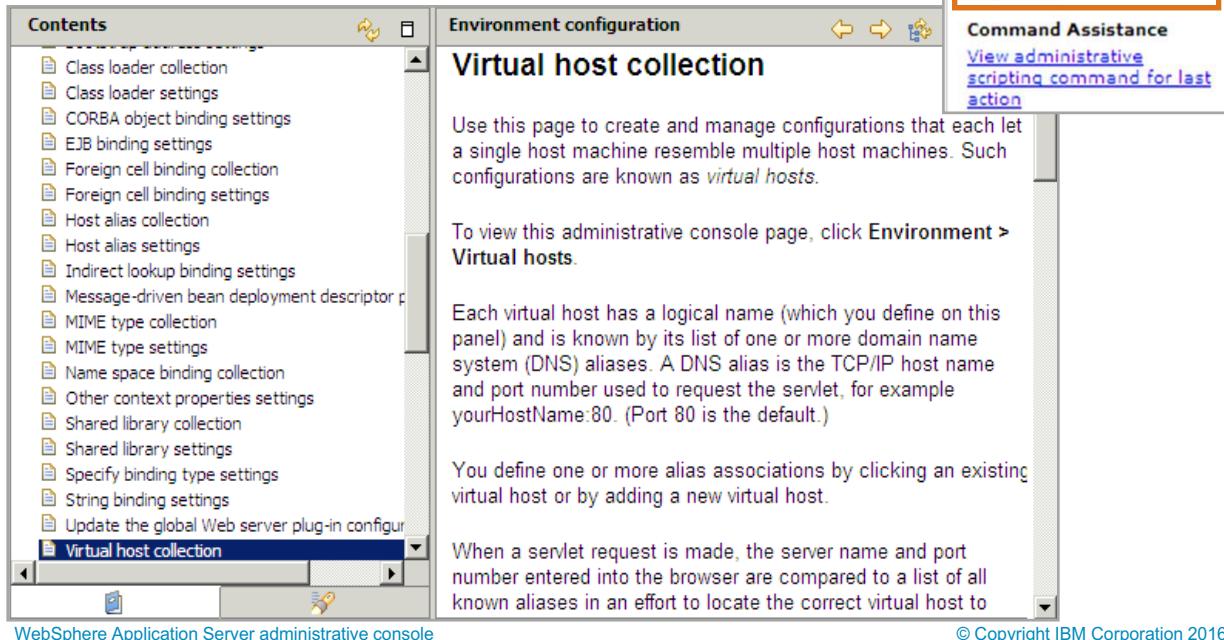


Figure 3-14. Administrative console help (2 of 2)

You can access help in the following ways:

- Click either of the following tabs of an online help page:
 - Click the **Help index** tab and select from the list of help panels to view administrative console help information.
 - Click the **Search** tab, provide search terms, and then click **Search**. Under Results, select a help panel that contains the search information.
- In the help portal that is on the right side of the administrative console panel, do one or all of the following tasks:
 - Click a field label or a list marker in the administrative console panel for the help to display under field help. Alternatively, place the cursor over the field label or the list marker for the corresponding help to display at the cursor. Attention: When you place the cursor over the field label or list marker, the help might be truncated in a Firefox browser. Click the field label or list marker so that the full help is displayed under field help.
 - Click the link under **Page help** to access the help panel for the administrative console panel. The help panel is the same help panel that displays when you click the "?" icon.

- If Command assistance is listed, click the link under **Command assistance** to view wsadmin scripting commands for the last action that completed within the console panel.



Administrative console preferences

The screenshot shows the 'Console preferences' page in the WebSphere Application Server administrative console. The left sidebar lists various administrative categories. Under 'System administration', the 'Console Preferences' option is selected and highlighted with a red box. The main panel displays several preference settings with checkboxes:

- Turn on workspace automatic refresh
- No confirmation on workspace discard
- Use default scope
- Show the help portlet
- Enable command assistance notifications
- Log command assistance commands

Below these settings is a link labeled 'Bidirectional support options'. At the bottom of the panel are 'Apply' and 'Reset' buttons. A yellow callout box on the right side of the panel contains the text: 'Specify how features of the administrative console workspace behave'.

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-15. Administrative console preferences

Use the preference settings to specify how you want information to be displayed on an administrative console panel. The preference settings vary from one administrative console panel to another.

Turn on workspace automatic refresh: Specifies whether you want the administrative console workspace to refresh automatically after the administrative configuration changes. The default is for the workspace to refresh automatically. If you delete a WebSphere variable, for example, the WebSphere variables page refreshes automatically and shows the updated list of WebSphere variables in the WebSphere variables collection.

Enable command assistance notifications: Specifies whether to send Java Management Extensions (JMX) notifications that contain command assistance data from the administrative console. Enablement of the notifications allows integration with product tools such as the Toolkit Jython editor for WebSphere Application Server. Enablement of this option is suggested for non-production environments only.



Administrative console preferences, filters, and scope

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

1 Preferences

Maximum rows: 20
 Retain filter criteria
 Show items at the following authorization group level: All Roles

2 Select resources

Start Stop Install Uninstall Update Rollout Update Remove File Export Export DDL
 Select Name Application Status
 To filter the following table, select the column by which to filter, then enter filter criteria (wildcards: *, ?, %).
 Filter: Name Search terms: * **3 Set filters**

You can administer the following resources:

	DefaultApplication	DefaultApp
<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	DefaultApp	
<input type="checkbox"/>	query	

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-16. Administrative console preferences, filters, and scope

Maximum rows: Indicates the maximum number of rows to display per page when the collection is large.

Show resources at one authorizing group level: Specifies the authorization group level that is used to filter the resources in the table. Only those roles that apply to your ID can display in the table. Valid values are All Roles, Administrator, Deployer (for application collection panels only), Operator, Configurator, and Monitor. If All Roles is selected, then all the resources that you are authorized to view are displayed in the table and grouped by role. Otherwise, the resources for the role that is selected are displayed in the table.

IBM Training

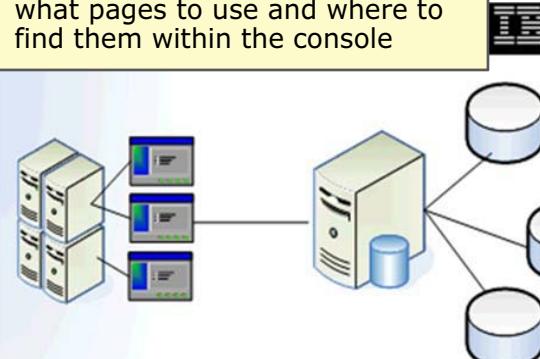
Guided Activities



The screenshot shows the 'Integrated Solutions Console' interface with a sidebar containing various administrative categories like Servers, Applications, and Security. The 'Guided Activities' section is highlighted with an orange box. It lists two tasks: 'Connecting to a database' and 'Routing requests through a Web server to an application server'. The main content area is titled 'Connecting to a database' and includes an 'Introduction' section with text about configuring database access for an application, followed by a list of steps: Start, Configure credentials for secure database access, Configure a JDBC provider, Configure WebSphere variables, Configure a data source, Save and synchronize configuration, and Test database connection.

• Structured steps through common administrative tasks

• Otherwise, must know exactly what pages to use and where to find them within the console



The diagram illustrates a network architecture. On the left, there is a cluster of three servers. One server in the cluster is connected to a central server, which in turn is connected to a network of three databases.

Assumptions

It is assumed that you are installing an application that need securely access data from a relational database. For more information on this task, see the following sources in the information center:

- Configuring a JDBC provider and data source
- Deploying data access applications
- Learn about data access resources

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-17. Guided Activities

The Guided Activities feature displays each administrative console page on which you must do a task. The following information surrounds the task to help you do the task successfully.

The Guided Activities feature displays an introduction to the task, introducing essential concepts and describing when and why to perform the task:

- Other tasks to perform before and after performing the task
- The main steps to complete during this task
- Hints and tips to help you avoid and recover from problems
- Links to field descriptions and extended task information, which can be found in the online documentation



My tasks

1 Select **My tasks** from the navigation view selection list

2 Select tasks to add to the My tasks list

3 The tasks are shown in the My tasks list

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-18. My tasks

Use “My tasks” to create and edit a list of tasks to view in the console navigation. A task includes a page that contains one or more web applications, or console modules, that are used to complete that task. When you first access the console, all tasks to which you have access are displayed in the navigation. “My tasks” is especially useful to customize the navigation to show only the tasks that you use most often. After you customize your tasks, “My tasks” is initially displayed each time you log in to the console.

IBM Training

IBM

Users and groups

The screenshot shows the 'Administrative user roles' page. On the left, a sidebar lists various administrative categories. The 'Users and Groups' section is highlighted with an orange box and contains four links: 'Administrative user roles', 'Administrative group roles', 'Manage Users', and 'Manage Groups'. A yellow circle labeled '1' points to this section. On the right, the main content area has a header 'Administrative user roles' and a sub-header 'Administrative user roles > User'. It includes a note about adding or removing administrative roles to users. Below this is a section titled 'Search and Select Users' with a search string input field containing an asterisk (*) and a 'Search' button. To the right is a list of 'Available' users ('wasadmin3') and a list of users 'Mapped to role' ('wasadmin2'). Both lists have 'Select All' and 'Deselect All' buttons. A yellow circle labeled '2' points to the 'Administrative roles' list on the right, which contains a bulleted list of roles: Monitor, Configurator, Operator, Administrator, ISC Admins, Deployer, Admin Security Manager, and Auditor.

Administrative user roles

Administrative user roles > User

Use this page to add, update or to remove administrative roles to users. Assigning administrative roles to users enables them to administer application servers through the administrative console or through wsadmin scripting.

* Role(s)

Admin Security Manager
Administrator
Auditor
Configurator

Search and Select Users

Decide how many results to display, enter a search string (use * for wildcard), and click Search. Select the Available list and add them to the Mapped to role list. Users which have already been mapped will be returned in the search results.

Search string: *

Maximum results to display: 20

Available: wasadmin3

Mapped to role: wasadmin2

2

Administrative roles

- Monitor
- Configurator
- Operator
- Administrator
- ISC Admins
- Deployer
- Admin Security Manager
- Auditor

1

User and group management

- Administer user and group roles
- Manage users and groups

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-19. Users and groups

The following pages are used as described:

- **Administrative user roles:** Use this page to add, update, or remove administrative roles for users. Assigning administrative roles to a user enables the user to administer application servers through the administrative console or through wsadmin scripting.
- **Administrative group roles:** Use this page to add, update, or remove administrative roles for groups. Assigning administrative roles to a group enables the group to administer application servers through the administrative console or through wsadmin scripting.
- **Manage Users and Manage Groups:** Enables you to create or delete users and groups within your user registry.



Troubleshooting information

The screenshot shows the 'Troubleshooting' section of the WebSphere Application Server administrative console. The left sidebar contains the following navigation items:

- Troubleshooting**
 - Logs and trace
 - Configuration problems
 - Class loader viewer
 - Java dumps and cores
- Configuration Validation**
 - Configuration error
 - Configuration warning
 - Configuration information
- Diagnostic Provider**
 - Tests
 - State Data
 - Configuration Data
- Runtime Messages**
 - Runtime error
 - Runtime warning
 - Runtime information

- **Troubleshooting**
 - Configure log and trace settings
 - Identify and view configuration problems
 - View class loaders for modules within the topology of enterprise applications
 - Generate thread, heap, and system memory dumps
- **Configuration Validation**
 - View problems that exist in the present configuration
- **Diagnostic Provider**
 - Review the startup configuration, current configuration, and current state of a diagnostic domain
- **Runtime Messages**
 - Review runtime error, warnings, and information messages

WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-20. Troubleshooting information

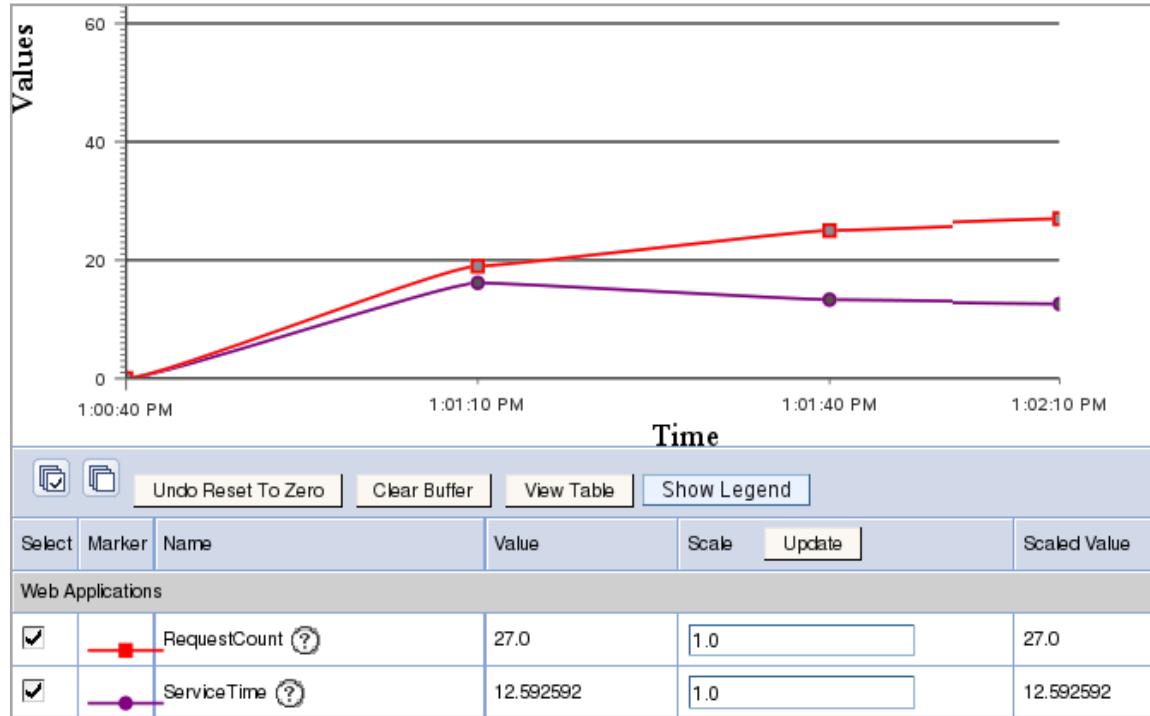
The Class Loader Viewer helps you diagnose problems with class loaders.

Diagnostic Providers are a quick method for viewing configuration and the current state of individual components within an application server environment.



Tivoli Performance Viewer

- Integrated browser-based performance viewer and advisor



WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-21. Tivoli Performance Viewer

Administrators and programmers can use Tivoli Performance Viewer to monitor the overall health of WebSphere Application Server from within the administrative console.

From Tivoli Performance Viewer, you can view current activity or log Performance Monitoring Infrastructure (PMI) performance data for the following resources:

- System resources such as processor utilization
- WebSphere pools and queues such as a database connection pool
- Customer application data such as average servlet response time

Unit summary

- Describe how to access the administrative console
- Describe the administrative console in a cell topology
- Describe the administrative console interface
- Describe the use of the following administrative tools:
 - Help
 - Preferences
 - Filters
 - Guided Activities
 - Troubleshooting
- Describe user and group administrative roles
- Describe the Tivoli Performance Viewer

Review questions

1. True or False: The AdminApp application can be configured with the administrative console.
2. True or False: Configurable console settings, including session timeout, can be set through the Console Preferences page of the system administration task.



WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-23. Review questions

Write your answers here:

- 1.
- 2.

Review answers

1. True or False: The AdminApp application can be configured with the administrative console.
The answer is False. You cannot configure the AdminApp with the administrative console.
2. True or False: Configurable console settings, including session timeout, can be set through the Console Preferences page of the system administration task.
The answer is False. The session timeout is set from a script.



WebSphere Application Server administrative console

© Copyright IBM Corporation 2016

Figure 3-24. Review answers

Exercise: Exploring the administrative console

© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 3-25. Exercise: Exploring the administrative console

Exercise objectives

After completing this exercise, you should be able to:

- Verify that WebSphere Application Server is started
- Start the administrative console
- Explore the navigation and functions of the administrative console
- Use the administrative console to examine configuration information, resources, and properties

Unit 4. Introduction to the PlantsByWebSphere application

Estimated time

00:15

Overview

This unit introduces the PlantsByWebSphere application architecture, and explains how it is used to demonstrate WebSphere Application Server concepts and functions.

How you will check your progress

- Lab exercises

References

WebSphere Application Server Network Deployment V9 documentation in IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome](http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html)

[base.html](http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html)

Unit objectives

- Describe the architecture and components of the PlantsByWebSphere application
- Explain how the application is used as a case study for WebSphere Application Server

[Introduction to the PlantsByWebSphere application](#)

© Copyright IBM Corporation 2016

Figure 4-1. Unit objectives

IBM Training

PlantsByWebSphere application



- Simple shopping cart application
 - Added server information
 - Uses Derby, but can use DB2
 - Built as an enhanced EAR

Introduction to the PlantsByWebSphere application

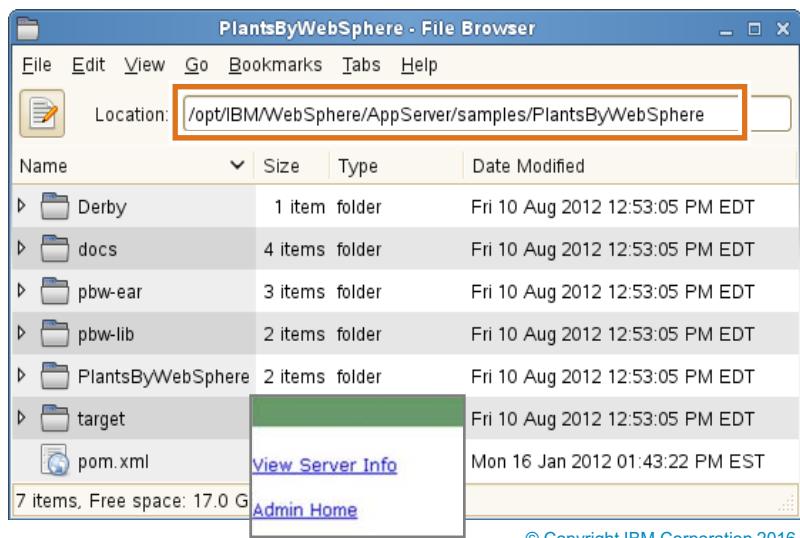
© Copyright IBM Corporation 2016

Figure 4-2. PlantsByWebSphere application

PlantsByWebSphere is a simple shopping cart application that is available with the WebSphere Application Server distribution. It uses Derby as its back-end database, but can also be configured to work with other databases such as DB2.

PlantsByWebSphere sample

- PlantsByWebSphere is available through the **Features > Samples Applications** that come with the WebSphere Application Server
 - Can be found in `<was_root>\samples\PlantsByWebSphere`
 - More samples are available through the IBM Knowledge Center
- The version of PlantsByWebSphere used in this course is altered slightly
 - To make the PlantsByWebSphere application more useful for educational purposes, two links are added to the bottom of the Help page
(View Server Info and Admin Home)



Introduction to the PlantsByWebSphere application

Figure 4-3. PlantsByWebSphere sample

PlantsByWebSphere is a sample application. Numerous other sample applications are now available through the IBM Knowledge Center. In previous versions of WebSphere, these other samples were available directly through the WebSphere Application Server distribution.

For educational purposes, the PlantsByWebSphere application is modified slightly. This modification is made to more easily demonstrate several functional issues that are useful to point out in a class such as this.

- To demonstrate session failover in a clustered environment, another link that is called View Server Info is added to the Help page. This link shows server information so that it is easy to understand where the server affinity is mapped.
- To more easily demonstrate application security, another link that is called Admin Home is also added to the Help page. This link makes it easier to access the Admin servlet, which is protected through Java EE security. Otherwise, the user must type in the URL (or use a bookmark).

This course also uses DB2 instead of Derby, giving students a chance to configure data sources and JDBC drivers (instead of relying on the embedded definitions as part of the enhanced EAR). To rebuild the DB2 database, run the CreateDB script in the software directory.

IBM Training IBM

Home page: <http://<hostname>/PlantsByWebSphere>

PLANTS BY WEBSHERE

Your shopping cart is currently empty

Flowers Fruits & Vegetables Trees Accessories

HOME : SHOPPING CART : LOGIN : HELP :

Gardens of Summer

They all start with the right flowers... and we've got them all

Tips

Preserve extra grass seed by keeping it dry. Tape boxes and bags closed, or seal them into plastic bags. Be sure to remove extra air from the bags. Store all seed in a cool, dry area such as a garage or basement.

Specials

	Bonsai Tree \$30.00 each		Red Delicious Strawberries \$3.50 (50 seeds)		Tulips \$17.00 (10 bulbs)
--	-----------------------------	--	---	--	------------------------------

Powered by **IBM WebSphere** e-business software

Flowers : Fruits & Vegetables : Trees : Accessories : Home : Shopping Cart : My Account : Login : Help

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-4. Home page: <http://<hostname>/PlantsByWebSphere>

This screen shows the home screen for the PlantsByWebSphere application.

The screenshot displays two side-by-side web pages for the "PLANTS BY WEBSPHERE" application.

Left Page (Registration Screen):

- Header:** PLANTS BY WEBSPHERE
- Navigation Bar:** Flowers, Fruits & Vegetables, Trees, Accessories
- Page Content:**
 - Home > Sign in
 - Registration**: Enter the information below to set up your account without your permission. With your permission, we can share your address with our trusted business partners.
 - Required fields are denoted with a red asterisk (*).
 - Login Information**:
 - E-mail address: *user@plants.com
 - Password: ******
 - Verify Password: ******
 - Contact Information**:
 - First Name: *Ima
 - Last Name: *Gardner
 - Address Line 1: *123 Main St
 - Address Line 2:
 - City: *Gotham
 - State: *PA
 - ZIP Code: *15222
 - Phone (daytime): *412-555-1234
 - Buttons:** Register (disabled), Sign in (disabled)

Right Page (Login Screen):

- Header:** HOME : SHOPPING CART **LOGIN** HELP :
- Navigation Bar:** Flowers, Fruits & Vegetables, Trees, Accessories
- Page Content:**
 - Home
 - Login or Register**: If you are a returning customer and previously set up an account, please enter your e-mail address and password below.
 - E-mail address:
 - Password:
 - Sign in** button (disabled)

Callout Box (Bottom Right):

- Log in with
 - User: plants@plantsbywebsphere.ibm.com
 - Password: plants
- Or, register as new user
 - Enter your own data

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-5. Login and Registration

This screen shows the Login and Registration screens for the PlantsByWebSphere application.

It is not necessary to log in to the application unless the user wants to go through the purchasing screens.

PLANTS BY WEBSHERE

Home > Sign in

Account Update

Enter the information below to update your account. This information will not be shared without your permission. With your permission we will only share your name and email address with our trusted business partners.

Required fields are denoted with a red asterisk (*).

Contact Information

First Name	*David
Last Name	*Grover
Address Line 1	*123 Main Street
Address Line 2	Apt C
City	*Raleigh
State	*NC
ZIP Code	*27604
Phone (daytime)	*919-555-1234

Update

- Click **My Account** on the bottom to see your account information
- If you are not logged in, you are prompted to log in or register

Powered by
IBM WebSphere

Flowers : Fruits & Vegetables : Trees : Accessories : Home : Shopping Cart **My Account** Login : Help

Introduction to the PlantsByWebSphere application © Copyright IBM Corporation 2016

Figure 4-6. My Account

This screen shows the My Account page for the PlantsByWebSphere application. If this screen shows user information, it means that the user is already logged in. If not, a login screen is displayed.



Shopping

Select a tab, and then click items to view or add to cart

The figure consists of three vertically stacked screenshots of the PlantsByWebSphere application's shopping interface.

- Screenshot 1: Trees Tab**
The title bar says "PLANTS BY WEBSPHERE". The navigation bar includes tabs for "Flowers", "Fruits & Vegetables", "Trees" (which is highlighted with an orange border), and "Accessories". Below the navigation bar, it says "Home" and "Trees". On the right, it says "Page 1 of 1" and shows a small image of a tree labeled "Maple".
- Screenshot 2: Flowers Tab**
The title bar says "PLANTS BY WEBSPHERE". The navigation bar includes tabs for "Flowers" (highlighted with an orange border), "Fruits & Vegetables", "Trees", and "Accessories". Below the navigation bar, it says "Home" and "Flowers". On the right, it says "Page 1 of 1" and shows a small image of flowers labeled "Yellow Shasta Daisy".
- Screenshot 3: Fruits & Vegetables Tab**
The title bar says "PLANTS BY WEBSPHERE". The navigation bar includes tabs for "Flowers", "Fruits & Vegetables" (highlighted with an orange border), "Trees", and "Accessories". Below the navigation bar, it says "Home" and "Fruits & Vegetables". On the right, it says "Page 1 of 1" and shows a small image of various fruits and vegetables labeled "Lily".

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-7. Shopping

This screen shows three of the shopping tabs for the PlantsByWebSphere application. These include Trees, Flowers, and Fruits & Vegetables. From these screens, users can click the individual items and add them to their cart.



IBM

Select an item

When you have an item page open, you see the details and are able to **change the quantity** and **Add to cart**

PLANTS BY WEBSPHERE

Flowers Fruits & Vegetables Trees Accessories

[Home](#) > Trees

Bonsai



Tabletop Fun

Bonsais are great miniature replicas of your favorite yard tree. They can be indoors or out -- and their size makes them perfect for tabletop decoration.

Item Selection:

ITEM#	DESCRIPTION	PRICE	QUANTITY
T0003	0.5 gallon mature tree	\$30.00	<input type="text" value="1"/>

Add to cart

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-8. Select an item

This screen shows the details for an individual item within the PlantsByWebSphere application. Users can click **Add to cart** and continue to shop. Or, if they want, they can choose to check out.

IBM Training 

Shopping cart

PLANTS BY WEBSPHERE

Flowers Fruits & Vegetables Trees Accessories

Home

Shopping Cart

Here are the items you have selected. To recalculate your total after changing the quantity of an item, select the 'Recalculate' button. To remove an item from your cart, enter "0" as the quantity. Select 'Checkout Now' to begin the checkout process.

ITEM #	ITEM DESCRIPTION	PACKAGING	QUANTITY	PRICE	SUBTOTAL
F0002	Baby Breath	2 plants	<input type="text" value="2"/>	\$6.00	\$12.00
V0007	Watermelon	1 pkt. (100 seeds)	<input type="text" value="1"/>	\$2.00	\$2.00

Order Subtotal:\$14.00

[Continue Shopping](#)
 [Recalculate](#)
 [Checkout Now](#)

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-9. Shopping cart

This screen shows the shopping cart screen for the PlantsByWebSphere application. From this screen, users can modify the quantities of their items, click **Checkout Now**, or continue to shop.



Checking out: Billing information

Shipping Information

Check here if the shipping address is the same as the billing address.

Full Name *	David Grover
Address Line 1 *	123 Main Street
Address Line 2	Apt.C
City *	Raleigh
State *	NC
Zip Code *	27604
Phone (daytime) *	919-555-1234

Shipping Method

Select a shipping method below. Your order total will be updated on the next page.

Shipping Method * Standard Ground (3 to 6 business days) \$4.99 ▾

Credit Card *	American Express ▾
Card Number	5432 5432 5432 5323 orderinfo:ccardnum: Credit card numbers must be entered as XXXX XXXX XXXX XXXX.
Expiration Month *	01 ▾
Expiration Year *	2015 ▾
Cardholder Name *	David Grover

Continue

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-10. Checking out: Billing information

This screen shows the first of the checkout screens for the PlantsByWebSphere application. This screen is where users confirm the billing and shipping information. They also must enter credit card data. Be careful here, as the format of the credit card information is required to be **XXXX XXXX XXXX XXXX** (with spaces).



Checking out: Submit

PLANTS BY WEBSPHERE

Flowers Fruits & Vegetables Trees Accessories

Home > Shopping Cart >

Review Your Order

Review your order below and select 'Submit Order' at the bottom to place your order. You can also add more items to your order by selecting 'Continue Shopping'.

Order Information

ORDER TOTAL	SHIPPING ADDRESS	BILLING ADDRESS
\$10.99	David Grover 123 Main Street Apt. C Raleigh, NC 27604 919-555-1234	David Grover 123 Main Street Apt. C Raleigh, NC 27604 919-555-1234

Order Details

ITEM #	ITEM DESCRIPTION	PACKAGING	QUANTITY	PRICE	SUBTOTAL
V0002	Ornamental Gourd	1 pkt. (100 seeds)	1	\$1.50	\$1.50
A0008	Gloves	3 pairs per pack	1	\$4.50	\$4.50

Order Subtotal: \$6.00

Shipping, Standard Ground (3 to 6 business days) \$4.99: \$4.99

Order Total: \$10.99

[Continue Shopping](#) **Submit Order**

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-11. Checking out: Submit

When the billing information is entered, the user is able to click **Submit Order**.

- Click **Submit Order** to complete the transaction

The Help page provides an entry point to the following links:

- View Server Info
- Admin Home

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-12. Help page

The Help page provides an entry point to the following links: View Server Info and Admin Home. These links are added to the PlantsByWebSphere application especially for this course. They are not part of the PlantsByWebSphere EAR file that is distributed with the product.

IBM Training IBM

View Server Info page

The **View Server Info** page shows which server is hosting the connection

- Used for demonstrating server failover

It also displays the **Session Data** (and time created)

- Used for demonstrating session failover

Introduction to the PlantsByWebSphere application © Copyright IBM Corporation 2016

Figure 4-13. View Server Info page

The View Server Info page shows which server is hosting the current connection. This information is useful for demonstrating server failover, since it is important to know which cluster member to stop.

The session data is used to demonstrate failover of session information through memory-to-memory replication.

The PlantsByWebSphere application does not store the shopping cart in an HTTP session object, so it does not fail over correctly upon a server failure. Instead, this session information field was created to demonstrate HTTP session object failover.



Admin home page

PLANTS BY WEBSHIRE

HOME : ADMIN HOME

[Manage BackOrders](#) - View backorder inventory, order from suppliers, add new stock to inventory.

[Supplier Configuration](#) - Configure the Supplier.

PLANTS BY WEBSHIRE

[Admin Home](#)

Supplier Configuration

Enter the Supplier's Configuration Information

Full Name	<input type="text" value="Greenhouse By WebSphere"/>
Street Address	<input type="text" value="4205 Miami Blvd."/>
City	<input type="text" value="Durham"/>
State	<input type="text" value="NC"/>
Zip	<input type="text" value="27709"/>
Phone	<input type="text" value="919-555-1212"/>
Location URL	<input type="text" value="http://localhost:9080/OrderProcessorEJB/services/FrontGate"/>
<input type="button" value="Update Configuration"/>	

- The Admin home page is used to demonstrate application security
- When configured, these pages require authentication

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-14. Admin home page

The administrative pages are mapped to the SampAdmin security role. When application security is enabled, authentication is required to access these pages.



HTML documentation for PlantsByWebSphere (1 of 2)

- For more information about PlantsByWebSphere, several HTML files in the sample directory can be useful
- Overview:**
`<was_root>/samples/PlantsByWebSphere/docs/index.html`



Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-15. HTML documentation for PlantsByWebSphere (1 of 2)

The following files contain potentially useful information about PlantsByWebSphere:

- `<was_root>/samples/PlantsByWebSphere/docs/index.html`
- `<was_root>/samples/PlantsByWebSphere/docs/techNotes.html`



HTML documentation for PlantsByWebSphere (2 of 2)

- **TechNotes:**

<was_root>/samples/PlantsByWebSphere/docs/techNotes.html

The screenshot shows a Mozilla Firefox window with the title bar "Plants By WebSphere Sample TechNotes - Mozilla Firefox". The address bar displays "file:///C:/Program Files/IBM/WebSphere/AppServer/samples/PlantsByWebSphere/docs/techNotes.html". The main content area has a blue header bar with the text "Plants By WebSphere Sample Technotes". Below the header, there is a sidebar with links: "Overview", "Getting Started", "Development and Build", and "Application Install and Management". The "Getting started" section contains text about creating a unique email account and a table with default login information.

E-mail address	plants@plantsbywebsphere.ibm.com
Password	plants

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-16. HTML documentation for PlantsByWebSphere (2 of 2)

The Technotes page provides information about the default email address and password. Also, the database tables and SQL statement for creating them are shown. A description of the Java objects that are used in the application is shown.

The Plants by WebSphere Sample incorporates the following technologies:

- Java Persistence API (JPA) entity beans
- Stateless session beans
- Stateful session beans
- Servlets
- JavaServer Faces (JSF) files and Facelets
- Java Platform, Enterprise Edition security

The Plants by WebSphere application is supported through a series of JSF pages and HTML pages. These pages communicate with the following servlets: AccountServlet, ShoppingServlet, ImageServlet, and AdminServlet. The servlets use the various enterprise bean business methods which, in turn, access data from the database as needed.

Unit summary

- Describe the architecture and components of the PlantsByWebSphere application
- Explain how the application is used as a case study for WebSphere Application Server

Introduction to the PlantsByWebSphere application

© Copyright IBM Corporation 2016

Figure 4-17. Unit summary

Unit 5. Application assembly

Estimated time

01:00

Overview

In this unit, you learn how to use the WebSphere Developer Tools for Eclipse to prepare and export an enterprise application for deployment to WebSphere Application Server.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server Network Deployment V9 documentation in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Describe the functions of the WebSphere Developer Tools for Eclipse
- Describe the application assembly process
- Describe the use of the WebSphere Developer Tools for Eclipse, including:
 - Importing and examining application components
 - Preparing and exporting an enterprise application for deployment on WebSphere Application Server
 - Explaining how annotations work and describing their benefits for programmers
 - Showing the metadata that annotations generate
 - Explaining the relationship between annotations and deployment descriptors

Application assembly

© Copyright IBM Corporation 2016

Figure 5-1. Unit objectives

Topics

- Overview of Java EE application packaging
- Application assembly and deployment tools
- Enhanced EAR
- Java EE modules
- Java EE annotations

Application assembly

© Copyright IBM Corporation 2016

Figure 5-2. Topics

5.1. Overview of Java EE application packaging

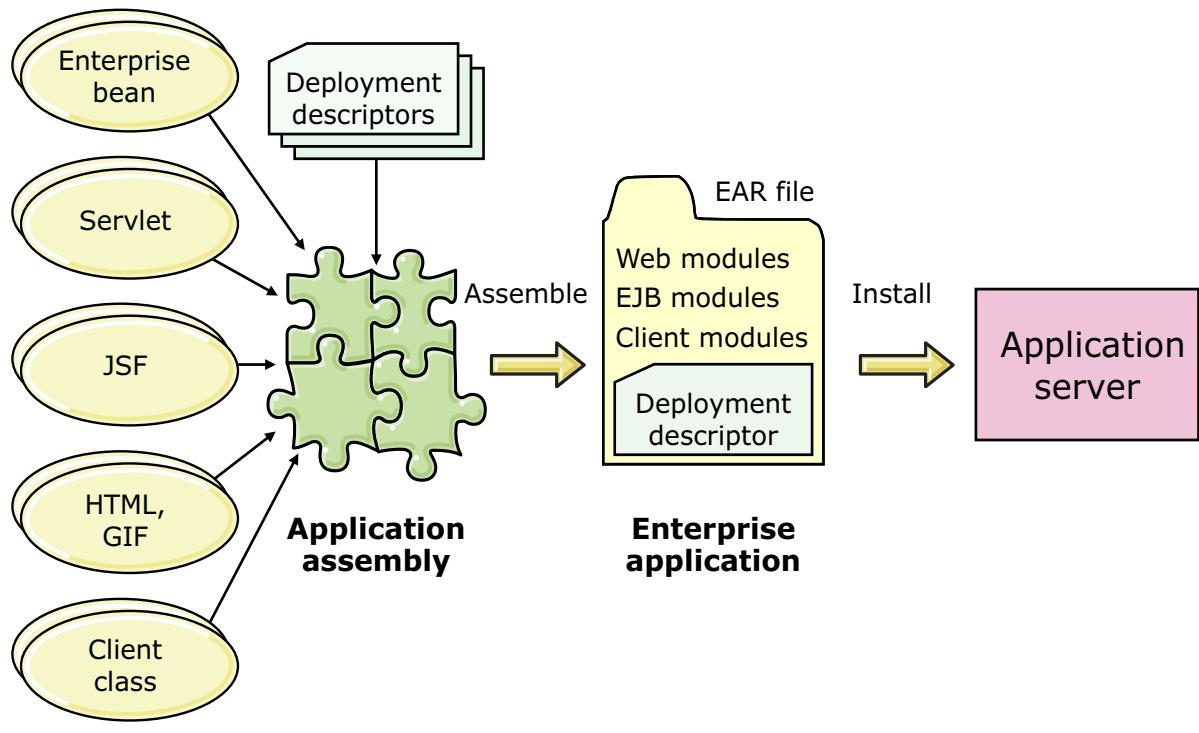
Overview of Java EE application packaging

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 5-3. Overview of Java EE application packaging

Overview of application assembly and installation



Application assembly

© Copyright IBM Corporation 2016

Figure 5-4. Overview of application assembly and installation

Two main activities are shown in this figure: assembly and installation. This unit describes the assembly process, and a following unit defines the installation activity.

Java EE packaging

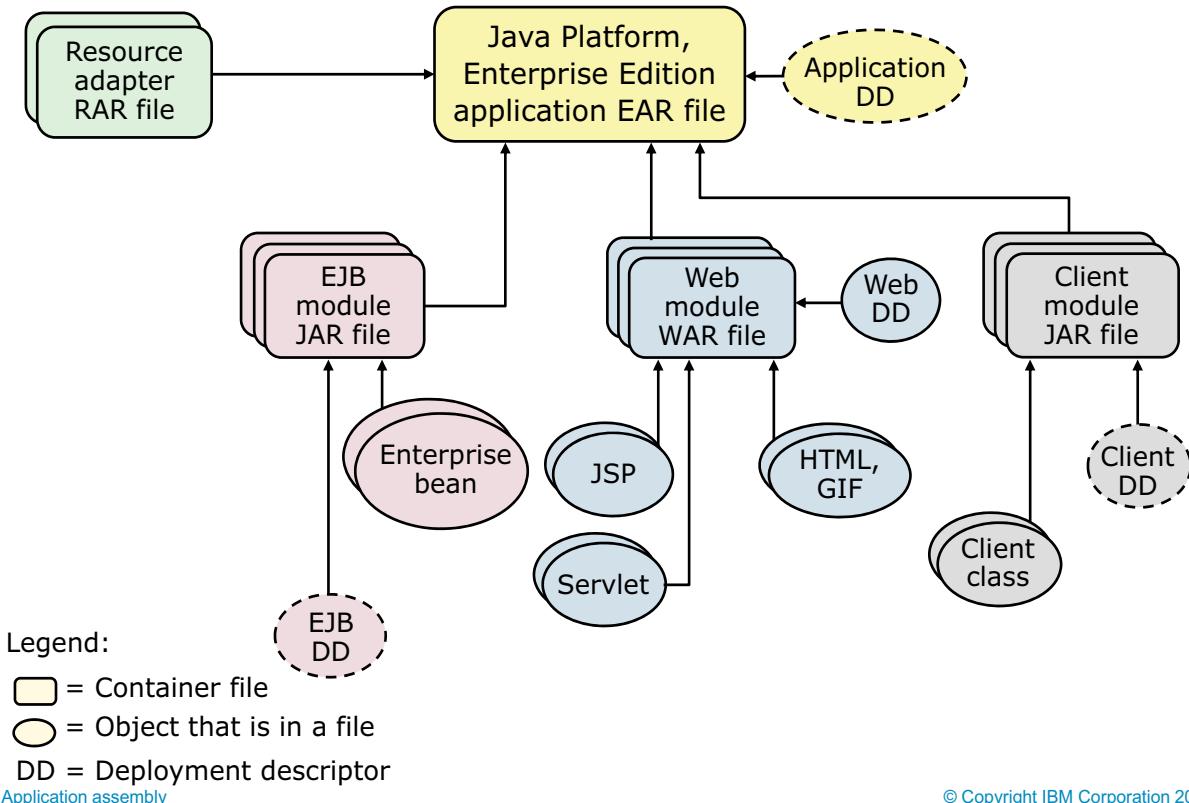


Figure 5-5. Java EE packaging

This diagram shows the pieces that can be put together to create a Java EE application.

WebSphere: Application packaging

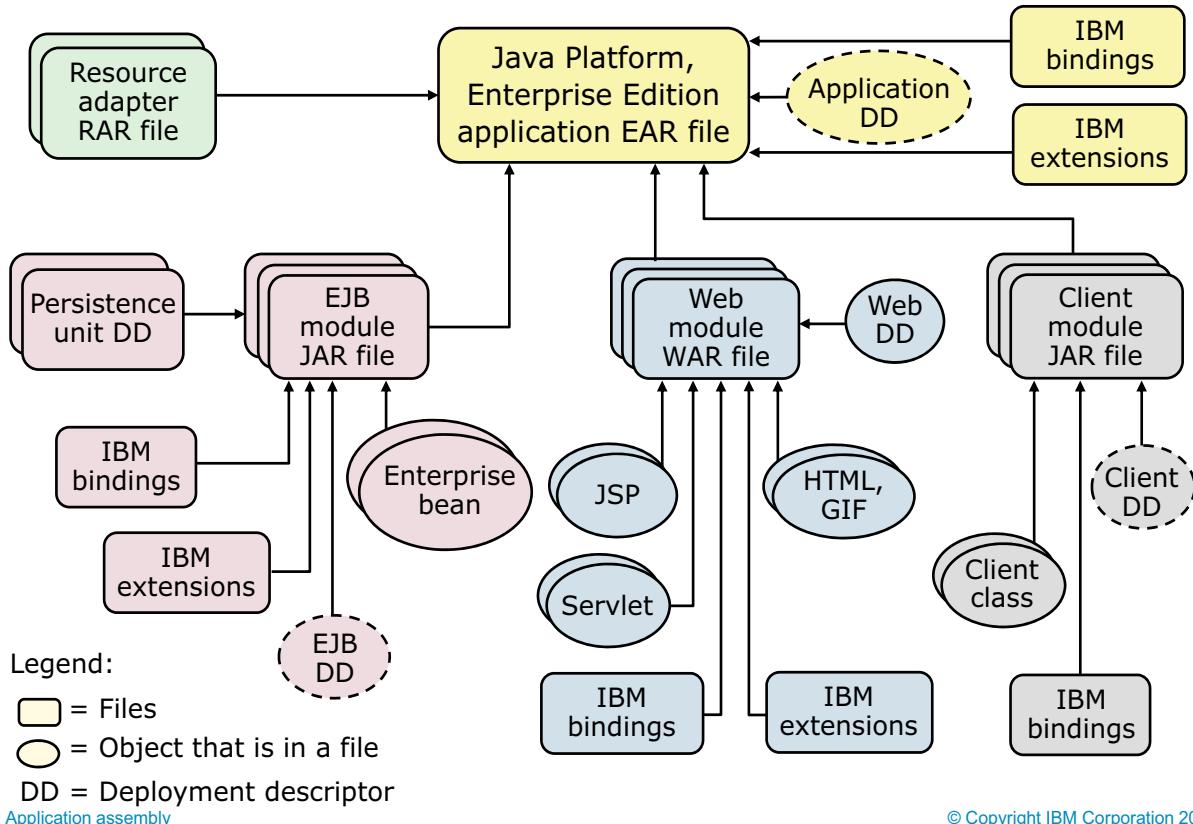


Figure 5-6. WebSphere: Application packaging

See how this diagram adds more items to the enterprise application (Hint: compare it to the previous diagram). These additional items are required for the WebSphere Application Server to “understand” how this application is going to run.

WebSphere application contents

Deployment descriptors

- XML-based text files that describe the application environment

IBM bindings

- Bind application names to deployment platform-specific resources
 - User-to-role mappings

IBM extensions

- Support other options, beyond the Java EE specification, such as:
 - Access intent attributes
 - Web application reloading
 - File serving and servlet invoker (by class name)

Application assembly

© Copyright IBM Corporation 2016

Figure 5-7. WebSphere application contents

A deployment descriptor is an Extensible Markup Language (XML) file that describes how to deploy a module or application by specifying configuration and container options.

Before an application can start, all enterprise bean (EJB) references and resource references that are defined in the application must be bound to the actual artifacts (enterprise beans or resources) defined in the application server.

When defining bindings, you specify Java Naming and Directory Interface (JNDI) names for the referenceable and referenced artifacts in an application. The jndiName values that are specified for artifacts must be qualified lookup names. An example of a referenceable artifact is an EJB defined in an application. An example of a referenced artifact is an EJB or a resource reference that the application uses.

5.2. Application assembly and deployment tools

Application assembly and deployment tools

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 5-8. Application assembly and deployment tools

Development and assembly tools

- Use an integrated development environment (IDE) to develop, assemble, and deploy Java EE modules for WebSphere Application Server
- The assembly tool that is used in this course is:
 - IBM WebSphere Application Server Developer Tools for Eclipse, which is a lightweight set of tools for developing, assembling, and deploying Java EE applications to WebSphere Application Server V7 and later
- In the IBM Knowledge Center, the phrase *assembly tool* refers to this tool and others that are used to develop, assemble, and deploy Java EE applications

Application assembly

© Copyright IBM Corporation 2016

Figure 5-9. Development and assembly tools

IBM WebSphere Application Server Developer Tools for Eclipse is a workbench that integrates with the application server to help you to quickly deploy and test applications. This product contains wizards and visual editors that support the Java EE programming model. The download for IBM WebSphere Application Server Developer Tools for Eclipse is available on the WASdev community website at: <http://developer.ibm.com/wasdev>

IBM WebSphere Application Server Developer Tools for Eclipse

- IBM WebSphere Application Server Developer Tools for Eclipse is an Eclipse plug-in
 - A subset of Rational Application Developer for the rapid assembly and deployment of modules for WebSphere
 - Requires Eclipse workbench
- Runs on Linux Intel, Windows, and Mac OS platforms
- Replaces the IBM Assembly and Deploy Tools for WebSphere Administration
- Can be downloaded from Eclipse Marketplace; can be purchased as part of a WebSphere Application Server bundle
- For more information, see the following website:
<https://www.ibm.com/software/webservers/appserv/was/tools/>

Application assembly

© Copyright IBM Corporation 2016

Figure 5-10. IBM WebSphere Application Server Developer Tools for Eclipse

IBM WebSphere Application Server Developer Tools for Eclipse enables rapid assembly and deployment of applications to WebSphere Application Server environments. These tools replace the previously available IBM Assembly and Deploy Tools function.

IBM WebSphere Application Server Developer Tools for Eclipse tasks

- Create and configure Java EE enterprise applications (EAR files):
 - Build from scratch
 - Java EE modules
- Generate and modify deployment descriptor information
- Generate and modify binding information attributes
- Generate and modify the IBM extension attributes
- Deploy applications to a remote server
- Create, debug, and run Jython scripts
- Import command assistance logs from the console in to Jython scripts
- View, analyze, and correlate log files

Application assembly

© Copyright IBM Corporation 2016

Figure 5-11. IBM WebSphere Application Server Developer Tools for Eclipse tasks

IBM WebSphere Application Server Developer Tools for Eclipse provides all of the core functions for assembling and deploying a Java EE application.

Features in assembly tools (1 of 2)

- Tools for publishing server-side code on WebSphere Application Server
- Automated deployment descriptor generation and visual editors
- Jython scripting editor and debugger
- Java Platform, Enterprise Edition and Java EE XML form-based deployment descriptor and binding editors
- Tools for assembling and deploying OSGi applications, bundles, fragments, and composites and a source editor for blueprint file editing
- Tools for JAX-RPC web service import, discovery, and deployment
- Tools for JAX-WS web service deployment, policy set association, and binding

Application assembly

© Copyright IBM Corporation 2016

Figure 5-12. Features in assembly tools (1 of 2)

Features in assembly tools (2 of 2)

- XML deployment descriptor, binding editors, and code validators
- Tools for importing, exporting, assembling, and deploying Session Initiation Protocol (SIP) applications
- Tools for assembling and deploying portlet applications
- Tools for adding database access to your applications, including built-in support and JDBC providers for many supported databases
- EJB deployment
- Enhanced EAR editor

Application assembly

© Copyright IBM Corporation 2016

Figure 5-13. Features in assembly tools (2 of 2)

IBM Assembly and Deploy Tools supports the following WebSphere Application Server Version 8 applications for assembly and deployment purposes:

- Java EE
- Basic OSGi
- Web services
- XML
- Basic SIP
- Basic Portlet

Assembling an enterprise application

- When working with a workspace that an application developer provides, no assembly is required (the tool already automatically does the assembly)
- The application assembler must:
 - Configure modules
 - Export an EAR file
- When you are assembling individual modules:
 - Create a workspace (first time)
 - Import modules into the workspace
 - Assign modules to an enterprise application
 - Configure the deployment descriptors of the module
 - Configure the Java EE module dependencies
 - Export the EAR file
- After assembly:
 - Optionally test within the tool by using a V9 test server

Application assembly

© Copyright IBM Corporation 2016

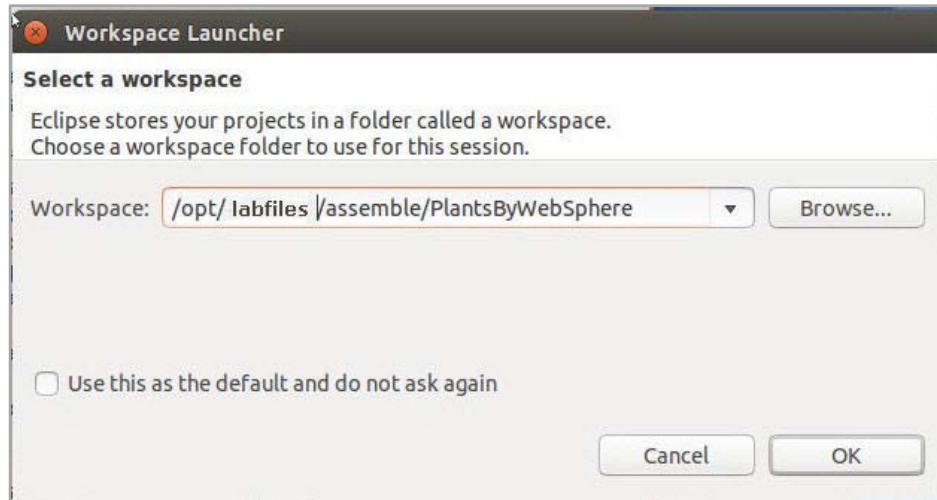
Figure 5-14. Assembling an enterprise application

The final product of the application assembly process is an EAR file.

IBM Training IBM

IBM WebSphere Application Server Developer Tools for Eclipse usage

- Upon opening the tool, you are asked to specify a workspace folder
 - Files and metadata are kept in the workspace
 - Create a workspace by pointing to an empty folder
 - Guideline is one workspace per enterprise application



Application assembly

© Copyright IBM Corporation 2016

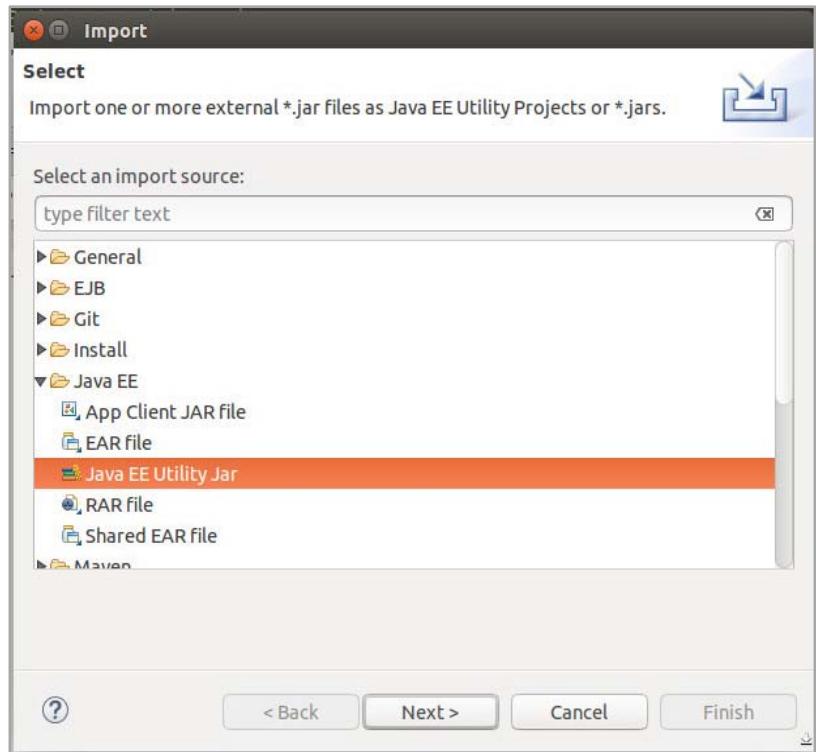
Figure 5-15. IBM WebSphere Application Server Developer Tools for Eclipse usage

With IBM WebSphere Application Server Developer Tools for Eclipse you get access to the complete set of Rational Application Developer documentation. Some documented features are available only with the full Rational Application Developer for WebSphere Software product. The tool documentation that is included with the WebSphere Application Server V9 documentation in IBM Knowledge Center is a supplement to the full Rational Application Developer for WebSphere Software documentation.

Import modules

Wizard driven:

- Import modules
 - EAR files
 - EJB JAR files
 - Application client JAR files
 - Web module WAR files
- Import Java Utility JAR files
- Imported into a new or existing enterprise application



Application assembly

© Copyright IBM Corporation 2016

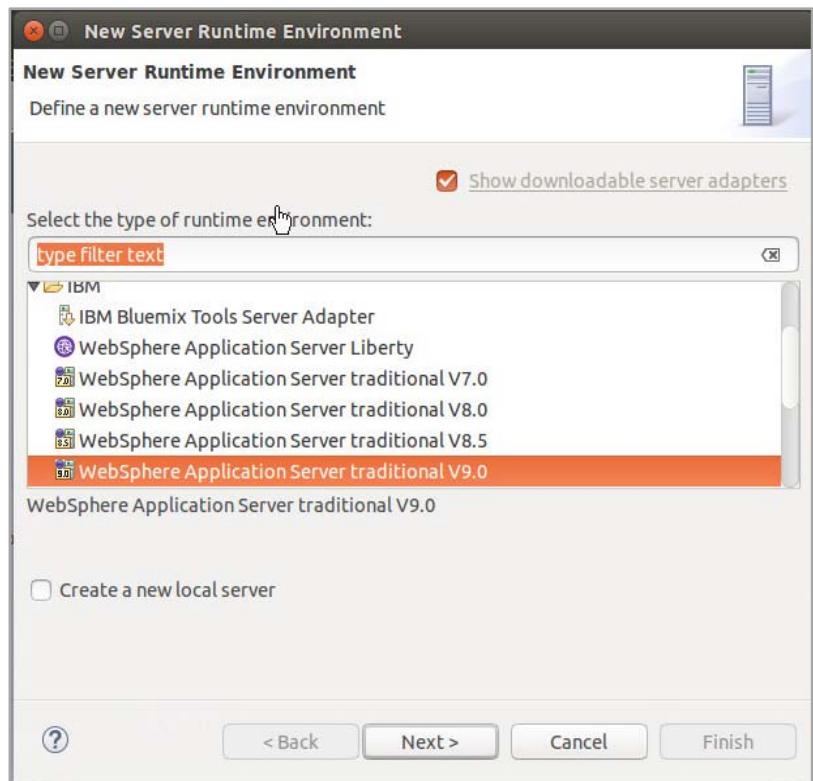
Figure 5-16. Import modules

Select **File > Import**.

Wizards are provided for importing application modules into the enterprise application.

Specify runtime environment

- Notice that different runtimes are available for different versions of WebSphere Application Server



Application assembly

© Copyright IBM Corporation 2016

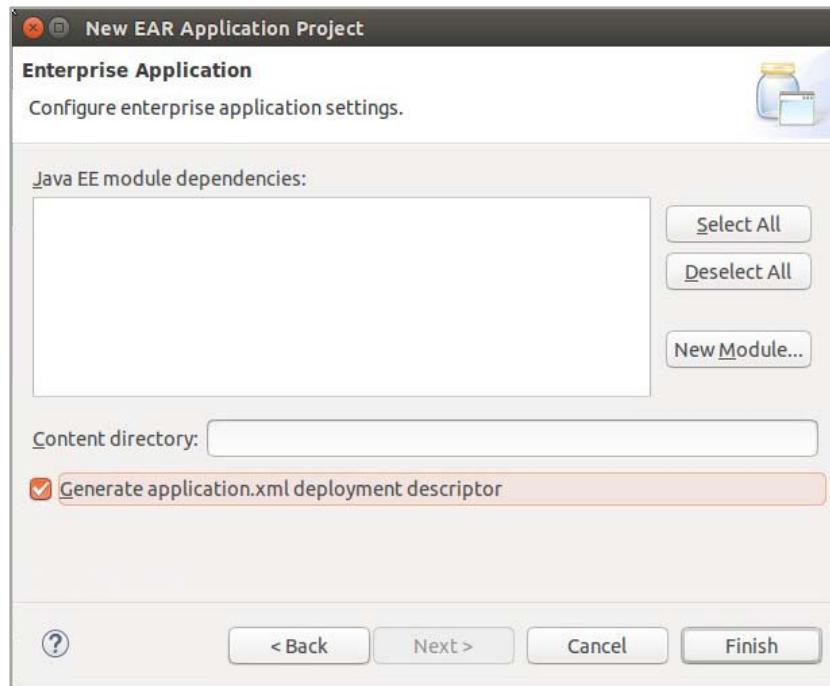
Figure 5-17. Specify runtime environment

This example screen capture shows the V9.0 runtime environment.



Generate deployment descriptor

- Check **Generate application.xml deployment descriptor**



Application assembly

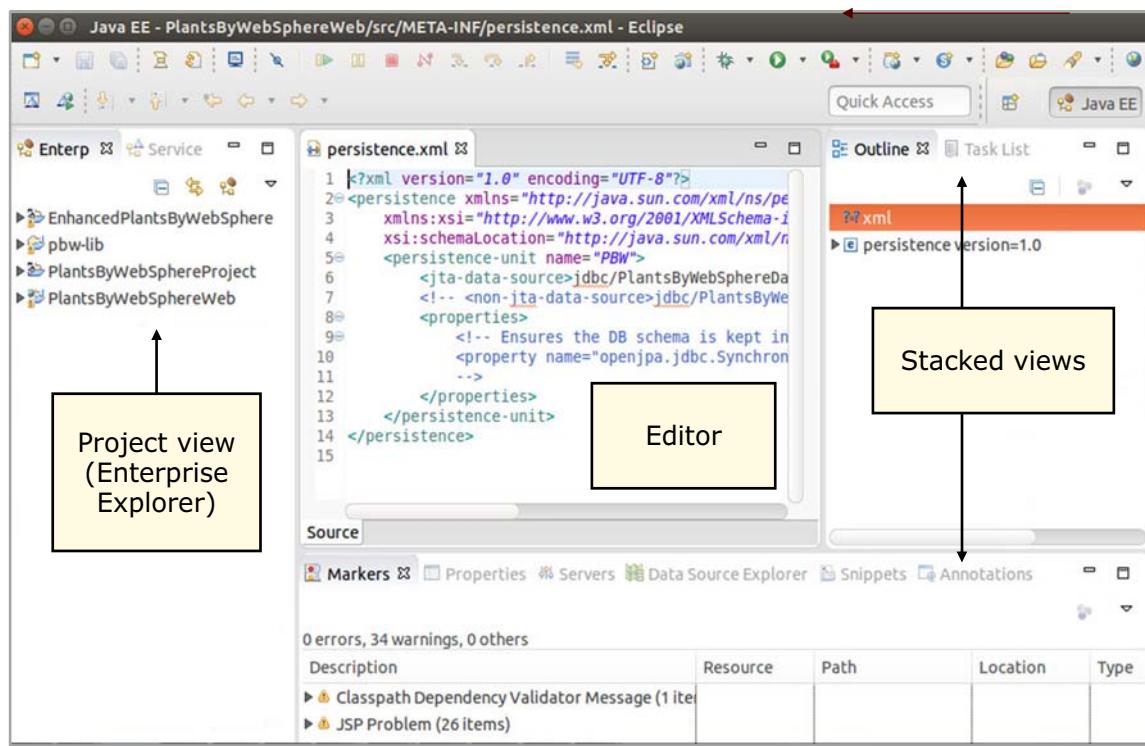
© Copyright IBM Corporation 2016

Figure 5-18. Generate deployment descriptor

When the `application.xml` deployment descriptor is automatically generated, default names are used.

IBM Training

Java EE perspective



Application assembly

© Copyright IBM Corporation 2016

Figure 5-19. Java EE perspective

This screen capture shows the primary perspective of IBM WebSphere Application Server Developer Tools for Eclipse for packaging Java EE applications.

Application deployment descriptor

Simplest of deployment descriptors (`application.xml`)

The editor can be used to:

- Edit the display name and description of the application
- Add and remove modules
 - Web
 - EJB
 - Application client
 - Resource adapter (connector)
- Work with security roles of the application

```
<?xml version="1.0" encoding="UTF-8"?><application id="Application_ID" version="6"
...
>
<display-name>PlantsByWebSphere</display-name>
<module id="Module_1320788080110">
  <web>
    <web-uri>PlantsByWebSphereWeb.war</web-uri>
    <context-root>PlantsByWebSphere</context-root>
  </web>
</module>
<security-role>
  <description>Samples Administrator</description>
  <role-name>SampAdmin</role-name>
</security-role>
</application>
```

Application assembly

© Copyright IBM Corporation 2016

Figure 5-20. Application deployment descriptor

The `application.xml` file identifies each module of an application. A Java EE application is not required to provide an `application.xml` file in the EAR file. When an `application.xml` file does not exist, the product examines the Java archive (JAR) file contents to determine whether the JAR file is an enterprise bean (EJB) module or an application client module.

Enterprise applications without application.xml

- Java EE enterprise applications can be built without specifying an application deployment descriptor (`application.xml`)
- The runtime product supplies a default deployment descriptor
 - The application name is assumed to be the name of the EAR file with the `.ear` extension removed
 - Files that end in `.war` are assumed to be web modules
 - The context root of the web module is assumed to be the name of the web module with the `.war` extension removed
 - Files not in the `/lib` directory that end in `.jar`, and contain an `ejb-jar.xml` file or at least one EJB, are assumed to be EJB modules

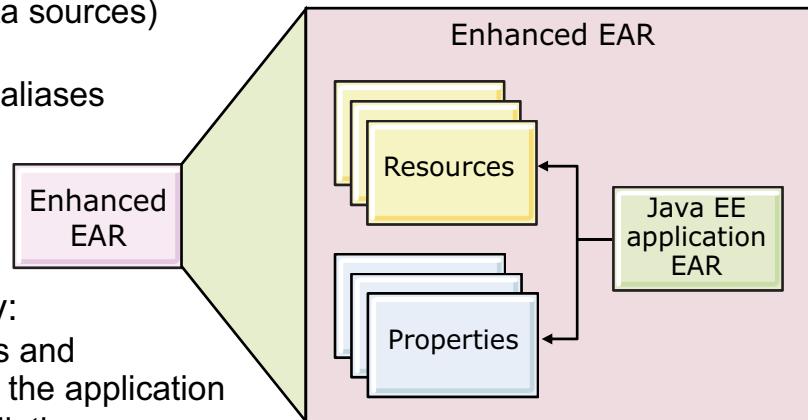
Note the limitation of the default context root name

Figure 5-21. Enterprise applications without application.xml

It is important to understand the limitation of the default context root name. For example, if the web module is named `PlantsByWebSphereWeb.war`, then the default context root is `PlantsByWebSphereWeb`.

Packaging enterprise applications for deployment

- You can deploy Java compliant **EAR** and **WAR** files
- An **enhanced EAR** includes Java EE artifacts plus resource information that is needed to install in the application server:
 - JDBC resources (data sources)
 - Class loader
 - JAAS authentication aliases
 - Shared libraries
 - Virtual host information
- Benefits in improved productivity:
 - Application resources and properties come with the application
 - The application installation process creates the necessary resources within the server or cluster
 - Moving an application from one server to another also moves the resources
- Assembly and deployment tools support WebSphere extensions



Application assembly

© Copyright IBM Corporation 2016

Figure 5-22. Packaging enterprise applications for deployment

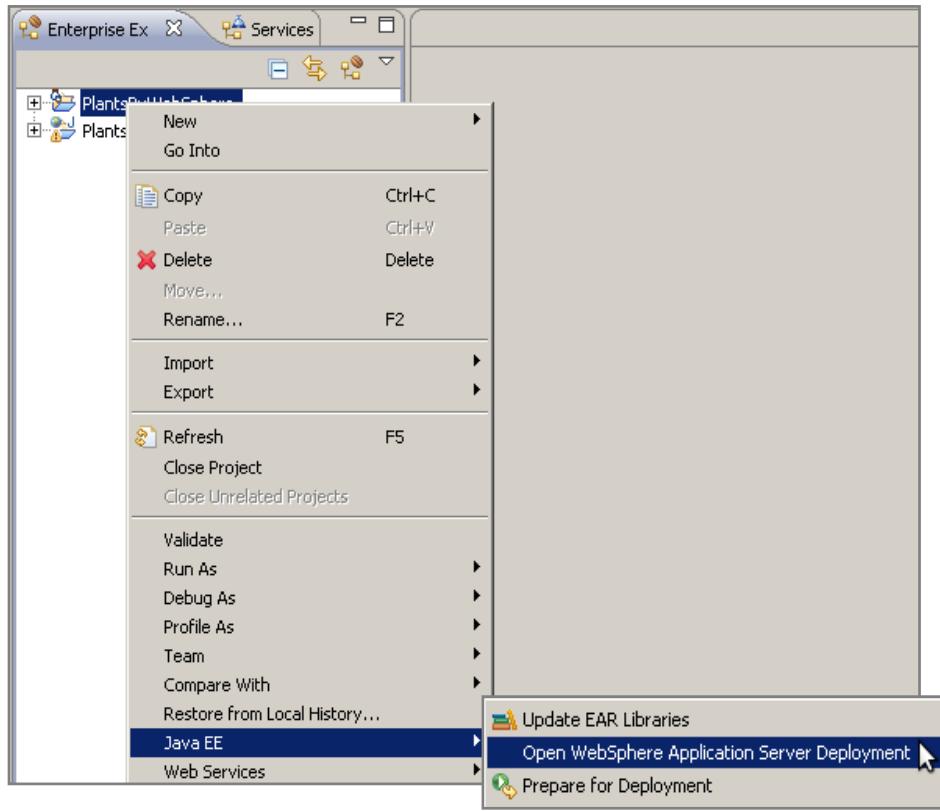
An enhanced EAR file is created if you use the administrative console to export an application.

To export applications, click **Export** on the Enterprise applications page. Using Export produces an enhanced enterprise archive (EAR) file that contains the application and the deployment configuration. The *deployment configuration* consists of the `deployment.xml` and other configuration files that control the application behavior on a deployment target.

By exporting applications, you can back up your applications and preserve binding information for the applications. You might export your applications before updating installed applications or migrating to a later version of the product.

Application scope resources

- Define resources to include in the enhanced EAR file



Application assembly

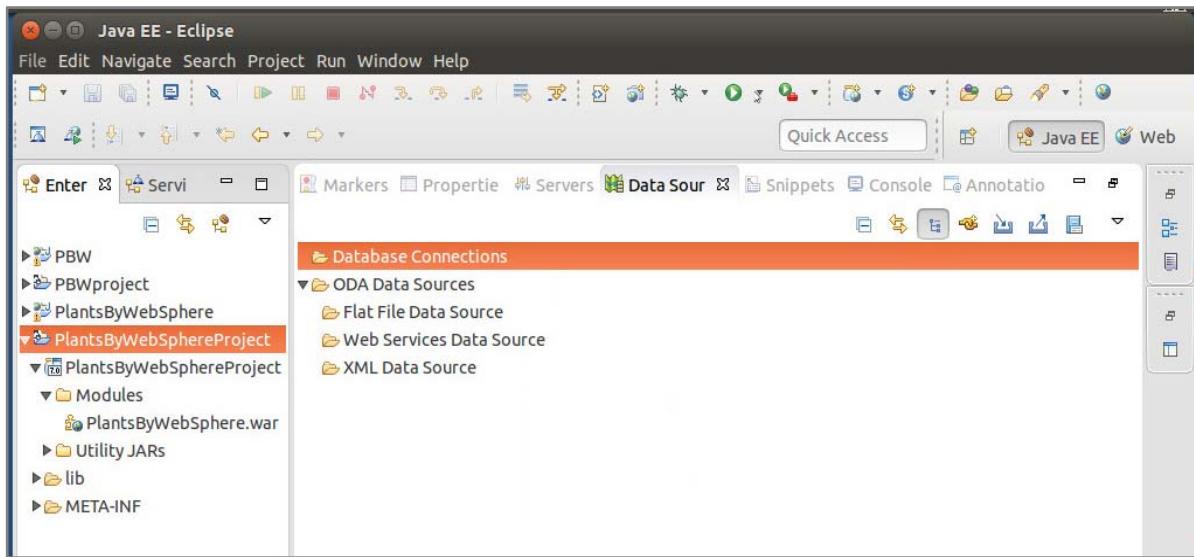
© Copyright IBM Corporation 2016

Figure 5-23. Application scope resources

This picture shows how to open the editor for resources that are going to be scoped at the application level. These types of resources are part of an enhanced EAR file. Later, more information is provided about enhanced EAR files.

IBM Training

Data sources



Application assembly

© Copyright IBM Corporation 2016

Figure 5-24. Data sources

This image shows part of the editing of an enhanced EAR, which is the entry point for defining a data source. You define a data source in the lab exercise. It is helpful to define a data source in the assembly tool so you can test the application without having to move it onto a server.

5.3. Enhanced EAR

Enhanced EAR

© Copyright IBM Corporation 2016

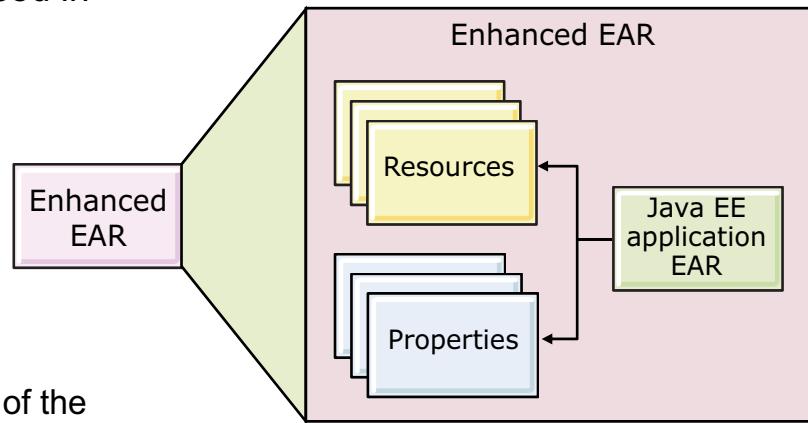
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 5-25. Enhanced EAR

Enhanced EAR

Not part of the Java EE specification

- Used mainly during development and test stages
- Not intended to be used in production
- Resources at application scope
- Resources are visible in the administrative console
 - Under the resources of the application, not with other scoped resources
 - Can be tested in the console (for example, test connection for data sources)
- Takes precedence over all other scopes



Application assembly

© Copyright IBM Corporation 2016

Figure 5-26. Enhanced EAR

Resources are defined at application scope, which takes precedence over higher-level scopes.

Unenhancing an EAR file

- Before deployment to production, remove resources that are defined in an enhanced EAR file
- Reconfigure resources to be at the correct scope
 - Preferably through scripting
- Resources can be removed:
 - By using the IBM Assembly and DeployTools (preferable)
 - Another .zip file-manipulating tool to remove the META-INF/ibmconfig folder
 - In UNIX systems, use the EARExander script:
 1. Expand the EAR file
 2. Remove only the META-INF/ibmconfig folder and all of its contents
 3. Compress the EAR file
 4. Deploy

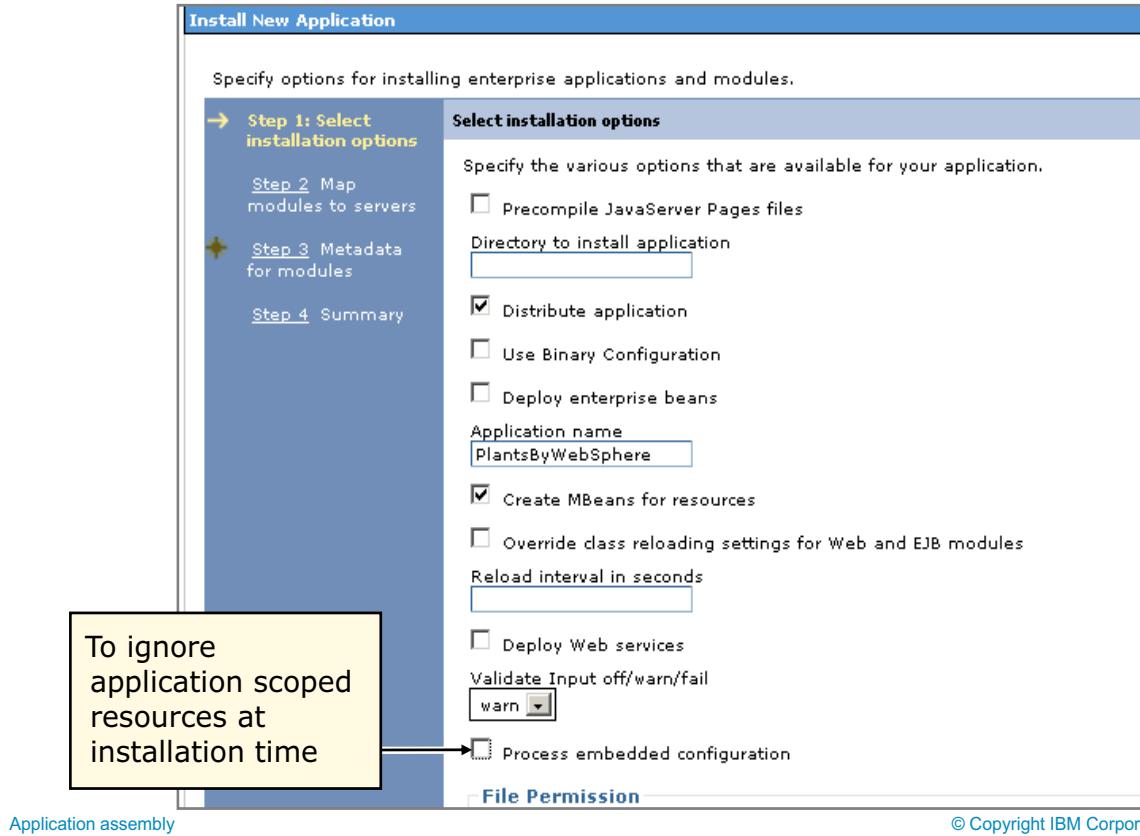
Application assembly

© Copyright IBM Corporation 2016

Figure 5-27. Unenhancing an EAR file

When an enhanced EAR file is uninstalled, the resources that are defined at the application scope are removed.

Dealing with enhanced EAR files at deployment time



Application assembly

© Copyright IBM Corporation 2016

Figure 5-28. Dealing with enhanced EAR files at deployment time

The check box named “Process embedded configuration” determines whether the application scoped resources are deployed to the server. If this box is checked, then the resources from the enhanced EAR file get used. If it is not checked, these resources are ignored and not installed in the WebSphere Application Server.

5.4. Java EE modules

Java EE modules

© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 5-29. Java EE modules

EJB module assembly

- Import wizard
 - Imports into a new or existing enterprise project
- Adds Enterprise JavaBeans in the module
 - Session bean
 - Entity bean
 - Message-driven bean
- Optionally use the deployment descriptor editor to configure
 - EJB references
 - Security roles
 - Method permissions
 - Application exceptions
 - Container transactions

Application assembly

© Copyright IBM Corporation 2016

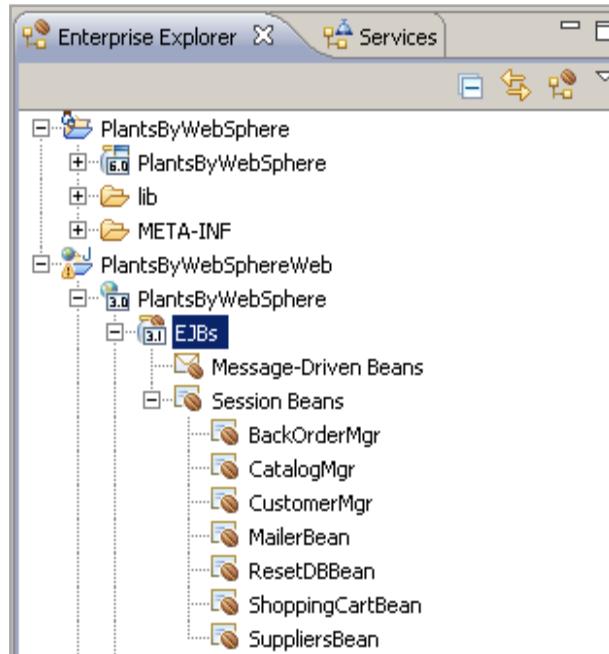
Figure 5-30. EJB module assembly

An enterprise bean is a managed Java component that can be combined with other resources to create Java Enterprise Edition (Java EE) applications.

Assemble an EJB 3.0 or 3.1 module to contain enterprise beans and related code artifacts. Group web components, client code, and resource adapter code in separate modules. After the EJB module is assembled, install it as a stand-alone application or combine it with other modules into an enterprise application.

EJB components are included in the web module

- In EJB 3.1, you can put enterprise bean classes in the WAR file along with web components
- No longer necessary to put EJB classes in the `ejb-jar` file



Application assembly

© Copyright IBM Corporation 2016

Figure 5-31. EJB components are included in the web module

This picture from the IBM Assembly and Deploy Tools shows an EJB JAR file that is packaged into a WAR module. According to the Java EE 6 specification, EJB files can now be packaged into the WAR file.

Web module assembly

- Import wizard
 - Imports into an existing enterprise application project, or creates a new one
 - Adds web components from a WAR module: servlets, JSP pages, static HTML, supporting Java code
- Use the deployment descriptor editor to configure
 - Named servlets and JSP pages (URL mappings)
 - Initialization parameters
 - Resource references
 - Security roles and constraints
 - Authentication mechanism
 - Welcome and error pages
 - Filters and listeners
- Context root is defined under **web_module > Properties > Web Project Settings**, not the deployment descriptor of the module

Application assembly

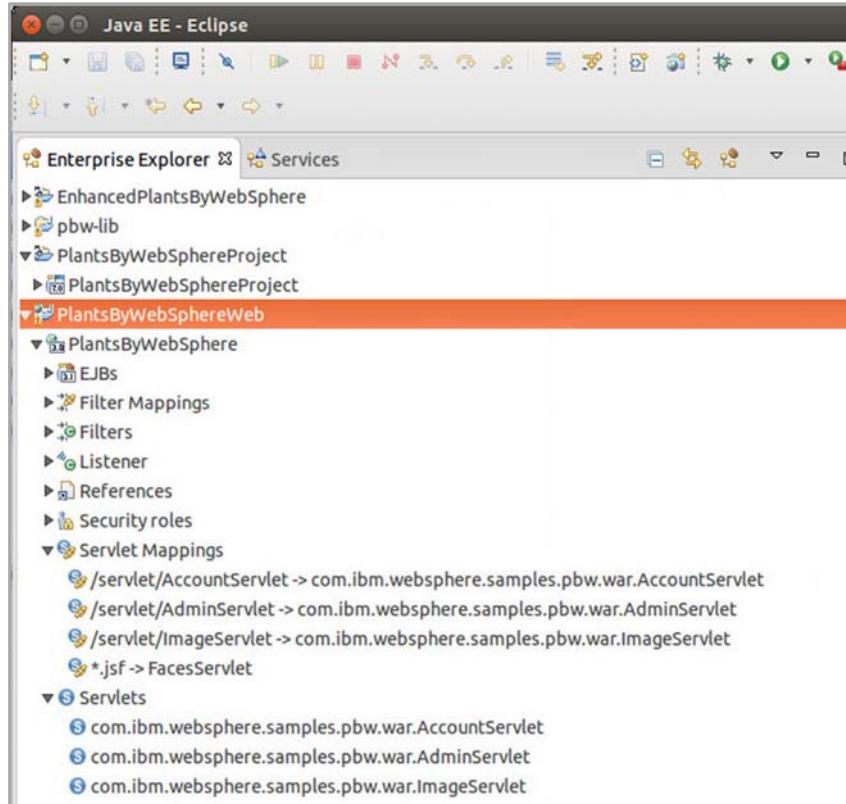
© Copyright IBM Corporation 2016

Figure 5-32. Web module assembly

The Import wizard imports into an existing enterprise application project, or creates a new one. You can use the wizard to add web components from a WAR module, such as servlets, JSP files, static HTML, and other supporting Java code.



Web module



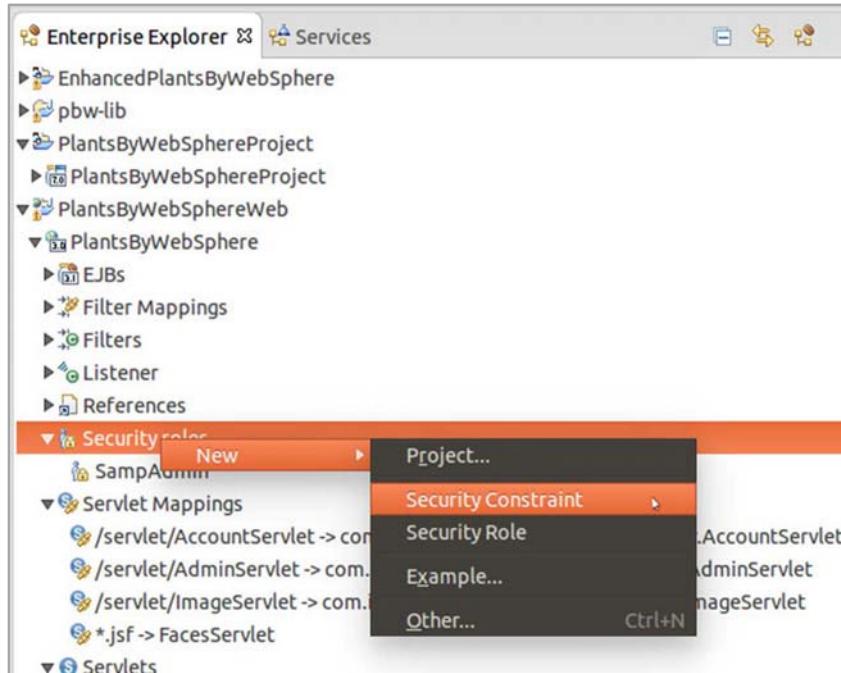
Application assembly

© Copyright IBM Corporation 2016

Figure 5-33. Web module

This screen gives a look at some of the files that are contained in the web module. This module is going to be exported as a WAR file.

Adding metadata through the menus



Application assembly

© Copyright IBM Corporation 2016

Figure 5-34. Adding metadata through the menus

By clicking an item in the Enterprise Explorer (typically a right-click), more information can be created or modified. In the example that is shown here, security data is going to be added to the web module.

Application client assembly

- Import wizard
 - Imports into existing enterprise application project, or creates a new one
 - Adds application client components from the JAR module
 - Specify the class path and the class that contains the `main()` method that the application client uses
- Add any icons
- Add EJB references
- Add resource references

Application assembly

© Copyright IBM Corporation 2016

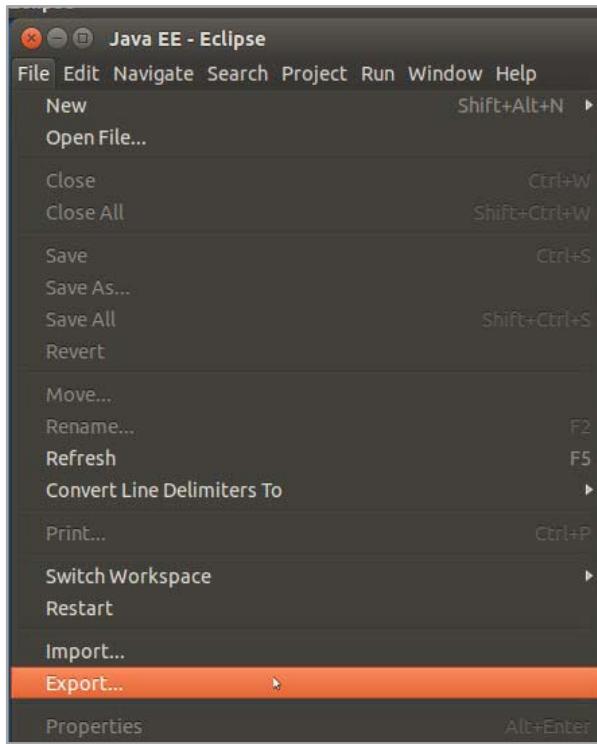
Figure 5-35. Application client assembly

Application client projects contain the resources that are needed for application client modules. An application client module is used to contain a full-function client Java application (non-web-based) that connects to and uses the Java EE resources that are defined in your server. When you place the client code in an application client module instead of a simple JAR file, the application client benefits from the resources of the application server. For example, it is not necessary to respecify the class path to Java EE and server JAR files. Also, JNDI lookup is easier since the client container provides the initial context and other parameters. By using the application client project, you can work as if you are creating a stand-alone Java application in a Java project.

You can use an application client project to do the following things:

- Develop the Java classes that implement the client module
- Set the application client deployment descriptor
- Test the application client

Generating an EAR file for deployment



- Assemble application modules
- Resolve Java EE dependencies
- Save all changes
- Export the EAR file
- If source is available, it can optionally be included in the EAR file
- The exported file is ready to be deployed

Application assembly

© Copyright IBM Corporation 2016

Figure 5-36. Generating an EAR file for deployment

Enterprise applications are deployed in the form of an EAR file. Use the Export wizard to export an enterprise application project into an EAR file for deployment.

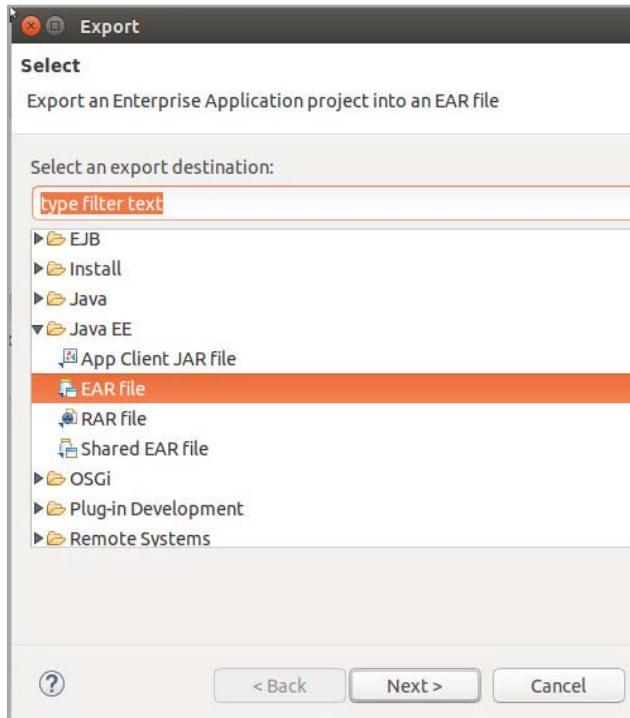
Optional: If you are exporting to an existing EAR file and you do not want to be warned about overwriting it, select the **Overwrite existing file** check box.

The wizard exports the contents of the EAR project to the specified EAR file. Additionally, for each project that corresponds to a module or utility JAR file in the application, the project contents are exported into a nested module or JAR file in the EAR file. If any unsaved changes exist on any of the files in any of the referenced projects, you are prompted to save these files before export.



Export an EAR file

- Specify the type of file to export



Application assembly

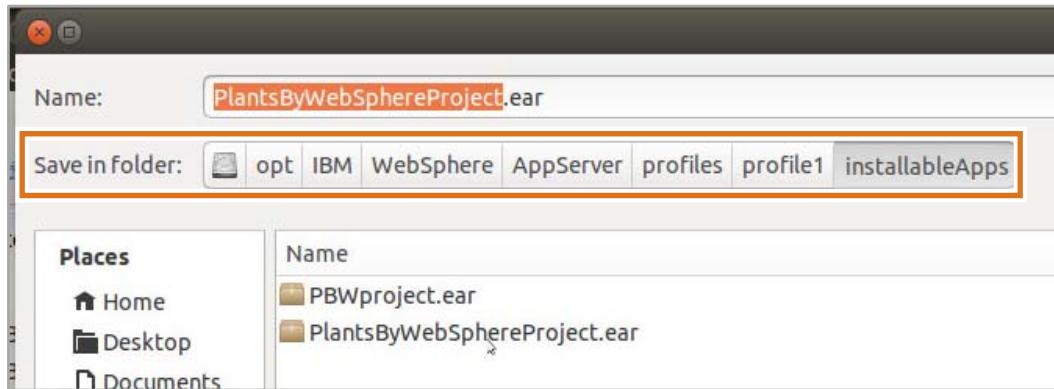
© Copyright IBM Corporation 2016

Figure 5-37. Export an EAR file

The second step of exporting an application is to specify the target file type: JAR, EAR, RAR, or shared EAR file.

Final export step

- Specify file name
- Place in the installableApps directory



Application assembly

© Copyright IBM Corporation 2016

Figure 5-38. Final export step

In the final step of the export, you name the exported file. You also specify the location for that file. In this case, the `installableApps` directory is used to make it easy to deploy the application.

5.5. Java EE annotations

Java EE annotations

© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 5-39. Java EE annotations

Annotations in Java code

- An important feature in Java EE and EJB is the use of annotations
 - Simplifies EJB development
 - Removes the need for the deployment descriptor for most purposes
- Annotations are metadata that can be embedded directly into the Java classes they describe
 - Allow application settings to be visible in the component they affect
 - You can use them to attach more information to a Java class or method
 - The server generates EJB infrastructure code that is based on annotations
- A combination of annotations and deployment descriptor can be used
 - The deployment descriptor augments or overrides the annotations
 - Allows for customization when application is assembled or deployed

```
@Stateless
@RolesAllowed ("SampAdmin")
public class BackOrderMgr {
    ...
}
```

Annotation syntax starts with @ ...

Application assembly

© Copyright IBM Corporation 2016

Figure 5-40. Annotations in Java code

You can use annotations to write metadata for Enterprise JavaBeans (EJB) inside your source code. You can use them instead of Extensible Markup Language (XML) deployment descriptor files. Annotations can also be used *with* descriptor files.

How annotations help the developer

- Using annotations reduces complexity:
 - Reduces the number of artifacts the developer must deal with
 - Metadata is in the code instead of the deployment descriptors
 - Default values can be used for most common scenarios
 - Developers must specify information only if they want the application to behave differently than the default
 - Plain old Java objects (POJO) plus annotation-based programming model

Application assembly

© Copyright IBM Corporation 2016

Figure 5-41. How annotations help the developer

EJB 3.0 represents a vast improvement in the EJB programming model and is one of the biggest potential sources of increased productivity for Java EE developers. An EJB can now be an annotated “plain old Java object” (POJO), which is not required to extend a certain class. It must implement a remote interface only, which you define or allow your IDE to create automatically. Deployment descriptors are no longer required because the EJB container can extract all that is necessary to know from the annotations on an EJB.

Example of declaring a session bean with annotations

```
package com.ibm.websphere.samples.pbw.ejb;
import javax.ejb.Stateless;
...
/**
 * The BackOrderMgr provides a transactional and secured
 * facade to access back order information. This bean no longer
 * requires an interface as there is one and only one implementation.
 */
@Stateless
@RolesAllowed ("SampAdmin")
public class BackOrderMgr
{
    @PersistenceContext(unitName="PBW")
    private EntityManager em;
...
}
```

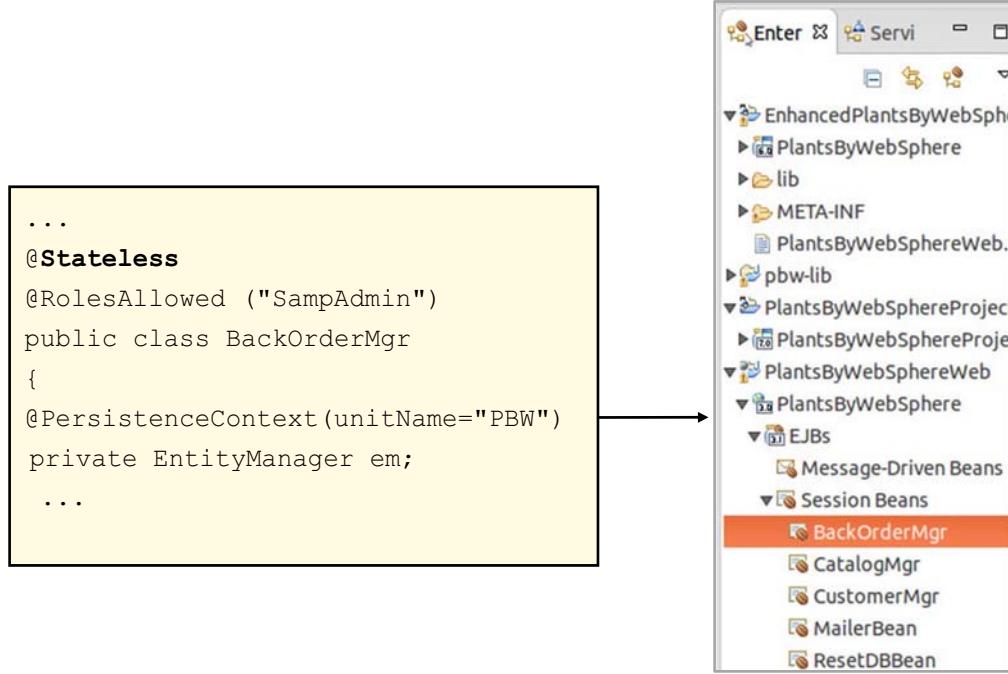
Application assembly

© Copyright IBM Corporation 2016

Figure 5-42. Example of declaring a session bean with annotations

In Java Platform, Enterprise Edition 1.4, this EJB had to implement the SessionBean interface, requiring six method implementations. In many cases, these method implementations wind up empty and exist only to satisfy the interface and allow the code to compile, leading to cluttered code. EJB 3.0 eliminates that problem by providing the lifecycle annotations `@PostConstruct`, `@PreDestroy`, `@PostActivate`, and `@PrePassivate`. You add these annotations as needed to any public, parameterless method that returns void to implement reactions to lifecycle events.

Stateless session bean as shown in the Explorer view



Application assembly

© Copyright IBM Corporation 2016

Figure 5-43. Stateless session bean as shown in the Explorer view

In the Enterprise Explorer view, you can see the hierarchy of the application, its modules, and the components of the modules.

Using annotations to inject EJB references

- Dependency injection:
 - Hides creation and lookup of resources from application code
 - Resources are “injected” into the source code by the container that is based on the annotations specified
 - Programmers are no longer required to do JNDI lookups in their code
 - Clients that are managed classes can access EJB components by declaring a dependency:

```
@EJB  
private CatalogMgr catalog;  
  
private ProductBean product;  
private LinkedList<ProductBean> products;  
private float shippingCost;
```

Application assembly

© Copyright IBM Corporation 2016

Figure 5-44. Using annotations to inject EJB references

Here you see how an annotation becomes application metadata as seen in the console.

Example of injecting an EJB reference

- Source

```
package com.ibm.websphere.samples.pbw.war;
import javax.ejb.EJB;
import javax.inject.Named;
...
@Named("help")
public class HelpBean {

    @EJB ResetDBBean rdb;
```

- Generated EJB references

The screenshot shows the 'Enterprise Applications' section of the administrative console. Under 'Enterprise Applications > PlantsByWebSphere > EJB references', there is a table listing one EJB reference:

Module	Bean	URI	Resource Reference
PlantsByWebSphere	ResetDBBean	PlantsByWebSphereWeb.war,WEB-INF/ejb-jar.xml	com.ibm.websphere.samples.pbw.ejb.ResetDBBean/cu

Application assembly

© Copyright IBM Corporation 2016

Figure 5-45. Example of injecting an EJB reference

The screen capture is from the administrative console. It shows a portion of the EJB references for the PlantsByWebSphere application. You can access this view by clicking **Applications > Application Types > WebSphere enterprise applications > PlantsByWebSphere > EJB > EJB references**.

Application bindings

- All EJB references and resource references that are defined in the application must be bound to the artifacts defined in the application server
 - These artifacts include enterprise beans or resources
- You specify Java Naming and Directory Interface (JNDI) names for the referenced artifacts in an application
- For EJB modules, binding definitions are stored in `ibm-ejb-jar-bnd.xml`
- For EJB modules, you do not need to specify JNDI binding names for each of the business interfaces on your enterprise beans
 - If you do not explicitly assign bindings, the EJB container assigns default bindings

Application assembly

© Copyright IBM Corporation 2016

Figure 5-46. Application bindings

An application assembler can define bindings when modifying deployment descriptors of an application. Bindings are specified in the WebSphere Bindings section of a deployment descriptor editor. Modifying the deployment descriptors might change the binding definitions in the `ibm-xxxx-bnd.xml` files that are created when developing an application. After defining the bindings, the assembler gives the application to a deployer. When installing the application onto a supported application server, the deployer does not modify, override, or generate default bindings unless changes are necessary for successful deployment.

WebSphere default bindings for bean interfaces

- WebSphere assigns default names that are based on patterns:

Description	Binding pattern
Short form local interfaces	<code>ejblocal:<package.qualified.interface></code>
Short form remote interfaces	<code><package.qualified.interface></code>
Long form local interfaces	<code>ejblocal:<component-id>#<package.qualified.interface></code>
Long form remote interfaces	<code>ejb/component-id>#<package.qualified.interface></code>

Application assembly

© Copyright IBM Corporation 2016

Figure 5-47. WebSphere default bindings for bean interfaces

An application deployer or server administrator can use the administrative console to modify the bindings when installing the application onto a supported application server. New binding definitions can be specified on the installation wizard pages.

Also, a deployer or administrator can select to generate default bindings during application installation. Selecting **Generate default bindings** during application installation instructs the product to define incomplete bindings in the application with default values. Existing bindings are not changed.

Specifying Java EE metadata for deployment

- Java EE modules can specify metadata:
 - In annotations
 - In deployment descriptors
 - If both, then deployment descriptors override or augment annotations
- EJB deployment descriptors (`ejb-jar.xml`) or web deployment descriptors (`web.xml`) are optional in Java EE EAR files
- You can view deployment descriptors in the WebSphere Application Server administrative console, even if you did not specify them
 - They are generated for you when you deploy the EAR in the application server

Application assembly

© Copyright IBM Corporation 2016

Figure 5-48. Specifying Java EE metadata for deployment

Bindings support in the EJB container is expanded in Java EE 6. The EJB container assigns default JNDI bindings for EJB 3.0 and 3.1 business interfaces. The bindings are based on application name, module name, and component name. You do not have to explicitly define JNDI binding names for each of the interfaces, or EJB homes within an EJB 3.0 module, or no-interface views within an EJB 3.1 module.

Unit summary

- Describe the functions of the WebSphere Developer Tools for Eclipse
- Describe the application assembly process
- Describe the use of the WebSphere Developer Tools for Eclipse, including:
 - Importing and examining application components
 - Preparing and exporting an enterprise application for deployment on WebSphere Application Server
 - Explaining how annotations work and describing their benefits for programmers
 - Showing the metadata that annotations generate
 - Explaining the relationship between annotations and deployment descriptors

Application assembly

© Copyright IBM Corporation 2016

Figure 5-49. Unit summary

Review questions

1. The result of packaging an enterprise application is _____. 
2. True or False: Java EE reduces or eliminates the need to deal with Java EE deployment descriptors in many cases.
3. True or False: Enhanced enterprise applications are the preferred way to deploy applications in a production environment.

Application assembly

© Copyright IBM Corporation 2016

Figure 5-50. Review questions

Write your answers here:

- 1.
- 2.
- 3.

Review answers

1. The result of packaging an enterprise application is _____.
The answer is: [An EAR file](#).



2. True or False: Java EE reduces or eliminates the need to deal with Java EE deployment descriptors in many cases.
The answer is True.

- Note: “Annotations reduce or eliminate the need to deal with Java EE deployment descriptors in many cases.” – *from the Java EE 6 specification*
- Caveat: If you do not specify the deployment descriptors, the product assumes certain default names.

3. True or False: Enhanced enterprise applications are the preferred way to deploy applications in a production environment.

The answer is False. Enhanced EAR files help the developer to test the application.

Exercise: Assembling an application

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 5-52. Exercise: Assembling an application

Exercise objectives

After completing this exercise, you should be able to:

- Explore the WebSphere Developer Tools for Eclipse
- Import and examine enterprise application components
- Define application-scoped resources: data source and authentication alias
- Export an enhanced EAR file that is ready for deployment

Application assembly

© Copyright IBM Corporation 2016

Figure 5-53. Exercise objectives

Unit 6. Application installation

Estimated time

00:45

Overview

In this unit, you learn ways to install and update enterprise applications on WebSphere Application Server.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server V9 documentation in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Describe methods of installing enterprise applications in WebSphere Application Server
- Explain how fine-grained application updates work
- Describe enterprise application properties
- Enable monitored directories
- Deploy an application by using the monitored directory
- Use a properties file-based configuration with monitored directories to deploy an application

Application installation

© Copyright IBM Corporation 2016

Figure 6-1. Unit objectives

Topics

- Application installation
- Application settings and interaction
- Monitored directory

Application installation

© Copyright IBM Corporation 2016

Figure 6-2. Topics

6.1. Application installation

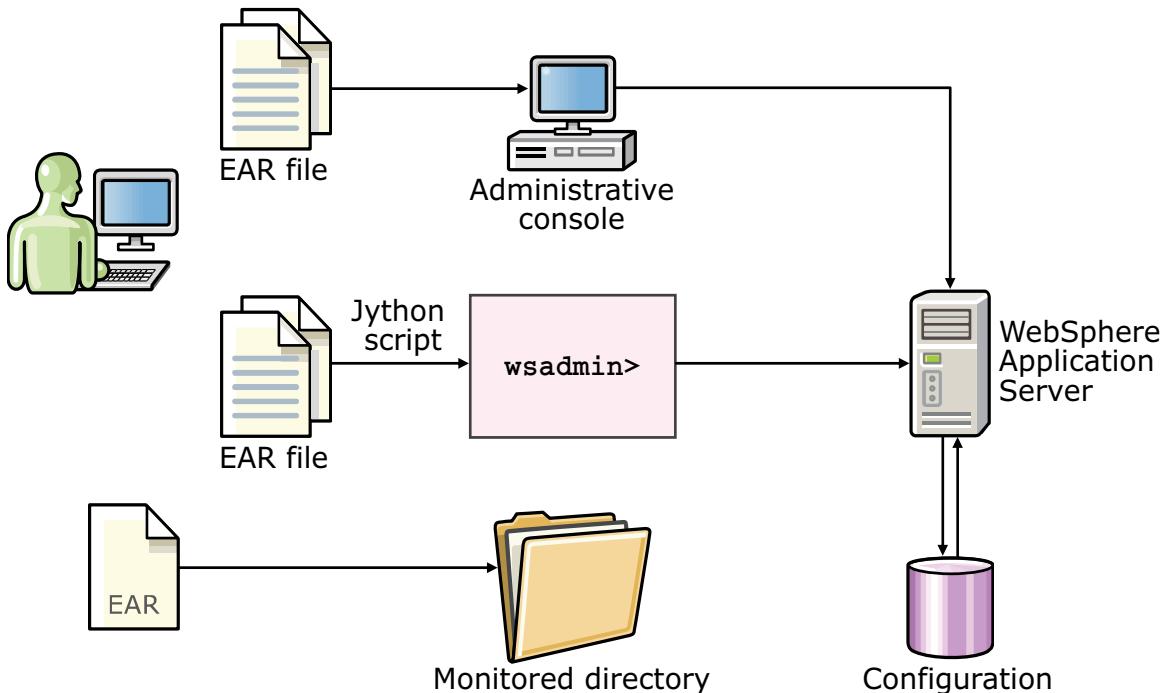
Application installation

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 6-3. Application installation

Installing enterprise applications



Application installation

© Copyright IBM Corporation 2016

Figure 6-4. Installing enterprise applications

This simple diagram shows the three paths for installing an EAR file (enterprise application) into the application server:

- From the administrative console
- Using a script or commands with wsadmin
- Dropping an archive file into a monitored directory

Installation tasks

- Configure the application environment as required
 - Variables, virtual hosts, class path, security
- Configure application resources
 - JDBC provider, data sources, JMS resources, or SIBus, if applicable
- Install the application
 - The default directory that is assigned to hold the EAR file before its installation is `<profile_root>\<profile>\installableApps`
 - Most often, the application file you receive is an enterprise archive (`.ear`) file
- Manage static content
 - Web server serves files, not in EAR file
 - Leave static content in EAR file

Application installation

© Copyright IBM Corporation 2016

Figure 6-5. Installation tasks

Installing an application involves configuring the runtime environment as required. You can define variables, virtual hosts, and any other resources that the application needs before you install the application. In general, it is a good practice to leave the static content that the application uses in the EAR file, and allow the infrastructure to take care of serving and caching the static content. In the end, leaving the static content in the EAR file can be as effective as moving it to the web server.

Creating a J2C authentication alias

- The wizard can be reached from many pages in the administrative console
 - Security > Global Security > Authentication > Java Authentication and Authorization Service > J2C authentication data**
- Provide
 - Alias name:
Console prefixes name with node name
 - User ID and corresponding password
 - Optional description
 - EJB components, data sources, JMS resources, and SIBus resources use them

Global security

Global security > JAAS - J2C authentication data > New...

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: PlantsApp

* User ID: db2inst1

* Password: *****

Description: For PlantsByWebSphere App

Apply OK Reset Cancel

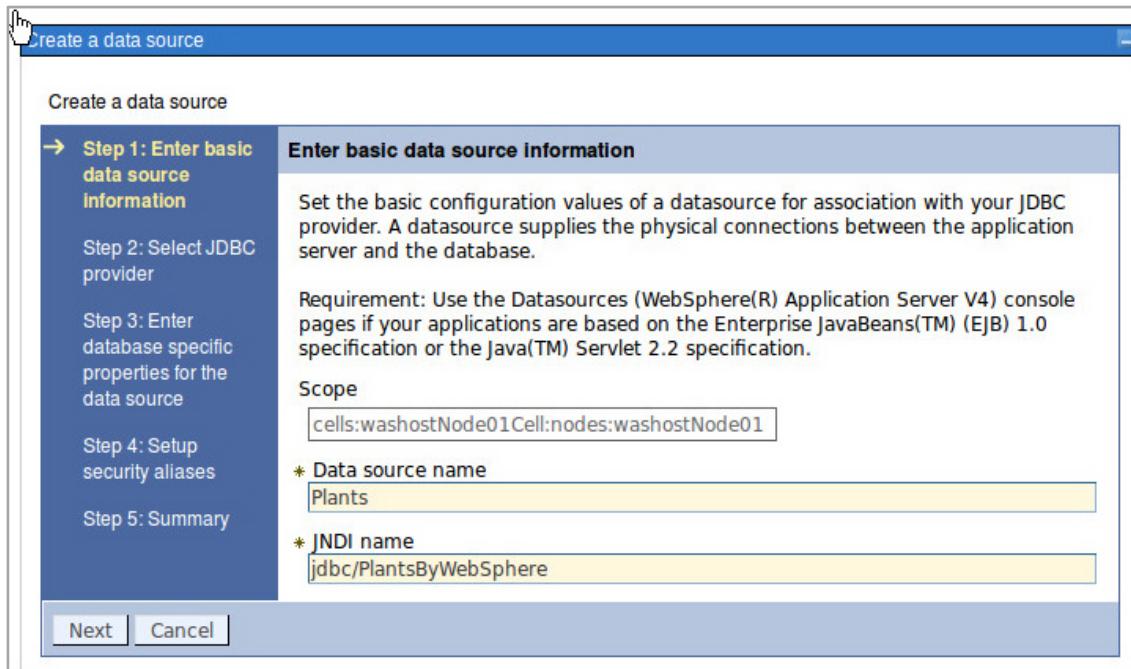
Application installation

© Copyright IBM Corporation 2016

Figure 6-6. Creating a J2C authentication alias

Here is the input screen for creating a security resource that is used to access a back-end resource.

Step 1: Enter basic data source information



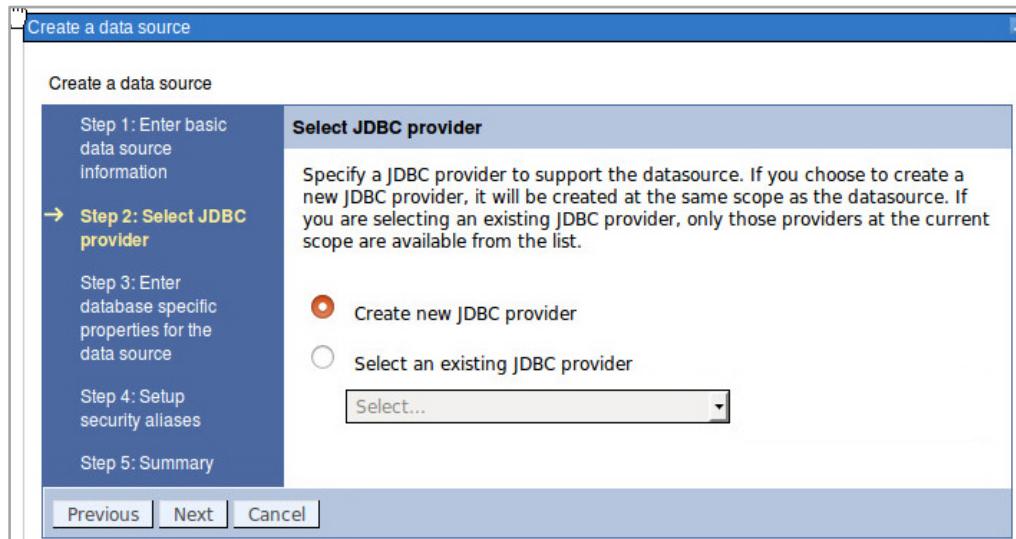
Application installation

© Copyright IBM Corporation 2016

Figure 6-7. Step 1: Enter basic data source information

These screens show the process for creating the required resources that an application uses to access a database. Here you see the setup for the database drivers (JDBC).

Step 2: Select JDBC provider



- Create JDBC provider before or while you are defining data sources
- One JDBC provider is needed for each database driver type
- JDBC providers can be defined at cell, node, server, or application scope (in an enhanced EAR file)

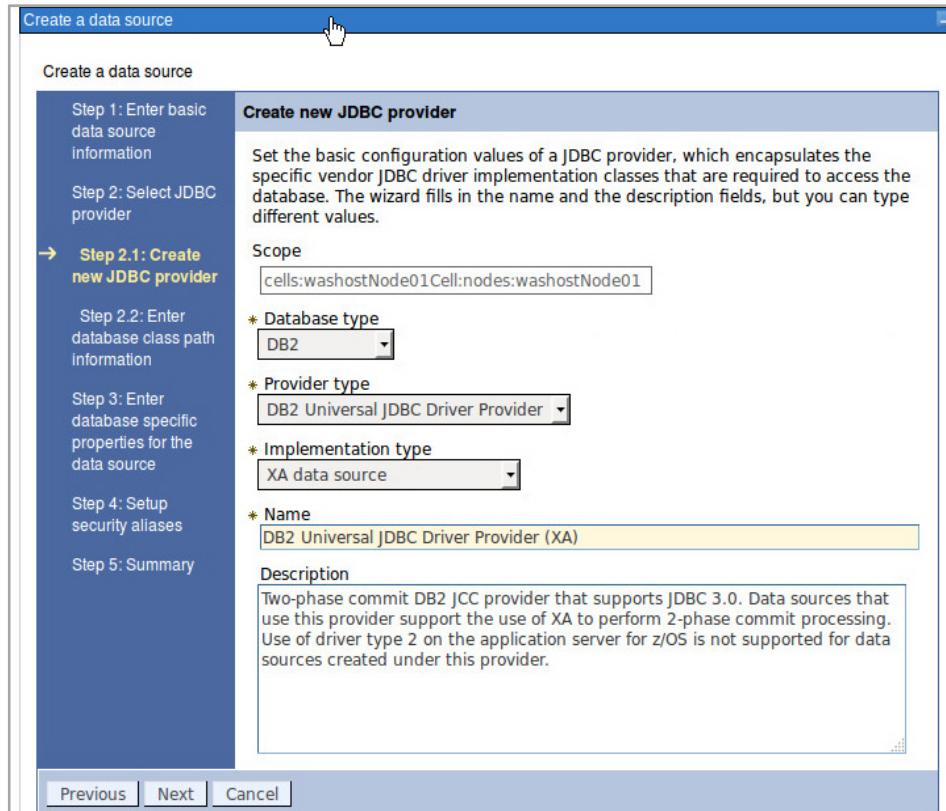
[Application installation](#)

© Copyright IBM Corporation 2016

Figure 6-8. Step 2: Select JDBC provider

These screens show the process for creating the required resources that an application uses to access a database. Here you see the setup for the database drivers (JDBC).

Step 2.1: Create new JDBC provider



Application installation

© Copyright IBM Corporation 2016

Figure 6-9. Step 2.1: Create new JDBC provider

These screens show the process for creating the required resources that an application uses to access a database. Here you see the setup for the database drivers (JDBC).



Step 2.2: Enter database class path information

Create a data source

Create a data source

Step 1: Enter basic data source information

Step 2: Select JDBC provider

Step 2.1: Create new JDBC provider

→ Step 2.2: Enter database class path information

Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

Enter database class path information

Set the class path for the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays a default list of jars and allows you to set the environment variables that define the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

Entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). If a value is specified for you, you may click Next to accept the value.

Class path:

```
$ {DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
$ {UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
$ {DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
```

JDBC driver paths can be defined in the wizard

Apply

Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is saved as WebSphere variable
\$ {DB2UNIVERSAL_JDBC_DRIVER_PATH}
/opt.ibm/db2/V10.5/java

Native library path

Directory location which is saved as WebSphere variable \$ {DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}
\$ {DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}
/opt.ibm/db2/V10.5/java

Previous | Next | Cancel

Application installation

© Copyright IBM Corporation 2016

Figure 6-10. Step 2.2: Enter database class path information

These screens show the process for creating the required resources that an application uses to access a database. These two screens show the definition of variables that hold the path to the database drivers.



Step 3: Enter database-specific properties for the data source

Create a data source

Create a data source

Enter database specific properties for the data source											
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.											
<table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>* Driver type</td> <td><input type="button" value="4"/></td> </tr> <tr> <td>* Database name</td> <td>PLANTS</td> </tr> <tr> <td>* Server name</td> <td>dbhost</td> </tr> <tr> <td>* Port number</td> <td>50000</td> </tr> </tbody> </table>		Name	Value	* Driver type	<input type="button" value="4"/>	* Database name	PLANTS	* Server name	dbhost	* Port number	50000
Name	Value										
* Driver type	<input type="button" value="4"/>										
* Database name	PLANTS										
* Server name	dbhost										
* Port number	50000										
<input checked="" type="checkbox"/> Use this data source in container managed persistence (CMP)											
<ul style="list-style-type: none"> • Provide the database-specific parameters: <ul style="list-style-type: none"> ▪ Driver type ▪ Database name ▪ Database server name and communication port number • Choose whether data source is going to be used with CMP beans 											

Step 1: Enter basic data source information
 Step 2: Select JDBC provider
 Step 2.1: Create new JDBC provider
 Step 2.2: Enter database class path information
 → Step 3: Enter database specific properties for the data source
 Step 4: Setup security aliases
 Step 5: Summary

Previous Next Cancel

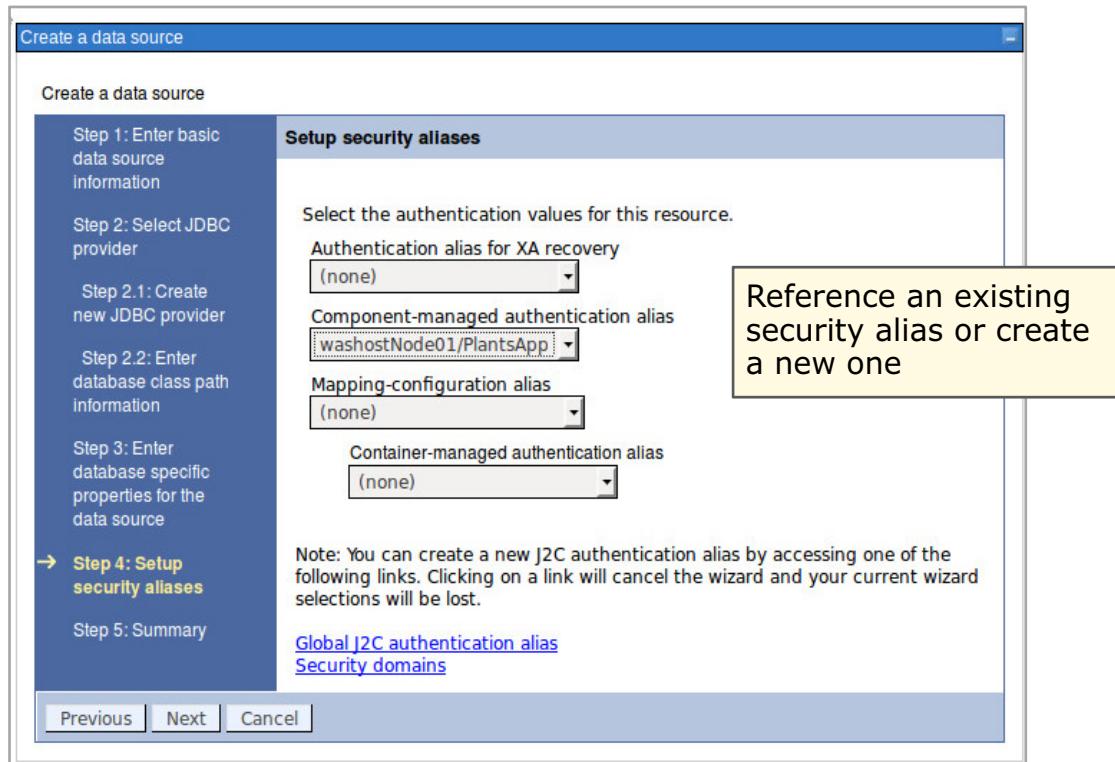
Application installation

© Copyright IBM Corporation 2016

Figure 6-11. Step 3: Enter database-specific properties for the data source

These screens show the process for creating the required resources that an application uses to access a database. This simple screen shows the data that is required to access a database.

Step 4: Set up security aliases



Application installation

© Copyright IBM Corporation 2016

Figure 6-12. Step 4: Set up security aliases



Step 5: Summary of data source creation

Create a data source

Create a data source		
Step 1: Enter basic data source information Step 2: Select JDBC provider Step 2.1: Create new JDBC provider Step 2.2: Enter database class path information Step 3: Enter database specific properties for the data source Step 4: Setup security aliases → Step 5: Summary	Summary	
	Summary of actions:	
	Options	Values
	Scope	cells:washostNode01Cell:nodes:washostNode01
	Data source name	Plants
	JNDI name	jdbc/PlantsByWebSphereDataSource
	JDBC provider name	DB2 Universal JDBC Driver Provider (XA)
	Description	Two-phase commit DB2 JCC provider that supports JDBC 3.0. D sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources created under this provider.
	Class path	\${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar \${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cis_only.jar
	\${DB2UNIVERSAL_JDBC_DRIVER_PATH} \${UNIVERSAL_JDBC_DRIVER_PATH}	/opt/ibm/db2/V10.5/java
Native path	\${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}	
\${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH} Implementation class name	/opt/ibm/db2/V10.5/java com.ibm.db2.jcc.DB2XADataSource	
Driver type	4	
Database name	PLANTS	

Confirm data source specification before creation

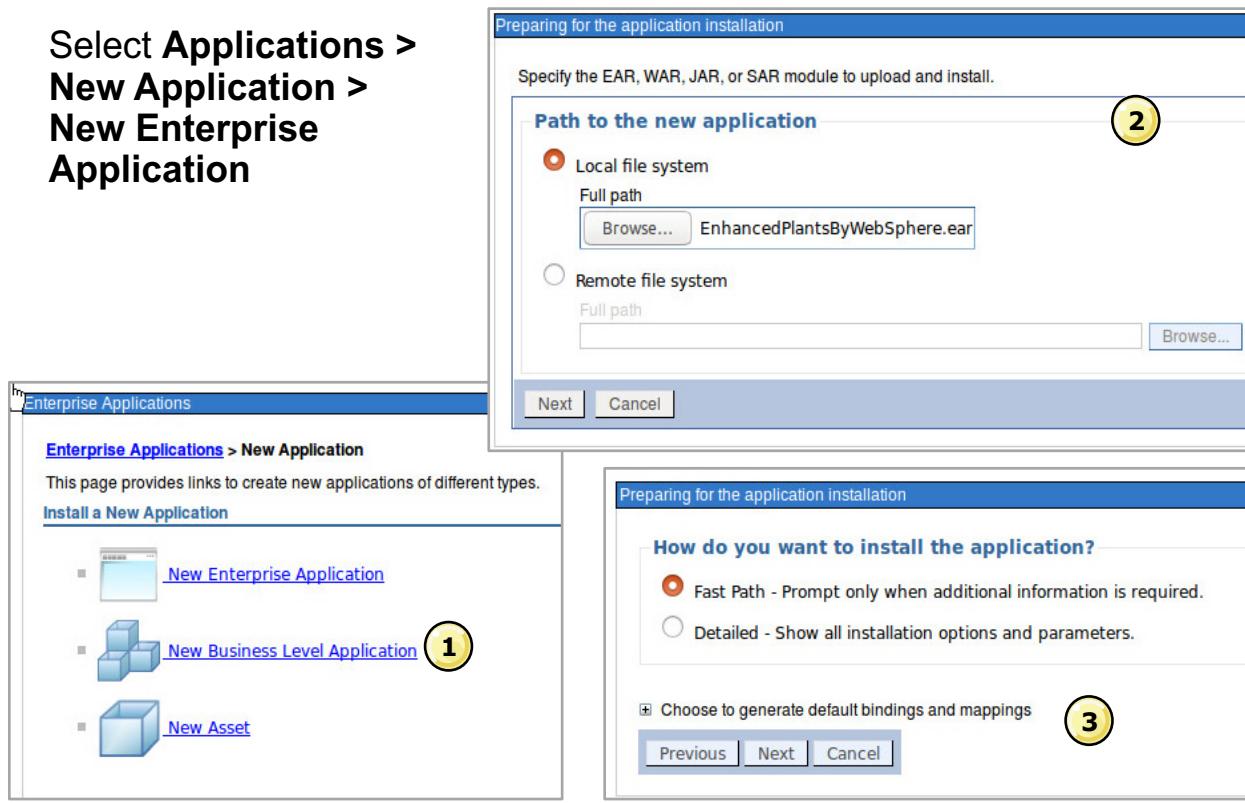
Application installation

© Copyright IBM Corporation 2016

Figure 6-13. Step 5: Summary of data source creation

Installing a new application

Select Applications >
New Application >
New Enterprise
Application



Application installation

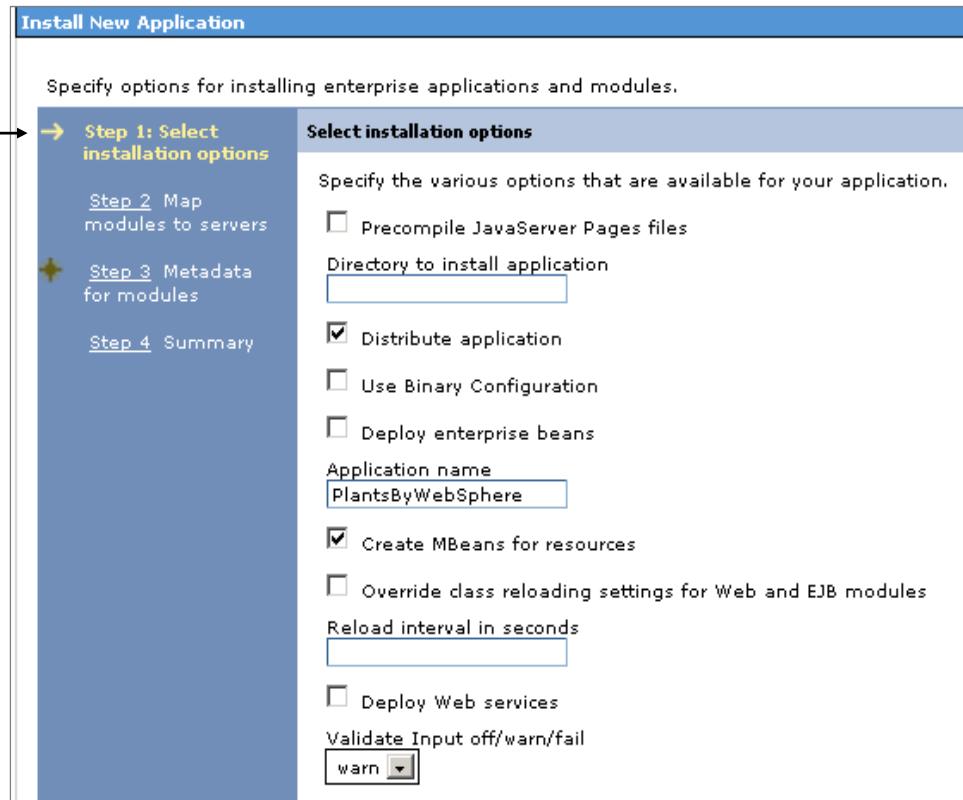
© Copyright IBM Corporation 2016

Figure 6-14. Installing a new application

Here are the screens for the simple method of installing an application. This method is called the “fast path.”

Example of fast path installation

Step 1:
Select
installation
options



Application installation

© Copyright IBM Corporation 2016

Figure 6-15. Example of fast path installation

Various options are available for the fast path installation.



Example of detailed installation

Summary
step →

Summary	
Summary of installation options	
Options	Values
Precompile JavaServer Pages files	No
Directory to install application	
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	PlantsByWebSphere
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\,dll=755#,.*\,so=755#,.*\,a=755#,.*\,sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No

Application installation

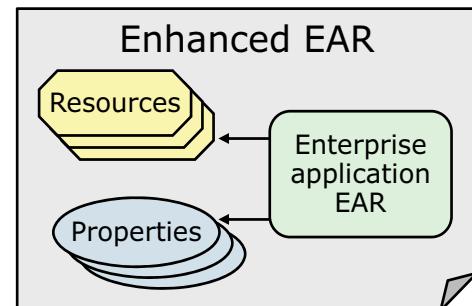
© Copyright IBM Corporation 2016

Figure 6-16. Example of detailed installation

This picture shows the summary of information just before the application is installed.

Enhanced EAR file

- Enterprise archive that contains Java EE artifacts plus resource information necessary to install on WebSphere Application Server
 - JDBC resources (data sources)
 - Class loader
 - JAAS authentication aliases
 - Shared libraries
 - Virtual host information
- Benefits: Improved productivity
 - Application resources and properties come with the application
 - Application installation process creates the necessary resources within the server or cluster
 - Moving application from one server to another also moves the resources
- **Warning:** If unintended application scoped resources are used in production, it can possibly cause problems
- Enhancements can be removed or ignored during application installation



Application installation

© Copyright IBM Corporation 2016

Figure 6-17. Enhanced EAR file

A JDBC resource can be defined in an enhanced EAR file. Enhanced EAR files are specific to IBM WebSphere and are not part of Java EE. However, if an enhanced EAR file is installed on an application server other than WebSphere, artifacts that are defined within this EAR file are ignored.

Developers or administrators can use tools to define resources and properties within an enterprise application, and import or export the enhanced EAR file.

Some resources still must be defined in the application server, for example, JMS and JavaMail. Settings are defined in IBM tools, which are stored in `deployment.xml`, and packaged with the EAR file. These resources are applied at the new application scope.

EAR files and enhanced EAR files are presented in more detail later in the course.



Removing enhancements

Step 1: Select installation options

Step 2 Map modules to servers

Step 3 Summary

Select installation options

Specify the various options that are available for your application.

Precompile JavaServer

Directory to install application
[]

Distribute application

Use Binary Configuration

Deploy enterprise beans

Application name
IVT Application

Create MBeans for resources

Override class reloading settings for Web

Reload interval in seconds
[]

Deploy Web services

Validate Input off/warn/fail
warn

Process embedded configuration

Enterprise Applications

Enterprise Applications > TradeApplication > Application scoped resources

Use this page to view the resources that are defined by the enhanced EAR within this application.

[Preferences]

Name	JNDI name	Resource type	Provider	Description
Trade	jdbc/tradeds	DataSource	Trade	Trade Datasource

- Resources can be ignored
 - Remove enhancements from EAR before you deploy (preferred)
 - Clear **Process embedded configurations**
 - Is prechecked only if it has enhancements
- Resources can be viewed, but not through the normal screens
 - Click **Application scoped resources** under the enterprise application

Application installation

© Copyright IBM Corporation 2016

Figure 6-18. Removing enhancements

Several methods are available to ignore resources in an enhanced EAR file. The preferred method is to remove enhancements from the EAR file before deploying. If the EAR file has enhancements, clear the check box for **Process embedded configuration**. This check box is filled only if enhancements are used.

You can view resources, but not through the normal screens. You must click **Application scoped resources** under the enterprise application.

6.2. Application settings and interaction

Application settings and interaction

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 6-19. Application settings and interaction

Starting an application

- After the application is installed, you can select from a number of options to manage the application
 - Click **Applications > Application types > WebSphere enterprise applications**

The screenshot shows the 'Enterprise Applications' management interface. At the top, there's a toolbar with buttons for Start, Stop, Install, Uninstall, Update, Rollout Update, Remove File, and Export. Below the toolbar is a section for selecting resources, featuring icons for file operations like copy, move, and delete. A table lists applications with columns for Select, Name, and Application Status. The 'PlantsByWebSphere' application is selected (indicated by a checked checkbox) and its status is shown as a red X. Other applications listed are 'DefaultApplication' and 'IBMLUTC', both with green arrow icons.

Select	Name	Application Status
<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	IBMLUTC	
<input checked="" type="checkbox"/>	PlantsByWebSphere	

Application installation

© Copyright IBM Corporation 2016

Figure 6-20. Starting an application

After the application is installed, you can see it listed on the Enterprise Applications page by going to **Applications > Application types > WebSphere enterprise applications**.

To start the application that you installed, click the check box beside it to select it, and then click **Start**. The application status symbol changes from a red X to a green arrow.



Application update

Specify the EAR, WAR, JAR, RAR, or SAR module to upload and update.

Application to be updated:
PlantsByWebSphere

Application update options

Replace the entire application
Upload an enterprise archive (*.ear) to replace the entire installed application.

Specify the path to the replacement ear file.

Local file system
Full path
ps\PlantsByWebSphere.ear

Remote file system
Full path

Replace or add a single module
If the path to the new module matches an existing path to a module in the installed application, the new module replaces the existing module. If the path to the module does not exist in the installed application, the new module is added to the application.

Replace or add a single file
If the path to the new file matches an existing path to a file in the installed application, the new file replaces the existing file. If the path to the file does not exist in the installed application, the new file is added to the application.

You can update the full application, a single module, a single file, or part of the application

Application installation

© Copyright IBM Corporation 2016

Figure 6-21. Application update

It is possible to update an application without uninstalling that application first. This screen is used to update an existing application. It is not important that you read all of the options on this slide, just that you understand the purpose of this activity.

Other application configuration settings

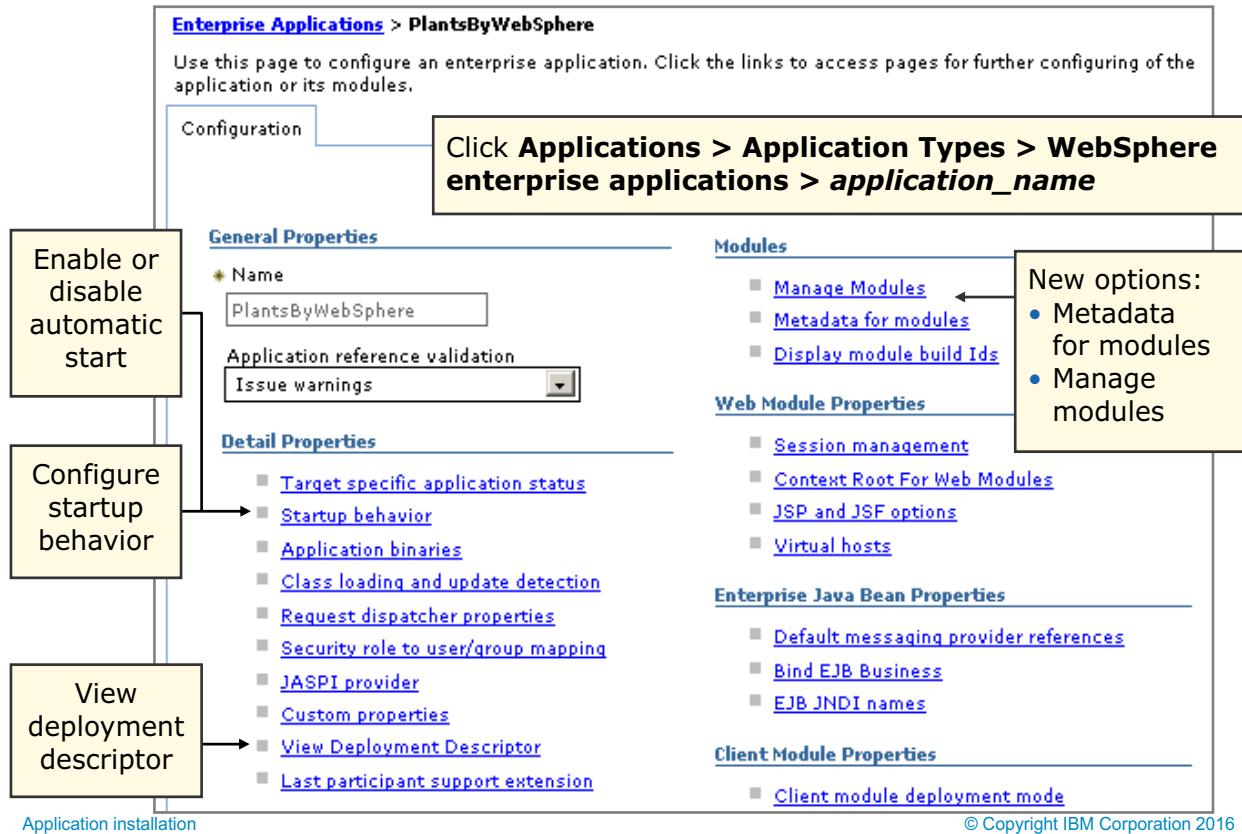


Figure 6-22. Other application configuration settings

This slide shows a broad overview of the metadata for an enterprise application that can be modified or configured through the application console.

Application startup behavior and auto start

- Startup behavior of an application
 - The values set determine how quickly an application starts and what occurs when an application starts
 - Click **Applications > Application Types > WebSphere enterprise applications > *application_name* > Startup behavior** in the console navigation tree to configure startup behavior settings
- Automatic starting of an application
 - By default, an installed application starts automatically when the server starts on which the application is deployed
 - Click **Applications > Application Types > WebSphere enterprise applications > *application_name* > Target specific application status** to configure auto startup

Figure 6-23. Application startup behavior and auto start

For the startup behavior of an application, the values that are set affect how quickly an application starts and what occurs when an application starts. By default, an application starts when its parent server starts.



View the application deployment descriptor

General Properties

- * Name: PlantsByWebSphere
- Application reference validation: Issue warnings

Detail Properties

- [Target specific application status](#)
- [Startup behavior](#)
- [Application binaries](#)
- [Class loading and update detection](#)
- [Request dispatcher properties](#)
- [Security role to user/group mapping](#)
- [JASPI provider](#)
- [Custom properties](#)
- [View Deployment Descriptor](#)
- [Last participant support extension](#)

References

- [Resource references](#)
- [EJB references](#)
- [Shared library references](#)
- [Shared library relationships](#)

Modules

- [Manage Modules](#)
- [Metadata for modules](#)
- [Display module build IDs](#)

Web Module Properties

- [Session management](#)
- [Context Root For Web Modules](#)
- [JSP and JSF options](#)
- [Virtual hosts](#)

Enterprise Java Bean Properties

- [Default messaging provider references](#)
- [Bind EJB Business](#)
- [EJB JNDI names](#)

Client Module Properties

- [Client module deployment mode](#)

Database Profiles

- [SQL profiles and pureQuery bind files](#)

Application installation

© Copyright IBM Corporation 2016

Figure 6-24. View the application deployment descriptor

This slide shows the link for viewing the application deployment descriptor. Note: IBM Assembly and Deploy Tools allows for editing this data before the application is installed.



The application deployment descriptor

The screenshot shows the "Enterprise Applications" interface. Under "Enterprise Applications > PlantsByWebSphere > Deployment Descriptor", it says "Expand and collapse the application deployment descriptor data to view." Below are two buttons: "Expand All" and "Collapse All". The XML code for the deployment descriptor is displayed:

```
<application version="5"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/application_5.xsd" >
  <module>
    <web>
      <web-uri> PlantsByWebSphereWeb.war </web-uri>
      <context-root> PlantsByWebSphere </context-root>
    </web>
  </module>
  <security-role>
    <description> Samples Administrator </description>
    <role-name> SampAdmin </role-name>
  </security-role>
</application>
```

Application installation

© Copyright IBM Corporation 2016

Figure 6-25. The application deployment descriptor

You can view the `application.xml` deployment descriptor for the installed application from the administrative console. Select **Enterprise applications > application_name** and then click **View Deployment Descriptor** under Detail Properties.

Manage modules (1 of 4)

- To view the web or EJB deployment descriptors for an enterprise application
 - Click **Manage Modules**

The screenshot shows the 'Enterprise Applications' configuration interface. At the top, it says 'Enterprise Applications > PlantsByWebSphere'. Below that, a message says 'Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.' A 'Configuration' tab is selected. On the left, there's a sidebar with 'General Properties' (Name: PlantsByWebSphere, Application reference validation: Issue warnings), 'Detail Properties' (Target specific application status, Startup behavior, Application binaries, Class loading and update detection), and 'Web Module Properties' (Session management, Context Root For Web Modules, JSP and JSF options, Virtual hosts). On the right, under 'Modules', there are three links: 'Manage Modules' (which is highlighted with a red box), 'Metadata for modules', and 'Display module build Ids'. At the bottom, there's a link for 'Enterprise Java Bean Properties'.

Application installation

© Copyright IBM Corporation 2016

Figure 6-26. Manage modules (1 of 4)

These four slides show the steps for viewing the deployment descriptor of an EJB JAR file (EJB module).

First, on the application details view, click the link for **Manage Modules**.

Manage modules (2 of 4)

Enterprise Applications

[Enterprise Applications](#) > [PlantsByWebSphere](#) > Manage Modules

Manage Modules

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are running through.

Clusters and servers:

WebSphere:cell=washostNode01Cell,node=washostNode01,server=server1	<input type="button" value="Apply"/>
--	--------------------------------------

Select	Module	URI	Module Type	Server
<input type="checkbox"/>	PlantsByWebSphere	PlantsByWebSphereWeb.war,WEB-INF/web.xml	Web Module	WebSphere:cell=washostNode01Cell,node=washostNode01,server=server1

Application installation

© Copyright IBM Corporation 2016

Figure 6-27. Manage modules (2 of 4)

These four slides show the steps for viewing the deployment descriptor of an EJB JAR file (EJB module).

Second, click the module that contains the EJBs (shown here as “PlantsbyWebSphere”).



Manage modules (3 of 4)

Enterprise Applications

[Enterprise Applications](#) > [PlantsByWebSphere](#) > [Manage Modules](#) > [PlantsByWebSphereWeb.war](#)

Use this page to configure an instance of a deployed web module in the application. This page contains deployment-specific information for a web module and session management settings.

Configuration

General Properties		Additional Properties
<ul style="list-style-type: none"> * URI <input type="text" value="PlantsByWebSphereWeb.war"/> Alternate deployment descriptor <input type="text"/> * Starting weight <input type="text" value="10000"/> * Class loader order <input type="text" value="Classes loaded with parent class loader first"/> 		<ul style="list-style-type: none"> ■ View Module Class Loader ■ Custom properties ■ Target specific application status ■ View EJB Deployment Descriptor ■ View Web Deployment Descriptor ■ Session Management
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>		

Application installation

© Copyright IBM Corporation 2016

Figure 6-28. Manage modules (3 of 4)

These four slides show the steps for viewing the deployment descriptor of an EJB JAR file (EJB module).

Third, on the right side, click the link titled **View EJB Deployment Descriptor**.



Manage modules (4 of 4)

Enterprise Applications

[Enterprise Applications](#) > [PlantsByWebSphere](#) > [Manage Modules](#) > [PlantsByWebSphereWeb.war](#) > Deployment Descriptor

Expand and collapse the application deployment descriptor data to view.

[Expand All](#) [Collapse All](#)

```

<ejb-jar id="ejb-jar_ID" version="3.1" metadata-complete="false"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/ejb-jar_3_1.xsd"
>
  <enterprise-beans>
    <session>
      <ejb-name>CustomerMgr</ejb-name>
      <mapped-name/>
      <ejb-class>com.ibm.websphere.samples.pbw.ejb.CustomerMgr</ejb-class>
      <session-type>Stateless</session-type>
      <init-on-startup>False</init-on-startup>
      <concurrency-management-type>Container</concurrency-management-type>
      <local-bean/>
    </session>
    <persistence-context-ref>
      <persistence-context-ref-name>
        com.ibm.websphere.samples.pbw.ejb.CustomerMgr/em</persistence-context-ref-name>
      <persistence-unit-name>PBW</persistence-unit-name>
      <persistence-context-type>Transaction</persistence-context-type>
      <injection-target>
        <injection-target-class>com.ibm.websphere.samples.pbw.ejb.CustomerMgr</injection-target-class>
        <injection-target-name>em</injection-target-name>
      </injection-target>
    </persistence-context-ref>
  </session>
  <session>

```

Application installation

© Copyright IBM Corporation 2016

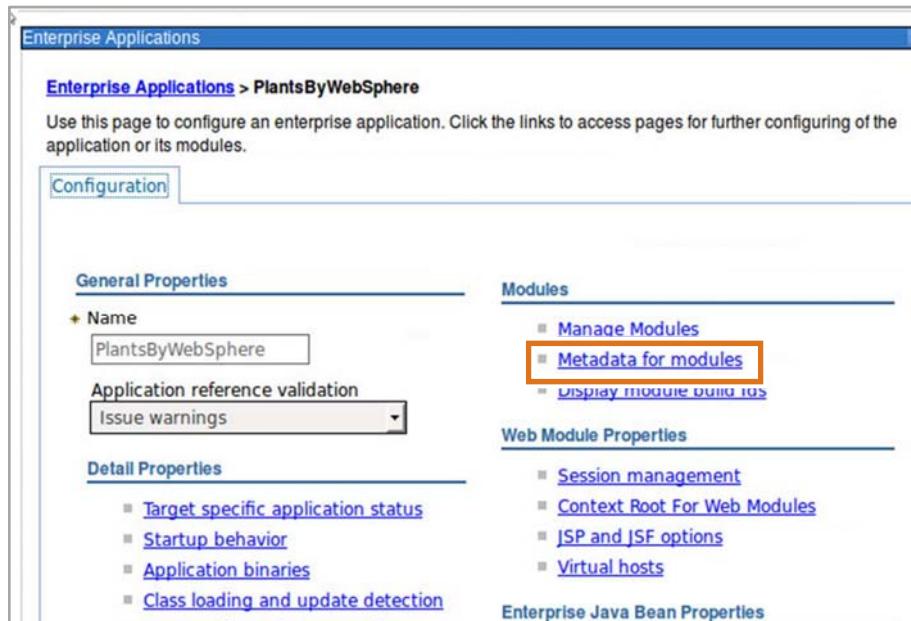
Figure 6-29. Manage modules (4 of 4)

These four slides show the steps for viewing the deployment descriptor of an EJB JAR file (EJB module).

Fourth, the details of the EJB deployment descriptor are shown.

Metadata for modules (1 of 2)

- With this option, you can either allow or ignore metadata that is coming from annotations in source code



Application installation

© Copyright IBM Corporation 2016

Figure 6-30. Metadata for modules (1 of 2)

It is not necessarily a suggested practice to set metadata-complete to *true* if your programmers intend to use annotation-based programming techniques. The technique that you adopt can be based on personal preference.

Values in deployment descriptors can augment or override the equivalent annotation-based metadata:

- After viewing the deployment descriptor information in the administrative console, the deployer can change the appropriate deployment descriptor to reflect the required changes.
- Expand the EAR file in the directory you installed from, for example:
`<profile_root>/<profile>/installableApps`
- Modify the deployment descriptor with your required changes and reinstall or update the application.

Metadata for modules (2 of 2)

The screenshot shows a window titled "Enterprise Applications" with the path "Enterprise Applications > PlantsByWebSphere > Set the metadata-complete attribute of the deployment descriptor for the module". Below this, it says "Metadata for modules". A descriptive text explains the "metadata-complete" attribute: "The metadata-complete attribute defines whether the deployment descriptor for this module is complete. Set the metadata-complete attribute to "true" to merge and persist annotation-based metadata with existing XML-based deployment descriptor metadata to avoid scanning of annotation-based metadata each time the module is read. If the attribute remains "false", then the annotation-based metadata is scanned each time the module is read and can impact performance." At the bottom, there is a table with two rows:

Module	URI	metadata-complete attribute
PlantsByWebSphere	PlantsByWebSphereWeb.war,WEB-INF/ejb-jar.xml	<input type="checkbox"/>
PlantsByWebSphere	PlantsByWebSphereWeb.war,WEB-INF/web.xml	<input type="checkbox"/>

Application installation

© Copyright IBM Corporation 2016

Figure 6-31. Metadata for modules (2 of 2)

The next slide gives some guidelines on how and when to set the metadata-complete options for the modules of an application.

6.3. Monitored directory

Monitored directory

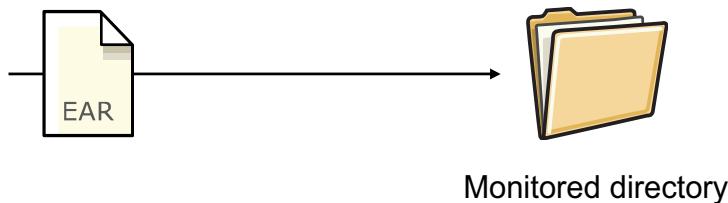
© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 6-32. Monitored directory

Overview of monitored directory deployment

- A simple and fast way to install, update, and uninstall applications without:
 - The administrative console
 - Rational Application Developer
 - wsadmin
 - A specially configured environment
- Tasks can be accomplished by copying archive files in or out of a monitored directory
 - The application must be an EAR, JAR, WAR, or SAR



Application installation

© Copyright IBM Corporation 2016

Figure 6-33. Overview of monitored directory deployment

A monitored directory application deployment is a new, simple, and fast way for administrators and developers to install, update, and uninstall applications by moving archive files in or out of a monitored directory.

A user who prepackages an application file with all bindings specified can quickly deploy that application without any tools other than a running application server, or in a network deployment environment, a deployment manager.

Monitored directory application deployment can be done with applications that are packaged as enterprise archive (EAR) files, web archive (WAR) files, Java archive (JAR) files, or SIP application resources (SARs).

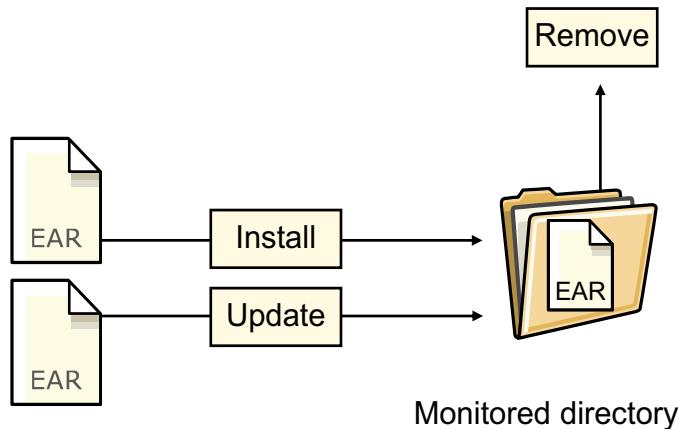
In addition to the mentioned archive files, a properties file can be used to deploy an application.

Properties files are different from the other file types. They are not archives. Instead, they contain properties that describe an application, including the source archive location and its installation parameters. Monitored directory deployment with properties files relies on the properties file based configuration (PFBC) feature that was introduced in WebSphere Application Server Version 7.

Using a properties file can allow a higher degree of control over the deployment than with a plain archive.

Supported tasks

- Install: Place an archive file into the monitored directory
- Uninstall: Remove an archive file from the monitored directory
- Update: Move or copy a new archive file with the same name as an existing archive file in the monitored directory



Application installation

© Copyright IBM Corporation 2016

Figure 6-34. Supported tasks

To install an application, use an operating system file management tool such as a graphical file manager or the command line to copy or move its archive file into a monitored directory.

Deleting a file from a monitored directory causes uninstallation of the corresponding application.

To do a full replacement update of a deployed application, move or copy an updated archive file with the same application name into a monitored directory. The archive file name determines the application name. Or, if it is an EAR archive and a display name is specified in the archive, the display name determines the application name.



Enabling the monitored directory

The screenshot shows the WebSphere administrative console interface. On the left, the navigation tree is expanded under 'Applications', with 'Global deployment settings' selected and highlighted by a red box. The main panel displays the 'Global deployment settings' configuration page. A large orange box highlights the 'Monitored Directory Deployment' section. Inside this section, there is a checkbox labeled 'Monitor directory to automatically deploy applications'. Below the checkbox are fields for 'Monitored directory' containing the value `\${USER_INSTALL_ROOT}/monitoredDeployableApps` and 'Polling interval' set to '5 seconds'. A callout box with a yellow background contains the following text:

- Disabled by default in both stand-alone federated environments
 - Click **Applications > Global deployment settings**

Application installation

© Copyright IBM Corporation 2016

Figure 6-35. Enabling the monitored directory

Before using monitored directory deployment, it must be enabled. Go to the administrative console under **Applications > Global deployment settings**. In addition to enabling the service, the Global deployment settings panel allows configuration of the root path to the monitored directories and the polling interval at which the monitored directories are checked for changes.

Enabling the service is required only once.

The deployment manager or base server where the service is running must be restarted to register any changes on the **Applications > Global deployment settings** panel.

Notes about the monitored directory

- Within the existing directory structure, it is possible to create specific server, cluster, and even node directories
 - These additional structures allow tasks to be directed to specific elements of your environment
- Directory location:
 - Stand-alone:
`<profile_root>/monitoredDeployableApps/servers/<servername>`
 - Federated:
`<dmgr_profile>/monitoredDeployableApps/servers/<servername>`
- For clusters:
 - Create a clusters directory:
`.../monitoredDeployableApps/clusters/<clusternode>`
- For servers with the same name on federated nodes:
 - Applications are deployed to all servers with the same name
 - Create:
`.../monitoredDeployableApps/nodes/nodename/servers/<servername>`

[Application installation](#)

© Copyright IBM Corporation 2016

Figure 6-36. Notes about the monitored directory

The path to a monitored directory depends on the type of application server, and what the target of the deployment is going to be. To control where the monitored directory applications are deployed, administrators create more directories to represent their clusters or servers. These names must exactly match what is in the cell.

If you are using a stand-alone application server, then the only possible target is the server itself, and the monitored directory is automatically created if the service is enabled. For example, if the profile is called AppSrv01, and the server is named server1, the path is:

`app_server_root/profiles/AppSrv01/monitoredDeployableApps/server1`

If you are using a network deployment system, it is necessary to create the monitored directories manually.

For application servers on a node that is federated with a deployment manager, you must create the monitored directories for servers under the deployment manager profile:

`app_server_root/profiles/dmgrfilename/monitoredDeployableApps/servers/server_name`

If multiple servers on different nodes have the same name and you want to target only one of the servers, you can specify the node and server in the path to the monitored directory. Create a directory for the node by using the node name, then servers, and finally the server_name

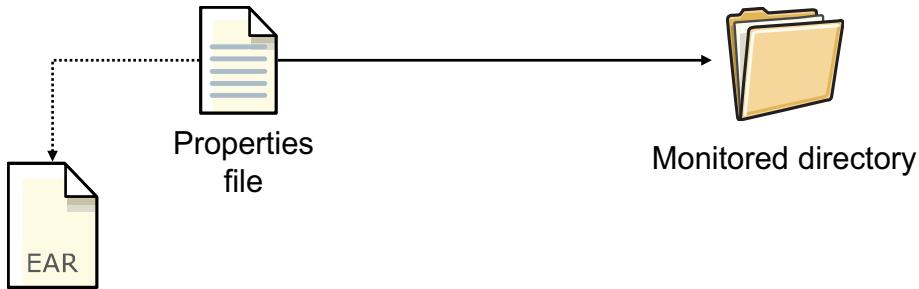
directory: app_server_root/profiles/dmgrfilename/monitoredDeployableApps/nodes
/node_name/servers/server_name

For clusters, create a monitored directory under the deployment manager profile with the name of the targeted cluster:

app_server_root/profiles/dmrfilename/monitoredDeployableApps/clusters
/cluster_name

Drag-and-drop properties files

- The standard drag-and-drop approach lacks the ability to do anything but the default
 - No ability is available to customize a deployment in any way
- Drag-and-drop technique also supports properties file based configuration
 - A property file can define which EAR file to install, and also configure any of the necessary attributes



Application installation

© Copyright IBM Corporation 2016

Figure 6-37. Drag-and-drop properties files

Monitored directory supports the ability to drag a properties file. This support allows administrators to customize any installation settings that might be required for monitored directory installations.

Properties file based configuration

- Provides a group of administrative commands
- Manage system configuration
- Troubleshoot configuration issues
- Replicate configuration properties across profiles, nodes, servers, or applications
- Use properties file on monitored directory deployment to deploy applications
- Introduced in WebSphere Application Server V7.0

Application installation

© Copyright IBM Corporation 2016

Figure 6-38. Properties file based configuration

Version 7 introduced the properties file based configuration feature, which provides a group of administrative commands to manage system configuration by using properties files. Users can use the properties file based configuration commands to copy configuration properties from one environment to another. They can also use the properties file to troubleshoot configuration issues and apply one set of configuration properties across multiple profiles, nodes, cells, servers, or applications. The details of the properties file based configuration commands are documented in the version 7 documentation of IBM Knowledge Center. The monitored directory deployment is extended to use the properties file to install, uninstall, or update an application in version 8.

Steps to use a properties file to deploy applications (1 of 2)

- Step 1: Create a properties file that defines deployment options
 - Use properties file based configuration command to create an application properties file
 - Extract application properties to a file

```
AdminTask.extractConfigProperties('[-propertiesFileName  
myApp.props -configData Deployment=MyApplication]')
```

- Note: The monitored directory installation process uses only the properties that relate to an application
- Create the file manually

[Application installation](#)

© Copyright IBM Corporation 2016

Figure 6-39. Steps to use a properties file to deploy applications (1 of 2)

A user can use the properties file based configuration command to extract application properties to an application properties file. After a properties file is extracted, edit application properties and copy the properties file to the `MonitoredDeployableApps` directory under the `deploymentProperties` directory. The Monitored directory deployment service runs the `applyConfigProperties` command to apply the application properties change to install, uninstall, or update an application for you.

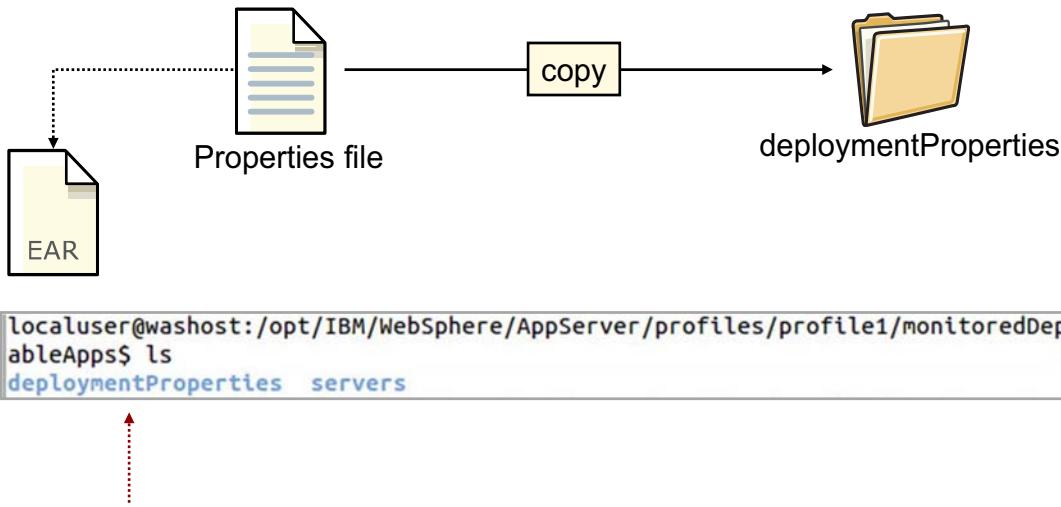
Step 1 shows how to use the properties file based configuration `extractConfigProperties` command to extract application properties to a file.

By default, the `extractConfigProperties` command produces output that displays all columns, including all hidden and non-hidden columns of installation tasks and task data values, in separate rows. You can also extract application properties in simple output format to display non-hidden columns of installation task data in `columnName=value` pairs.

You can also use the properties file examples that are documented in the IBM Knowledge Center to create an application properties file to deploy, uninstall, or update an application.

Steps to use a properties file to deploy applications (2 of 2)

- Step 2: Verify that the targeted server or cluster member is running
- Step 3: Verify that monitored directory deployment is enabled
- Step 4: Copy the properties file to the deploymentProperties directory



Application installation

© Copyright IBM Corporation 2016

Figure 6-40. Steps to use a properties file to deploy applications (2 of 2)

The monitored directory deployment service starts the application after it is deployed or updated. Therefore, you must verify that the application server or cluster member on which you want to install the enterprise application files is running.

Step 3 is to verify that the monitored directory deployment is enabled. You can look at the IBM Knowledge Center “Setting monitored directory deployment values” section.

The last step is to copy the application properties file to the `monitoredDeployableApps` directory. You can use a file browser to drag the properties file to the monitored directory or use the operating system command to copy the properties file to the directory.

Unit summary

- Describe methods of installing enterprise applications in WebSphere Application Server
- Explain how fine-grained application updates work
- Describe enterprise application properties
- Enable monitored directories
- Deploy an application by using the monitored directory
- Use a properties file-based configuration with monitored directories to deploy an application

Application installation

© Copyright IBM Corporation 2016

Figure 6-41. Unit summary

Review questions

1. True or False: You can update a single module or part of an application in the console.
2. True or False: The default startup behavior for an application is to automatically start when the server starts.
3. True or False: Monitored directory deployment is enabled in the Applications > Global Deployment settings pane.



Application installation

© Copyright IBM Corporation 2016

Figure 6-42. Review questions

Write your answers here:

- 1.
- 2.
- 3.

Review answers

1. True or False: You can update a single module or part of an application.
The answer is True.

2. True or False: The default startup behavior for an application is to automatically start when the server starts.
The answer is True.

3. True or False: Monitored directory deployment is enabled in the Applications > Global Deployment settings pane.
The answer is True.



Application installation

© Copyright IBM Corporation 2016

Figure 6-43. Review answers

Exercise: Installing an application

© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 6-44. Exercise: Installing an application

Exercise objectives

After completing this exercise, you should be able to:

- Use the administrative console to install an application
- Use a web browser to test the application
- Use the drag function to deploy an application

Unit 7. Problem determination

Estimated time

00:45

Overview

This unit introduces you to resources and basic methods for problem determination.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server V9 documentation in IBM Knowledge Center, troubleshooting and support topics:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Describe a basic approach for problem determination
- List resources for completing a problem investigation
- Locate relevant log files
- Examine log activity
- Enable tracing on specific components
- Enable High Performance Extensible Logging (HPEL) and use the Log Viewer
- Describe features of the IBM Support Assistant

Problem determination

© Copyright IBM Corporation 2016

Figure 7-1. Unit objectives

Topics

- Server logs and diagnostic tracing
- Gathering diagnostic data
- Problem determination tools
- IBM Support Assistant

Problem determination

© Copyright IBM Corporation 2016

Figure 7-2. Topics

7.1. Server logs and diagnostic tracing

Server logs and diagnostic tracing

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 7-3. Server logs and diagnostic tracing

Examining server log files

- Log files are an initial source of diagnostic data
- WebSphere provides several useful logs, including:
 - JVM logs
 - Process logs
 - HTTP plug-in logs
 - Console runtime messages

Problem determination

© Copyright IBM Corporation 2016

Figure 7-4. Examining server log files

WebSphere provides several useful logs and messaging facilities, including JVM logs, HTTP plug-in logs, and console runtime messages.

WebSphere Application Server logs

- **JVM logs:** Created by redirecting the `System.out` and `System.err` streams of the JVM to independent log files
 - One set of JVM logs for each application server and all of its applications that are located by default in the following directory:
`<profile_root>/<profile_name>/logs/<server_name>`
 - `SystemOut.log` and `SystemErr.log`
- **Process logs:** Contain two output streams (`stdout` and `stderr`) which are accessible to native code that runs in the process
 - One set for each application server
 - `native_stderr.log` and `native_stdout.log`
- **IBM service log:** Contains both the WebSphere Application Server messages that are written to the `System.out` stream and some special messages that contain extended service information
 - One per profile (node)
 - `activity.log`

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-5. WebSphere Application Server logs

The following list describes the various WebSphere Application Server logs:

- **Java virtual machine (JVM) logs**

The JVM logs are created by redirecting the `System.out` and `System.err` streams of the JVM to independent log files. WebSphere Application Server writes formatted messages to the `System.out` stream. In addition, applications and other code can write to these streams by using the `print()` and `println()` methods of the streams.

In a WebSphere Application Server Network Deployment configuration, JVM logs are also created for the deployment manager and each node agent because they also represent JVMs.

- **Process logs**

WebSphere Application Server processes contain two output streams, which are accessible to native code that is running in the process. These streams are the `stdout` and `stderr` streams. Native code, including Java virtual machines (JVM), might write data to these process streams. JVM-provided `System.out` and `System.err` streams can be configured to write their data to these streams.

As with JVM logs, each application server has a set of process logs, since each JVM is an operating system process. For a WebSphere Application Server Network Deployment configuration, it has a set of process logs for the deployment manager and each node agent.

- **IBM service logs**

The IBM service log contains the WebSphere Application Server messages that are written to the `System.out` stream. It also contains some special messages that provide extended service information, which is normally not of interest, but can be important when analyzing problems. One service log exists for all WebSphere Application Server JVMs on a node, including all application servers. The IBM service log is maintained in a binary format and requires a special tool to view. This viewer, the Log Analyzer, provides more diagnostic capabilities. In addition, the binary format provides capabilities that IBM support organizations use.

The HTTP server plug-in log is covered later in this presentation.

Server log files: Types and locations

- Application server log file destinations and names are configurable
 - The default location is:
`<was_root>/profiles/<profile_name>/logs/<server_name>`
- Application server log files are all text files
 - JVM logs: `SystemOut.log` and `SystemErr.log`
 - Start and stop logs: `startServer.log` and `stopServer.log`
 - Diagnostic trace log: `trace.log`
 - Process logs: `native_stdout.log` and `native_stderr.log`
- Web server plug-in log file:
 - `http_plugin.log`
 - Location: `<plugin_root>/logs/<webserver_name>`
- IBM service log is in binary format
 - In `<profile_root>/<profile_name>/logs/activity.log` directory
 - Use Log Analyzer in IBM Support Assistant to view

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-6. Server log files: Types and locations

All WebSphere Application Server log files are under the
`<was_root>\profiles\<profile_name>\logs` directory.

- The default names for the JVM logs are `SystemOut.log` and `SystemErr.log`. They contain server information and user program information (sent by `System.out.xxx` code in the program).
- The `startServer.log` and `stopServer.log` files can also be found under the `<was_root>\logs\<servername>` directory; they contain information that the server logs as it starts and shuts down.
- The `activity.log` file size can be set by using the administrative console as IBM service logs. You can also disable `activity.log`.
- The standard JVM output and error logs are `SystemOut.log` and `SystemErr.log`.
- The `startServer.log` and `stopServer.log` files are written to during startup and shutdown of the application servers.
- If tracing is enabled, the `trace.log` file contains output from a diagnostic trace. The location and name of this log file are configurable.

- If verbose GC is enabled, the operating system uses the `native_stdout.log` and `native_stderr.log` files to log out-of-memory exceptions and verbose garbage collection data.
- The `http_plugin.log` file is the web server plug-in log file. It is not in `<was_root>`, but in `<plugin_root>\logs\<webserver_name>`.



Configuring JVM logs

Configuration Runtime

General Properties

System.out

* File Name:
\${SERVER_LOG_ROOT}/SystemOut.log

File Formatting

Basic (Compatible) ▾

Log File Rotation

File Size
Maximum Size
3 MB

Time
Start Time
24
Repeat Time
24 hours

Maximum Number of Historical Log Files. Number in range 1 through 200.
2

- From the administrative console, select **Troubleshooting > Logs and Trace > server_name > JVM Logs**
- SystemOut and SystemErr logs can be configured from this page
- Logs are self-managing
 - Can roll over based on time or file size
 - Number of historical log files is configurable
- To view logs through the console, use the runtime tab

Problem determination

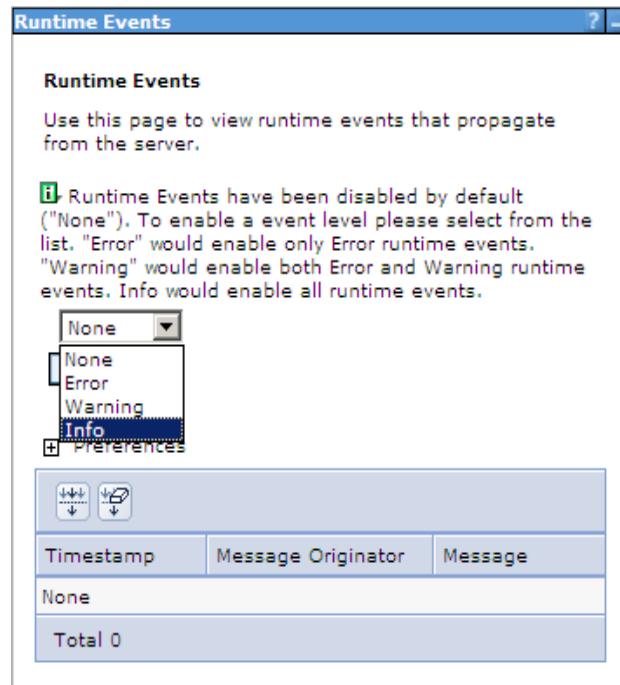
© Copyright IBM Corporation 2016

Figure 7-7. Configuring JVM logs

The JVM logs can be configured from the administrative console. For both log files, systemOut and systemErr, you can specify the path to their location, file formatting (basic or advanced), and log file rotation (by file size or time interval). You can also configure the maximum number of historical files to store on the file system.

Viewing runtime messages in the console (1 of 2)

- Runtime events are grouped according to severity: error, warning, information
- To view, select:
Troubleshooting > Runtime Messages >
 - Runtime Error
 - Runtime Warning
 - Runtime Information
- Runtime events are disabled by default
 - None
- Select **Info** to enable all runtime events



Problem determination

© Copyright IBM Corporation 2016

Figure 7-8. Viewing runtime messages in the console (1 of 2)

When viewing runtime messages, first select the **Error**, **Warning**, or **Info** category link (a count of zero means that nothing is available). Then, the details for the selected category are shown. Selecting one of these links gives you detail information (see the next slide).

You can have multiple pages of messages, and you can click the button at the bottom of the page to view and read all of them.

Viewing runtime messages in the console (2 of 2)

- Runtime events details include:
 - Message code and text
 - Brief explanation of the event
 - Action for the user to take
 - What server component issued the message
- Other details include:
 - Timestamp
 - Thread ID
 - Node name
 - Server name

Runtime Events

Runtime Events > Message Details

Use this page to view runtime events that propagate from the server.

General Properties

Message
SRVE0255E: A WebGroup/Virtual Host to handle /PlantsByWebSphere has not been defined.

Message type
Runtime error

Explanation
Could not find a web group (web module) or virtual host to handle the request. This is an application error.

User action
Be sure the web group and virtual host is defined and deployed.

Message Originator
com.ibm.ws.webcontainer

Source object type
RasLoggingService

Problem determination

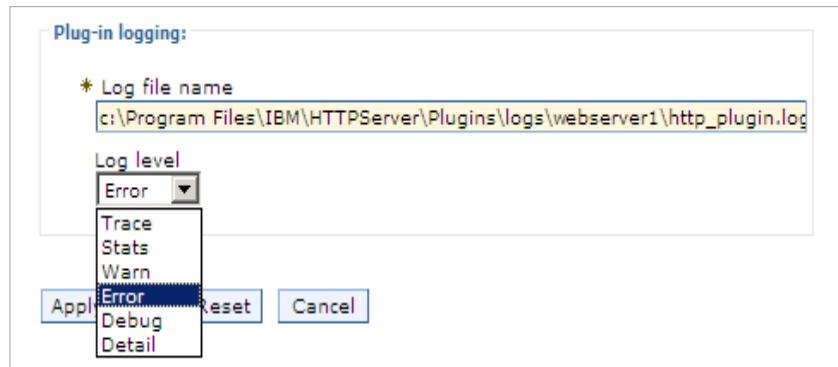
© Copyright IBM Corporation 2016

Figure 7-9. Viewing runtime messages in the console (2 of 2)

Most runtime messages are designed with improved message text. A message code, for example SRVE0255E, can be used to look up the message in the IBM Knowledge Center. Information is shown on the detail screen for the event, and sometimes a user action is provided to resolve the problem.

HTTP plug-in logs and tracing

- To configure plug-in logs and tracing from the administrative console, click **Servers > Web Servers > web_server_name > Plug-in Properties > Configuration tab > Plug-in logging**
- Default location:
 $<\text{plugin_root}>/\text{logs}/<\text{web_server_name}>/\text{http_plugin.log}$
- To trace all the steps in the HTTP request process, set the Log level to **Trace** (caution: this trace produces much log data)



Problem determination

© Copyright IBM Corporation 2016

Figure 7-10. HTTP plug-in logs and tracing

The HTTP plug-in logs messages to a log file that is stored under the `plugin_root` directory on the web server. The log level can be set to different values, depending on the amount of detail that you want written to the log. Setting the log level to **Trace** causes it to log all steps in the request process.

- **The embedded HTTP server logs**
 - Administrative console panels are available for configuring embedded HTTP server (HTTP transport channel) logs. An access log and an error log are also available.
 - From main application server configuration panel, click **Troubleshooting > server_name > HTTP Error and NCSA Access Logging**.
 - This service is disabled by default. To enable, check the **Enable logging service** box, and restart the server.
 - Error logs can be controlled separately with log levels: Critical, Error, Warning, Information, Debug.

Diagnostic tracing

- Diagnostic tracing can be used to collect detailed processing data for all WebSphere Application Server components
- To take advantage of tracing, you must:
 - Enable tracing of one or more WebSphere components
 - Configure and view trace logs
 - Interpret trace logs and trace messages
- Trace files show the time and sequence of methods that WebSphere Application Server base classes call
 - You can use these files to pinpoint the failure

Problem determination

© Copyright IBM Corporation 2016

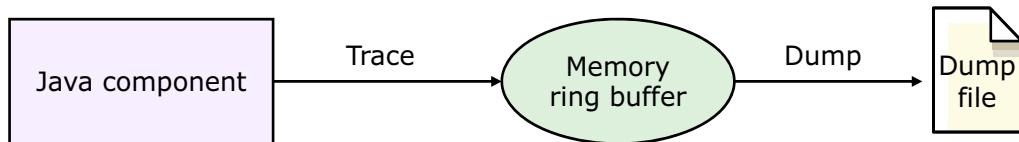
Figure 7-11. Diagnostic tracing

To take advantage of tracing, you do these steps:

1. Enable tracing of one or more WebSphere components.
2. Configure and view trace logs.
3. Interpret trace logs and trace messages.

Using diagnostic tracing

- Tracing can be started:
 - While the server is running, by using Runtime Diagnostic Trace
 - When the server is starting and running, by using Configuration Diagnostic Trace
- Trace output can be directed to:
 - File (default)
 - Memory ring buffer, and dumped after trace stops
- Tracing has a significant effect on performance
 - Enable temporarily for problem determination
 - Tracing to a file is slower than tracing to a memory ring buffer



Problem determination

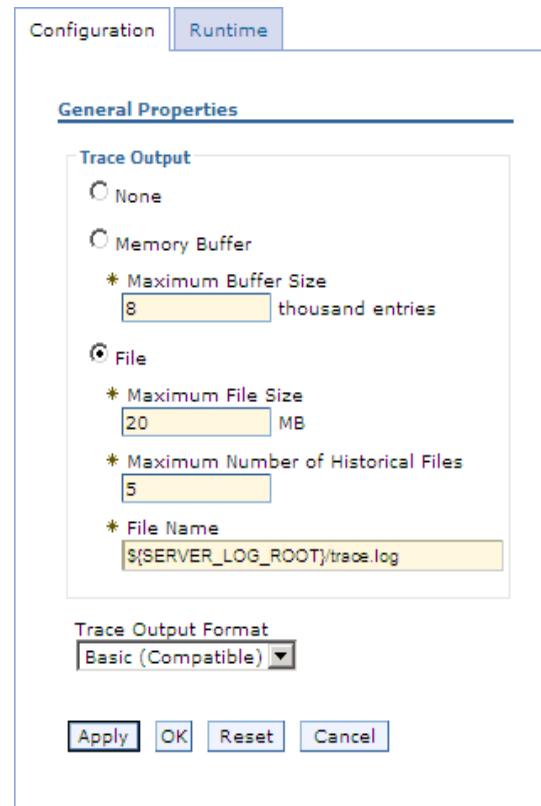
© Copyright IBM Corporation 2016

Figure 7-12. Using diagnostic tracing

Trace output allows administrators to examine processes in the application server and diagnose various issues. On an application server, trace output can be directed either to a file or to an in-memory circular buffer. If trace output is directed to the in-memory circular buffer, it must be dumped to a file before it can be viewed. On an application client or stand-alone process, trace output can be directed either to a file or to the process console window. In all cases, trace output is generated as plain text in basic, advanced, or log analyzer format as the user chooses. The basic and advanced formats for trace output are similar to the basic and advanced formats that are available for the JVM message logs.

Enable and configure tracing

- Troubleshooting > Logs and Trace > *server_name* > Diagnostic Trace
- Configure Trace Output
 - None
 - Memory buffer
 - File (default)
- Configure Trace Output Format
 - Basic (IBM Support preference)
 - Advanced
- **Note:** Configure Log Detail Level to get trace output



Problem determination

© Copyright IBM Corporation 2016

Figure 7-13. Enable and configure tracing

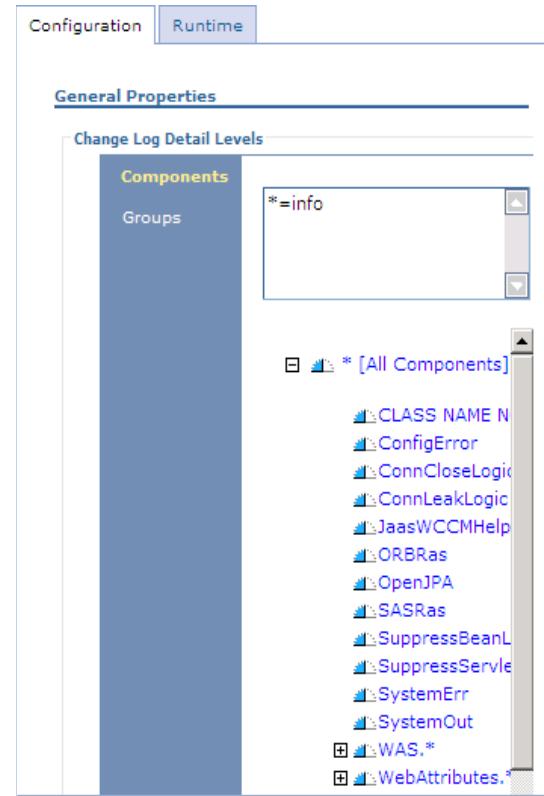
The Diagnostic Trace Service box looks almost the same as it did in previous versions. The Configuration and Runtime tabs behave as they always have, with Configuration affecting the configuration repository and taking effect at the next startup. The Runtime settings take effect immediately but are only optionally persisted to the server configuration.

Trace strings must be configured on a separate panel (Log Detail Level).

View and dump are available in the Runtime tab of diagnostic trace. The Log Analyzer can be used to analyze trace output, but you can use your favorite editor. Before you can view or dump a trace, you must specify the log detail level.

Setting the log detail level (1 of 2)

- Logs and trace > `server_name` > Change Log Detail Level
- Log detail level affects tracing **and** regular logging
 - Setting levels below **info** reduces the amount of data in logs
 - ***=off** disables logging altogether
- Trace levels (**fine**, **finer**, **finest**) are not displayed in the trace file unless logging is enabled
- Use the graphical menu to type in or set the log string
 - Default is ***=info**
- User-created applications can be instrumented too, and be included in the trace output



Problem determination

© Copyright IBM Corporation 2016

Figure 7-14. Setting the log detail level (1 of 2)

Log levels control which events Java logging processes.

WebSphere Application Server controls the levels of all loggers in the system. The level value is set from configuration data when the logger is created and can be changed at run time from the administrative console.

Note: Trace information, which covers events at levels fine, finer, and finest, can be written to the trace log. Therefore, if you do not enable diagnostic trace, setting the log detail level to fine, finer, or finest does not affect the data that is logged.

- Log string syntax: `<component / group> = <log level>`

Examples include:

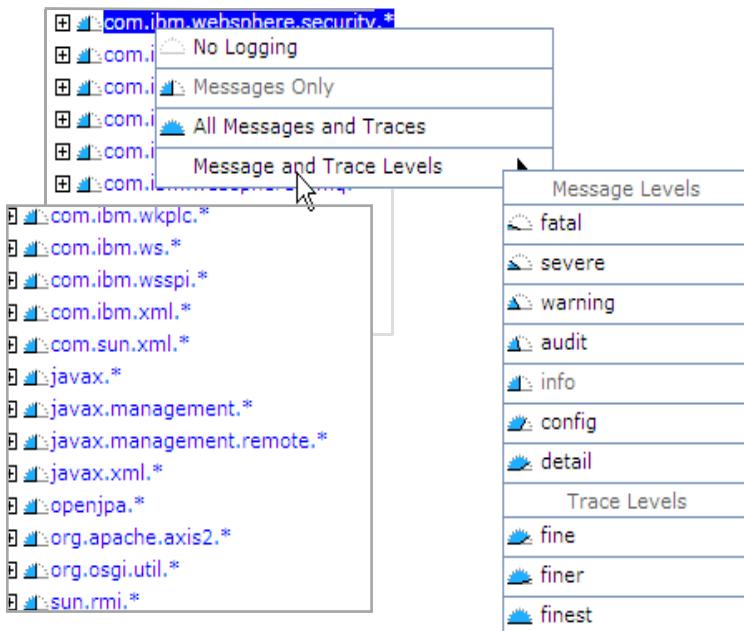
- `com.ibm.ws.classloader.ClassGraph=fine` enables the finest trace level for `com.ibm.ws.classloader.ClassGraph`
- `EJBContainer=fine` enables the least verbose trace level for all components in the EJBContainer group
- `com.ibm.ws.classloader.*=finer` enables detailed trace for all classes in the `com.ibm.ws.classloader` package
- `*=info` sets the log level for all components to info (default is no trace output)

Setting the log detail level (2 of 2)

- Select component

- Select one of
 - No Logging
 - Messages Only
 - All Messages and Traces

- Or select **Message and Trace Levels** and select **Message Levels**



Problem determination

© Copyright IBM Corporation 2016

Figure 7-15. Setting the log detail level (2 of 2)

Examples include:

- **No Logging** results in the following trace string: com.ibm.websphere.security.*=off
- **Messages Only** results in the following trace string: *=info (the default)
- **All Messages and Traces** results in the following trace string:
com.ibm.websphere.security.*=all

Trace output content and format

- Trace output allows administrators to examine processes in the application server and diagnose various issues
- Two formats can be configured:
 - Basic (IBM Support preference)
 - Advanced
- Trace events that are displayed in basic use the following format:
 - <timestamp><threadId><shortName><eventType>[className] [methodName]<textmessage> [parameter 1] [parameter 2]
- Possible values of **eventType** include:
 - >: A trace entry of type method entry
 - <: A trace entry of type method exit
 - 3: A trace entry of type finest, debug, or dump

Problem determination

© Copyright IBM Corporation 2016

Figure 7-16. Trace output content and format

Other values of **eventType** include:

- 1: A trace entry of type fine or event
- 2: A trace entry of type finer
- Z: A placeholder to indicate that the trace type was not recognized

Reading a log or trace file (1 of 2)

- Example log record format
- [5/9/16 12:27:56:237 EDT] 00000000 PMIImpl A CWPMI1001I:
PMI is enabled
 - **Timestamp** = [5/9/16 12:27:56:237 EDT]
 - **Thread ID** = 00000000
 - **Logger** = PMIImpl
 - **Message type** = A
 - **Message code** = CWPMI1001I
 - **Message** = PMI is enabled

Message type	Description
1, 2, 3	Trace information: fine, finer, finest
A	Audit
W	Warning
Z	Type was not recognized
E	Error
D	Detail
C	Configuration
F	Fatal (exits process)
I	Information
O	Program output (system.out)
R	Program output (system.err)

Problem determination

© Copyright IBM Corporation 2016

Figure 7-17. Reading a log or trace file (1 of 2)

The following list defines the sections of a trace entry:

- **TimeStamp**: The time stamp is formatted by using the locale of the process where it is formatted. It includes a fully qualified date (YYMMDD), 24-hour time with millisecond precision, and the time zone. **ThreadId** is an eight-character hexadecimal value that is generated from the hash code of the thread that issued the trace event.
- **ThreadName**: The name of the Java thread that issued the message or trace event.
- **ShortName**: The abbreviated name of the logging component that issued the trace event. This value is typically the class name for WebSphere Application Server internal components, but can be some other identifier for user applications.
- **LongName**: The full name of the logging component that issued the trace event. This value is typically the fully qualified class name for WebSphere Application Server internal components, but can be some other identifier for user applications.
- **EventType**: A one-character field that indicates the type of the trace event. Trace types are in lowercase.
- **ClassName**: The class that issued the message or trace event.
- **MethodName**: The method that issued the message or trace event.

Reading a log or trace file (2 of 2)

- Timestamps give good clues:
 - Timestamps are real host time values
 - Good when you are comparing traces from different processes and correlating events of different servers
- Look for exceptions (search for exception from top of stack trace)
 - Events before the exception are probable causes
 - Events after the exception are recovery attempts
- Often useful to follow a single thread
 - Use the Thread ID to gather related messages

Problem determination

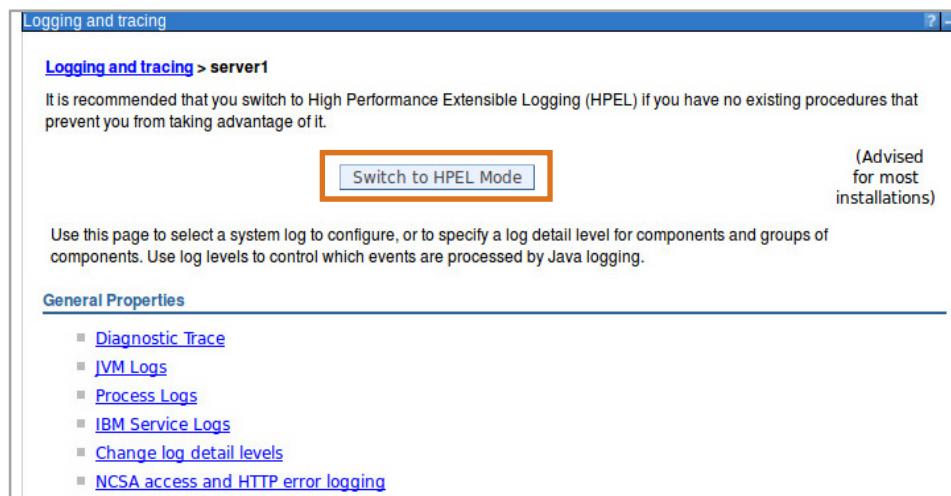
© Copyright IBM Corporation 2016

Figure 7-18. Reading a log or trace file (2 of 2)

While it is possible to use a text editor to read logs and trace files, it is suggested that you use a tool such as Log Analyzer, which is available in the IBM Support Assistant.

High Performance Extensible Logging (HPEL)

- HPEL can be enabled on any server in the cell
 - Deployment manager
 - Node agent
 - Application server
- Click **Troubleshooting > Logging and tracing > server_name**
- Click **Switch to HPEL Mode**



Problem determination

© Copyright IBM Corporation 2016

Figure 7-19. High Performance Extensible Logging (HPEL)

After the log level is switched to HPEL mode for a server, a new list of links is in the **General Properties** section. One new link is **Change log and trace mode**, which you can use to switch back to basic logging.

HPEL logging and tracing configuration

- Use the “Logging and tracing” page to configure HPEL logging and tracing

The screenshot shows the 'Logging and tracing' configuration page for 'server1'. It includes sections for 'General Properties', 'Configure HPEL logging', 'Configure HPEL trace', and 'Configure HPEL text log'. The 'Configure HPEL text log' section has its 'Current status:' set to 'Enabled'.

General Properties	
<u>Configure HPEL logging</u>	
Directory	/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of logs	50 Megabytes
<u>Configure HPEL trace</u>	
Directory	/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of trace	50 Megabytes
<u>Configure HPEL text log</u>	
Current status:	Enabled
Directory	/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of text log	50 Megabytes

Problem determination

© Copyright IBM Corporation 2016

Figure 7-20. HPEL logging and tracing configuration

This screen capture shows the default configuration for HPEL logging and tracing. To modify the configuration, click any of the links on this page.

After HPEL is enabled for a server, it is suggested that you disable the HPEL text log. Disabling the text log improves server performance.

Configure HPEL logging

- Clicking the **Configure HPEL logging** link starts the configuration page
- Changes that are made on the **Configuration** tab require you to restart the server
- Changes that are made on the **runtime** tab take effect immediately

General Properties

* Directory path
\${SERVER_LOG_ROOT}

Enable log record buffering

Start new log file daily at: Time 12 AM

Log record purging policies

Begin cleanup of oldest records
when log size approaches maximum

Log record age limit
48 Hours old

Maximum log size
20 Megabytes

* Out of space action
Stop logging

Problem determination

© Copyright IBM Corporation 2016

Figure 7-21. Configure HPEL logging

Clicking the **Configure HPEL logging** link starts the configuration page. Changes that are made on the **Configuration** tab require a server restart. Changes that are made on the runtime tab take effect immediately.

The HPEL Log Viewer

- Click **Troubleshooting > Logging and tracing > server_name > Log Viewer**
 - Numerous filtering options are available to modify which records are displayed

The screenshot shows the 'Logging and tracing' interface with the 'Log Viewer' selected. The main content area displays log records from 'server1' on August 16, 2016, at 11:49:50. The records are listed in a table with columns: TimeStamp, Thread ID, Logger, Level, and Message. The 'Message' column contains several underlined message IDs (e.g., TRAS0017, FFDC1007) which are likely links to detailed explanations.

TimeStamp	Thread ID	Logger	Level	Message
8/16/16, 11:49:50.548	00000001	ManagerAdmin	INFO	TRAS0017 : The startup trace state is *=info.
8/16/16, 11:49:50.554	00000001	ManagerAdmin	INFO	TRAS0111 : The message IDs that are in use are deprecated
8/16/16, 11:49:50.563	00000001	ProviderTracker	INFO	com.ibm.ffdc.osgi.ProviderTracker AddingService FFDC1007 : FFDC Provider Installed: com.ibm.ffdc.util.provider.FfdcOr
8/16/16, 11:49:50.578	00000001	nfig.ModelMgr	INFO	WSVR08001 : Initializing core configuration models
8/16/16, 11:49:50.790	00000001	MetaDataMgr	INFO	WSVR0179 : The runtime provisioning feature is disabled. All components will be started.
8/16/16, 11:49:50.851	00000001	ProviderTracker	INFO	com.ibm.ffdc.osgi.ProviderTracker AddingService FFDC1007 : FFDC Provider Installed: com.ibm.ws.ffdc.impl.FfdcProvid
8/16/16, 11:49:50.905	00000001	dminInitialzier	AUDIT	ADMN0015 : The administration service is initialized.
8/16/16, 11:49:51.120	00000001	figServiceImpl	INFO	PLGC0057 : The plug-in configuration service started successfully.

Problem determination

© Copyright IBM Corporation 2016

Figure 7-22. The HPEL Log Viewer

The log view section lists the records. Use the **First Page**, **Previous Page**, **Next Page**, and **Last Page** buttons to move through the list of records. You can also specify filter criteria in the Content and Filtering Details section to limit the rows that are shown. Records are always listed in the order that the server recorded them. By default, the log view has five columns:

- **Time Stamp:** The time when the event was recorded.
- **Thread ID:** The identity of the thread that recorded the event in hexadecimal notation.
- **Logger:** The logger that recorded the event.
- **Level:** The type of event that was recorded.
- **Message:** The message from the recorded event. If the message has a message ID, the message ID is underlined. Click the message ID to get an explanation and suggested user action for the message.

To manipulate the log view, you can use available buttons to complete the following actions:

- **Refresh View:** Uses records from the server to clear the contents of the viewer and reinitialize the view. Use this button to retrieve information about any additional rows that are created since the log viewer was started.

- **Show Only Selected Threads:** Show only records with the same thread as the one that is selected in the selection area. Clicking this button enables the **Show All Threads** button.
- **Show All Threads:** Lists any records that were filtered when you clicked **Show Only Selected Threads**. This button is enabled only when you use the **Show Only Selected Threads** button to restrict the view.
- **Select Columns:** Click to select the columns in the viewer that you want to view.
- **Export:** Exports logs to the local workstation in any of basic, advanced, or binary (HPEL) formats.
- **Copy to Clipboard:** Copies the records that are highlighted in the selection area into the operating system clipboard.
- **Server Instance Information:** Lists attributes for the selected server instance process. Use this table to find attributes and corresponding values for the server instance process environment. These properties are similar to the ones that are found in the header of basic mode logs.

logViewer command-line tool (1 of 2)

- Use the `logViewer` command to query the contents of the High Performance Extensible Logging (HPEL) log and trace repositories since they are in binary format
 - `logViewer.sh/bat`

- You can also use the `logViewer` command to view new log and trace repository entries as the server writes content to them

```
logViewer.sh/bat -monitor [interval]
```

- Using parameters, you can specify a time period to view

- Use the `startDate` and `stopDate` parameters

- You can specify a minimum, maximum or a single one of these levels: FINEST, FINER, FINE, DETAIL, CONFIG, INFO, AUDIT, WARNING, SEVERE, FATAL

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-23. *logViewer command-line tool (1 of 2)*

The first tool for analyzing HPEL logs is the command-line log viewer. The command-line log viewer is a simple, intuitive, and fast tool for doing analysis on the logs in problem determination efforts.

You are no longer required to be aware of whether data is written to the `SystemOut.log`, `SystemErr.log`, or `trace.log` file. With HPEL, the logs are consolidated, and the log viewer can be used to view all the data or filtered according to what subset of messages you need:

- `monitor [interval]`: Specifies that you want the `logViewer` to continuously monitor the repository and output new log record entries as they are created. You can provide an optional integer argument after this parameter to specify how often you want the `logViewer` tool to query the repository for new records. By default, the `logViewer` queries the repository for new records every 5 seconds. When used with other filtering options, only those new records that match the filter criteria are displayed.

logViewer command-line tool (2 of 2)

- To get a listing of available `logViewer` options:
 - `logViewer.sh/bat -help`
- The `logViewer` command started from the profile `bin` directory without any options
 - Results in a display of all the log and trace messages of a server
- To view the log and trace entries of the most recent server, run:
 - `logViewer.sh/bat -latestInstance`
 - With this option, it is no longer necessary to clear log files before restarting a server

Problem determination

© Copyright IBM Corporation 2016

Figure 7-24. *logViewer command-line tool (2 of 2)*

To get a listing of available `logViewer` options, use the `-help` option.

The `logViewer` command can be started from the `bin` directory of a profile without any options, and a listing of all the entries of a server is shown. If a profile has multiple servers, the user is first prompted to select the server for viewing.

What is Cross Component Trace (XCT)?

- A feature that annotates the logs so that entries that are related to a request are identified as belonging to the same unit of work
- The request might traverse more than one
 - Thread
 - Process
 - Server
- XCT helps identify the root cause of problems across components, which provides the following benefits:
 - Enables administrators and support teams to follow the flow of a request from end-to-end
 - The request is traced as it traverses thread or process boundaries, or travels between stack products and WebSphere Application Server
 - Helps to resolve questions about which component is responsible for a request that fails

Problem determination

© Copyright IBM Corporation 2016

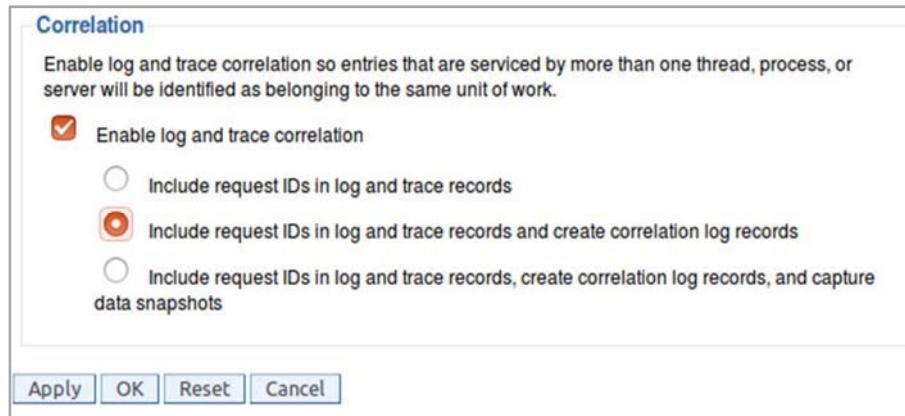
Figure 7-25. What is Cross Component Trace (XCT)?

Depending on the nature of your applications, multiple threads within an application server might be used to handle requests, such as HTTP requests or JMS requests. More than one application server can handle some requests, such as when one application server makes a request to another application server for a web services request.

Applications that are built by using distributed architectures, such as service-oriented architecture, can benefit from XCT, since XCT helps facilitate problem determination across multiple services on different systems.

Administering XCT

- HPEL must be enabled on the server before XCT can be enabled
- Click **Troubleshooting > Logs and trace > *server_name* > Change log detail levels**
- Check **Enable log and trace correlation**



- Select option for including request IDs, creating correlation logs, capturing data snapshots

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-26. Administering XCT

Enable XCT to include request IDs in log and trace files when you want to see which log and trace entries, in all threads and application server processes, are related to the same request. Request IDs are recorded only when using HPEL log and trace mode and can be seen or used for filtering when using the `logViewer` command.

Enable XCT to create correlation log records when you want to log how requests branch between threads and processes, and see extra information about each request. Enabling XCT to create correlation log records might have a significant performance impact on your system, so it is best suited to test and development environments.

XCT request IDs

- XCT request IDs are identifiers added to log and trace records that the server produces
- XCT adds the same request ID to every log or trace record that is a part of the same request, regardless of which thread or JVM produces the record
- When XCT is used with the HPEL log and trace infrastructure, you can view request IDs when logs are output in advanced format
 - `logViewer.sh -minLevel WARNING -format advanced`

```
[Time_stamp] 00000094 W UOW= source=com.ibm.ws.webcontainer.srt  
class=com.ibm.ws.webcontainer.srt.SRTServletResponse method=setIntHeader  
org= prod= component= thread=[WebContainer : 4]  
requestID=[AAAsirk1Njr-AAAAAAA+] appName=[PlantsByWebSphere]
```

Figure 7-27. XCT request IDs

You can use XCT to augment your log and trace files with correlation information. This correlation information clarifies which threads and which application server processes participated in the handling of each request.

Use XCT request ID information to track requests

- Filter your logs by request ID by using the HPEL logViewer command-line tool
 - `logViewer.sh -includeExtensions requestId=AAAsirk1Njr-AAAAAAA+AAA`

```
[Time_stamp] 00000094 XCT           I   BEGIN AAAsirk1Njr-AAAAAAA+AAA
000000000000-cccccccccc2 HTTPCF(InboundRequest
/PlantsByWebSphere/javax.faces.resource/jsf.js.jsf
RemoteAddress(127.0.0.1) RequestContext(-957274864))

[Time_stamp] 00000094 srt           W
com.ibm.ws.webcontainer.srt.SRTServletResponse setIntHeader SRVE8094W:
WARNING: Cannot set header. Response already committed.

[Time_stamp] 00000094 XCT           I   END   AAAsirk1Njr-AAAAAAA+AAA
000000000000-cccccccccc2 HTTPCF(Request AsyncWrite RequestContext(-
957274864))
```

Problem determination

© Copyright IBM Corporation 2016

Figure 7-28. Use XCT request ID information to track requests

The following information is important to XCT log records:

- XCT log records are typically added to the logs to:
 - Demarcate the beginning and ending of work for a particular request on a particular thread
 - Demarcate when work is transferred to another thread or process, or to indicate when work is returned from another thread or process
 - Demarcate when work moves from major component to major component, even if work continues on the same thread; for example, to show transfer of control from application server code to application code
- XCT log records are composed of:
 - XCT type (BEGIN / END)
 - XCT parent correlator ID (for example, 0000000000-cccccccccc2)
 - XCT current correlator ID (for example, AAAsirk1Njr-AAAAAAA+AAA)
 - XCT annotations, for example:

```
HTTPCF( InboundRequest ) /PlantsByWebSphere/javax.faces.resource/jsf.js.jsf
RemoteAddress(127.0.0.1) RequestContext(-957274864)
```

- XCT tools

The HPEL logViewer tool is able to filter log and trace records by request ID.

Tools such as the XCT Log Viewer can also take advantage of XCT log records or XCT request IDs, or both, when rendering log and trace content. The XCT Log Viewer is available as a tool add-on for the IBM Support Assistant.

7.2. Gathering diagnostic data

Gathering diagnostic data

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 7-29. Gathering diagnostic data

Gathering JVM diagnostic data

- A wealth of valuable JVM diagnostic data can be gathered by doing the following tasks:
 - Enable logging of verbose garbage collection data for servers
 - Generate a Java thread dump
 - Generate a Java heap dump
 - Generate a system core dump
 - Configure hung thread detection
 - Enable tracing of connection leaks

Problem determination

© Copyright IBM Corporation 2016

Figure 7-30. Gathering JVM diagnostic data

The tasks that are listed on this slide are described in detail in the WebSphere Application Server V9 documentation in IBM Knowledge Center. See the articles on troubleshooting.

Enable verbose garbage collection

- Verbose GC is an option that the JVM run time provides
- Provides a garbage collection log
 - Interval between collections
 - Duration of collection
 - Compaction required
 - Memory size, memory that was freed, memory available
- Turns on verbose GC for each server through the administrative console
 - **Servers > server_name > Process Definition > Java Virtual Machine**
 - Select “Verbose Garbage Collection” check box
 - Save and distribute
 - Restart the server or servers
- Usually writes to `native_stderr.log` file
 - Varies depending on platform and WebSphere version
 - Some load on system resources because of disk I/O, but minimal unless thrashing occurs

[Problem determination](#)

© Copyright IBM Corporation 2016

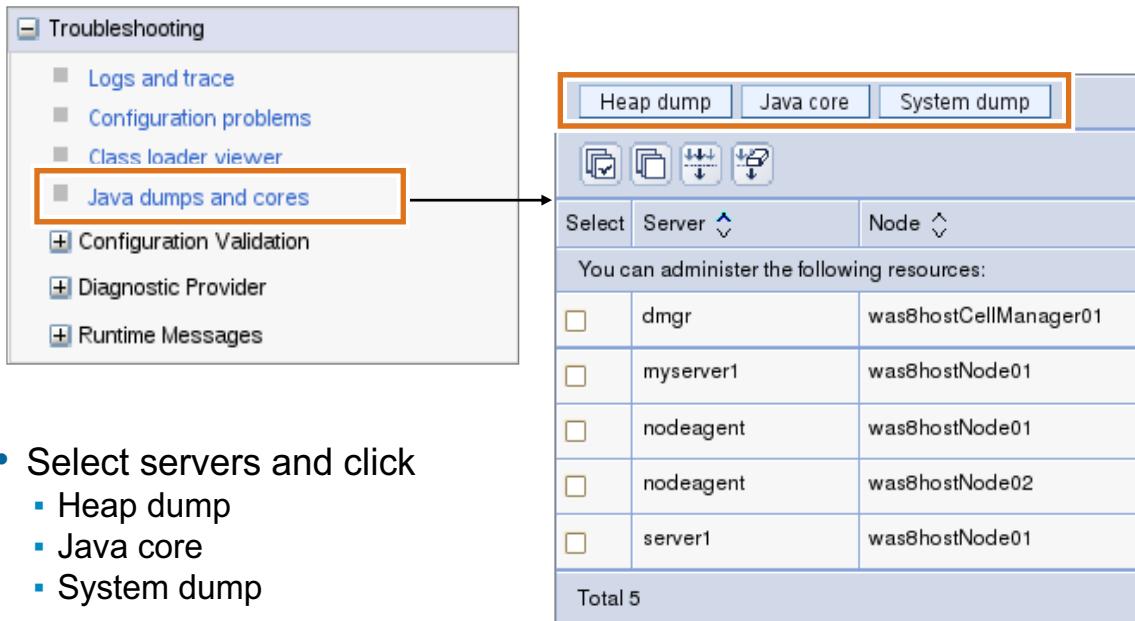
Figure 7-31. Enable verbose garbage collection

It is often suggested that you have verbose GC enabled permanently in production. The cost on a reasonably well-tuned JVM is small. The benefits of having it on the first time that something happens are considerable (no necessity to reproduce the problem a second time after enabling). It is also good to keep an eye on the verbose GC regularly as a way to monitor the health of the system, even when nothing bad is noticed.

Enabling verbose GC is a decision that each system administrator must make conscientiously. However, it is no longer “not recommended as a normal production setting.”

Java memory dumps and cores

- New feature in the Troubleshooting section is **Java dumps and cores**



Problem determination

© Copyright IBM Corporation 2016

Figure 7-32. Java memory dumps and cores

Clicking the Java dumps and cores link opens the panel that is shown in the screen capture. Use this panel to generate heap dumps, Java cores, or system dumps for a running process. Select the server and click the appropriate button for **Heap dump**, **Java core** (thread dump), or **System dump** (JVM core). The files that result from these operations are placed on the local file system; by default they are written to the profile root directory, `was_root/profiles/profile-name`.

- **Heap dump:** A heap dump is a snapshot of JVM memory. It shows live objects in the memory and references between them.
- **Java core:** Use this button to investigate why a server is hanging or to investigate messages in the logs that indicate a thread did not complete its work in the expected amount of time.
- **System dump:** Use this button to generate system native dumps of the server process. These dumps can be large.

A note on verbose garbage collection data: As in previous versions, verbose GC is not enabled by default. When you enable verbose GC for a server in version 8, the default garbage collection policy is generational-concurrent (gencon). The data is written to the `native_stderr.log` or `native_stdout.log` file, depending on the operating system of the server.

Generating a JVM thread dump (javacore)

- What is a javacore?
 - Small diagnostic text file that the JVM produces
 - Contains much vital information about the running JVM process
 - Provides a snapshot of all the running threads, their stack traces, and the monitors (locks) held by the threads
 - Useful for detecting hang or deadlock conditions

- How to manually trigger a thread dump
 - Warning: Triggering a thread dump can terminate the server process
 - Use operating system facilities:
`kill -3 <JVM PID>` (UNIX or Linux)
 - Explicitly tell WebSphere to generate a thread dump
 - Issue wsadmin Jacl or Jython commands

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-33. Generating a JVM thread dump (javacore)

Run the following wsadmin Jacl commands:

```
set jvm [$AdminControl completeObjectName=JVM, process=<SERVER_NAME>, *]
$AdminControl invoke $jvm dumpThreads
```

Thread dumps also can be helpful in detecting performance problems. Take a few (at least three) snapshots of the JVM (about 2 – 3 minutes apart). Analyze the javacore file to see what different threads are doing in each snapshot.

For example, a series of snapshots where container threads are in the same method or waiting on the same monitor or resource is an indication of a bottleneck, hang, or deadlock.

Generating a JVM heap dump

- JVM heap dump is created in the `<profile_root>` directory when an `OutOfMemoryError` exception is thrown
 - Heap dump is in PHD (Portable Heap Dump) format by default
 - Different locations and formats can be configured by using command-line arguments
- Use the `-Xdump` generic JVM arguments to configure heap dumps
- For example:
 - `Xdump:heap:file=/dumps/heapdump.%Y%m%d.%H%M%S.%pid.%seq.phd` specifies the location and format of the heap dump file name
 - `Xdump:heap, opts=PHD+CLASSIC` enables heap dump creation and creates the file in both binary and text format
 - `Xdump:heap:none` disables heap dump creation
- Can also trigger heap dumps by using wsadmin Jython commands
 - `AdminControl.invoke(AdminControl.completeObjectName
("type=JVM, process=<server_name>, *"),
"generateHeapDump")`

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-34. Generating a JVM heap dump

IBM Support Assistant includes a tool for inspecting Java memory dumps, called the Memory Analyzer. This tool can analyze several Java heap dump formats, including the standard dump formats from both the IBM and Sun Java runtime environments, and also z/OS SVC dumps. It lists and analyzes the data structures in the heap and their relationships, helping you identify the structures that are most likely responsible for memory leakage.

Hung thread detection

- Hung threads can be hard to diagnose
 - They are often not noticed until many threads are hung, resulting in a performance problem
- Application threads can hang for a number of reasons:
 - Endless loops
 - Deadlocks
 - Inaccessible resources
- The **ThreadMonitor** component monitors the web container, ORB, and Async bean thread pools for hung threads
- Example thread monitor message

```
[8/1/11 10:51:01:955 EDT] 00000021 ThreadMonitor W
WSVR0605W: Thread "WebContainer : 0" (00000020) has been
active for 65103 milliseconds and may be hung. There is/are 1
thread(s) in total in the server that may be hung.
```

Figure 7-35. Hung thread detection

Application threads can hang for a number of reasons, including infinite loops or deadlocks.

A component that is known as the ThreadMonitor monitors the web container, ORB, and Async bean thread pools for hung threads. The thread monitor does not try to deal with the hung threads; it just issues notifications so that the administrator or developer can deal with the issues.

When a hung thread is detected, three notifications are sent: a JMX notification is sent for JMX listeners, PMI Thread Pool data is updated for tools like the Tivoli Performance Viewer, and a message is written to the SystemOut log.

When the thread pool gives work to a thread, it notifies the thread monitor. The thread monitor notes the thread ID and time stamp. The thread monitor compares active threads to time stamps.

Threads that are active longer than the time limit are marked “potentially hung.” Performance impact is minimal (less than 1%).

Connection leak diagnostic messages

- Poorly written applications often do not properly release database connections
 - Forget to call `connection.close()`
 - Most often in the exception case
 - Connections must be closed in a `finally{}` block
- Orphaned connections return to the pool only after timeout
 - Can cause a backup of new connections that are waiting for old connections to time out
 - New connections that wait too long throw a `connectionWaitTimeoutException`
- Connection manager is instrumented to print stack traces when a `connectionWaitTimeoutException` occurs
 - Enable by using trace string `WAS.j2c=fine`

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-36. Connection leak diagnostic messages

Applications can suffer from performance problems and even “hang” if they do not close their connections properly. Not properly using the `connection.close()` method often causes a leak. To ensure that connections are closed properly, they must be closed in a `finally{}` block.

WebSphere eventually times out orphaned connections and returns them to the pool, but for an application that makes frequent use of database connections, this timeout might not be enough. New connections can get queued up waiting for the database while old connections are waiting to be timed out. This behavior can halt the application, and you can see `connectionWaitExceptions`.

The connection manager has lower performance impact than connection manager tracing. When activated, it enables a connection manager wrapper that holds the stack trace of all `getConnection()` calls in a throwable object. When an exception is thrown because of waiting on a full connection pool, it is best to print stack traces of **all** open connections. In this way, you significantly narrow your search area when you look at the application source code in an attempt to find the responsible code.

Connection leaks are difficult to diagnose because the error messages do not usually provide enough specific information about the source of the problem. Usually a source code review is needed to find points in the code where connections are not properly closed. The connection manager makes this task much easier.

Checking version levels and applying APARs

- The `versionInfo` command generates a report from data that is extracted from XML files in the `properties/version` folder
 - The report includes a list of changed components and installed or uninstalled maintenance packages
- APAR: Authorized Program Analysis Report
 - Tracks software defects that customers report
- Download suggested fixes, and fixes by version from the WebSphere Support page
- IBM Installation Manager: New standard tool for installing fixes

Problem determination

© Copyright IBM Corporation 2016

Figure 7-37. Checking version levels and applying APARs

The `versionInfo` command generates a report from data that is extracted from XML files in the `properties/version` folder. The report includes a list of changed components and installed or uninstalled maintenance packages.

APAR is an authorized program analysis report; it tracks software defects that customers report.

You can download suggested fix packs for your version of the product from the WebSphere Support website.

IBM Installation Manager is a new standard tool for installing fixes.

7.3. Problem determination tools

Problem determination tools

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 7-38. Problem determination tools

Problem determination tool availability

- Some tools are integrated into the administrative console
 - Tivoli Performance Viewer
 - Configuration validation
 - Class loader viewer
 - Others
- Some tools are separate programs that are included with WebSphere Application Server itself, or sold as separate products in their own right
 - dumpNameSpace
 - versionInfo
 - Rational Application Developer
- IBM Support Assistant:
 - Serves as a central point from which many tools can be found and even run directly inside IBM Support Assistant

Problem determination

© Copyright IBM Corporation 2016

Figure 7-39. Problem determination tool availability

In the overall process of problem determination, diagnostic data must be collected or generated, and the data must be analyzed. Various tools are available to help you collect and analyze diagnostic data for solving problems that are related to JVM, server and application configuration, performance, and namespace issues.

Problem determination tools: Administrative console

- Tivoli Performance Viewer and Performance Advisor
 - Captures and analyzes statistical performance data from a running WebSphere Application Server system
- Configuration validation
 - Checks for errors in a WebSphere Application Server configuration
- Class Loader Viewer
 - Provides a hierarchical view of class loaders that applications use
 - Helps diagnose problems that are related to loading of classes in applications
- Diagnostic Provider
 - Tool for viewing configuration and the current state of individual application server components

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-40. Problem determination tools: Administrative console

The administrative console provides several tools that can be used for troubleshooting and problem determination. See the WebSphere Application Server V9 documentation in IBM Knowledge Center for more details about how to use these tools.

Dumping the JNDI namespace

- The **dumpNameSpace** utility shows JNDI directory content
- Useful to ensure correct association of named objects:
 - JDBC resources
 - EJB components
 - JMS resources
 - Other resources
- Syntax and some of the options:

```
<was_root>\bin\dumpNameSpace
  [-host bootstrap_host_name (defaults to localhost)]
  [-port bootstrap_port_number (defaults to 2809)]
  [-startAt subcontext/in/the/tree]
```

- Output can be redirected to a file and inspected

Figure 7-41. Dumping the JNDI namespace

Usage: `dumpNameSpace [-keyword value]`

If a keyword occurs more than one time with different values, the value that is associated with the last occurrence is used.

The keywords and associated values are:

- `-host myhost.austin.ibm.com`: Bootstrap host, which is the WebSphere host whose namespace you want to dump. It defaults to “localhost.”
- `-port nnn`: Bootstrap port, defaults to 2809.
- `-factory com.ibm.websphere.naming.WsnInitialContextFactory`: The initial context factory that is going to be used to get the JNDI initial context. It defaults as shown, and normally is not required to be changed.
- `-root [cell | server | node | host | legacy | tree | default]`

7.4. IBM Support Assistant



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 7-42. IBM Support Assistant

Introduction to IBM Support Assistant

The IBM Support Assistant 5.0 Team is a free tool to help diagnose problems with IBM software

- The IBM Support Assistant has two types of components:
 - Data collectors: Specific to IBM products such as WebSphere Application Server
 - Team Server: Root cause analysis for the data that is returned from the data collectors
- Benefits of IBM Support Assistant include:
 - Saves time in searching product, support, and educational resources
 - Helps with opening a problem management report (PMR)
 - Allows for easy location and installation of useful product support tools by using a support tool framework
 - Easily downloaded from
<http://www.ibm.com/software/support/isa/>

Figure 7-43. Introduction to IBM Support Assistant

The IBM Support Assistant is a free tool for problem determination for certain IBM products, including WebSphere Application Server.

Benefits of IBM Support Assistant include:

- Improves your ability to locate IBM support, development, and educational information through a federated search interface (one search: multiple resources).
- Provides quick access to the IBM Education Assistant and key product education roadmaps.
- Simplifies access to IBM product home pages, product support pages, and product forums or newsgroups, through convenient links.
- Saves time in submitting problems to IBM Support by collecting key information, then electronically creating a problem management record (PMR) from within IBM Support Assistant. If a PMR must be opened, IBM Support Assistant helps with:
 - Gathering support information that is based on problem type
 - Creating and updating the problem report
 - Tracking your electronic problem report
- Includes a support tool framework, allowing for the easy installation of support tools that are associated with different IBM products.

- Provides a framework for IBM software products to deliver customized self-help information into the different tools within it. Customize your IBM Support Assistant client by using the built-in update capability to find and install new product features or support tools.

IBM Support Assistant: Overview videos available

- IBM Support Assistant is not part of WebSphere Application Server so is not covered in detail here
- Problem determination tools
 - Report tools are non-interactive; they analyze logs and trace files to create humanly readable reports
 - Web tools are interactive tools that are accessed through a browser
 - Web tools run on a team server
 - Desktop tools are interactive and also accessed through a browser
 - Desktop tools run locally by using the local Java runtime environment
- Information on how to download, install, and use these tools is found at <http://www.ibm.com/software/support/isa/>
- The following slides show some of the problem determination features that are relevant to WebSphere Application Server

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-44. IBM Support Assistant: Overview videos available

IBM Support Assistant can change independent of WebSphere Application Server. IBM Support Assistant is mentioned because of its capabilities in problem diagnosis in WebSphere Application Server. However, IBM Support Assistant not covered in detail in this section as it might change.

For high-level information, view the educational videos available on You Tube.

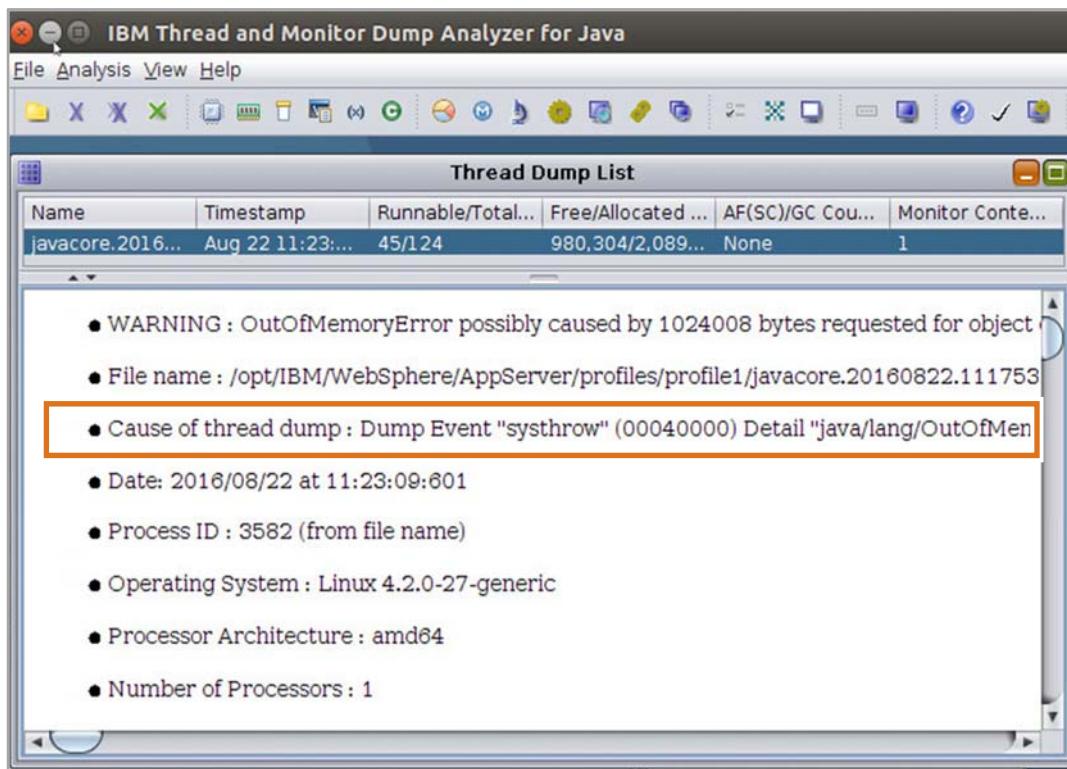
<https://www.youtube.com/playlist?list=PLmooy53GkKpmoIUDX6k67X3WyPV43SyUC>

Most relevant to this unit, watch the IBM Support Assistant 5.0 Team Server: Problem Determination Tools Overview.

https://www.youtube.com/watch?v=7QK_sfJAG1o&index=6&list=PLmooy53GkKpmoIUDX6k67X3WyPV43SyUC



IBM Support Assistant example: Javacore analysis



Problem determination

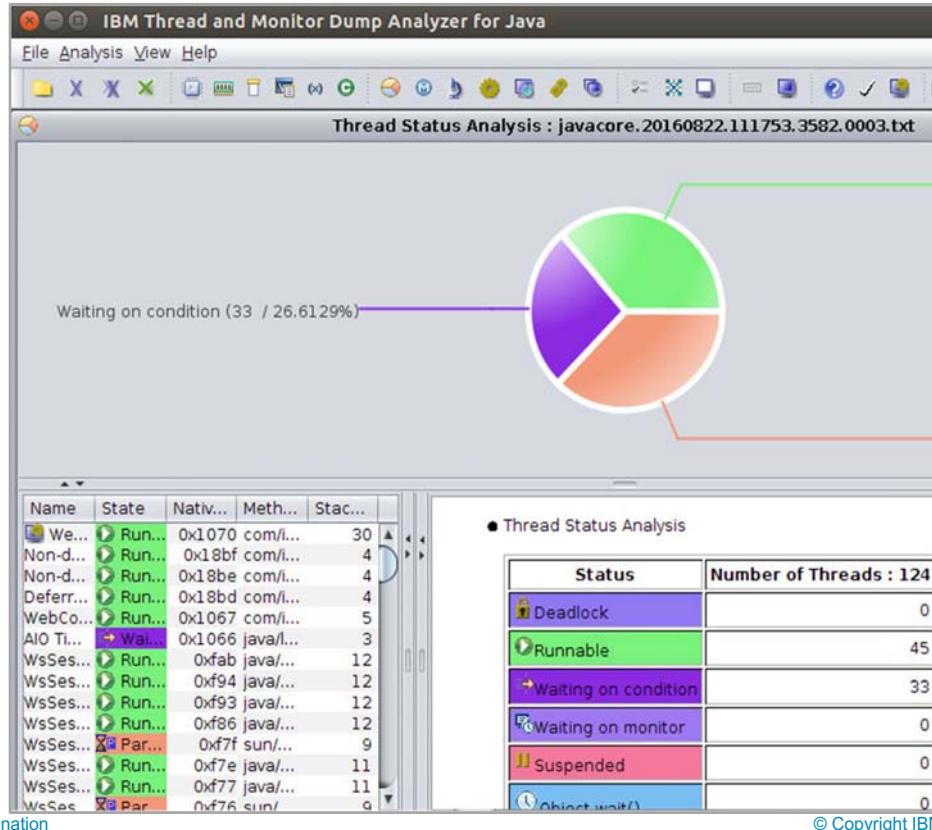
© Copyright IBM Corporation 2016

Figure 7-45. IBM Support Assistant example: Javacore analysis

This image shows an example of a Javacore analysis, specifically the thread dump list. It was run after the BadApp that you use in the lab exercise on exhausted memory. Notice that the cause of this thread dump was an OutOfMemory event.



IBM Support Assistant example: Thread status analysis



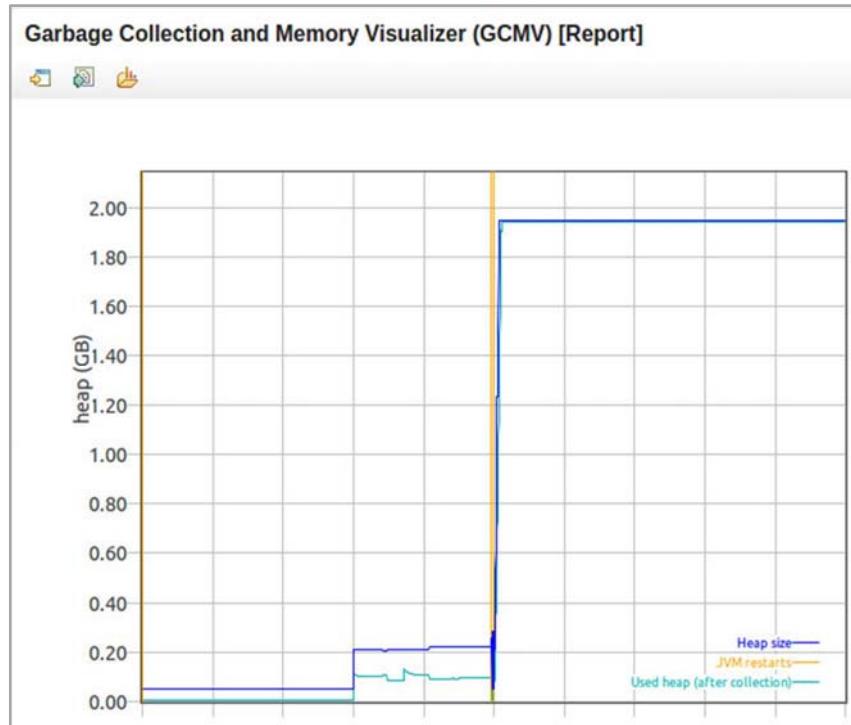
Problem determination

© Copyright IBM Corporation 2016

Figure 7-46. IBM Support Assistant example: Thread status analysis

This screen capture shows the thread analysis, also from a javacore analysis. On the left is a sortable list of all threads, their state, and the method where that thread is running. Clicking any thread name gives you a stack trace of the method and identifies what threads it is waiting on (if any), or what threads are waiting on it. A table also summarizes the number of threads in each state.

IBM Support Assistant example: Memory visualizer



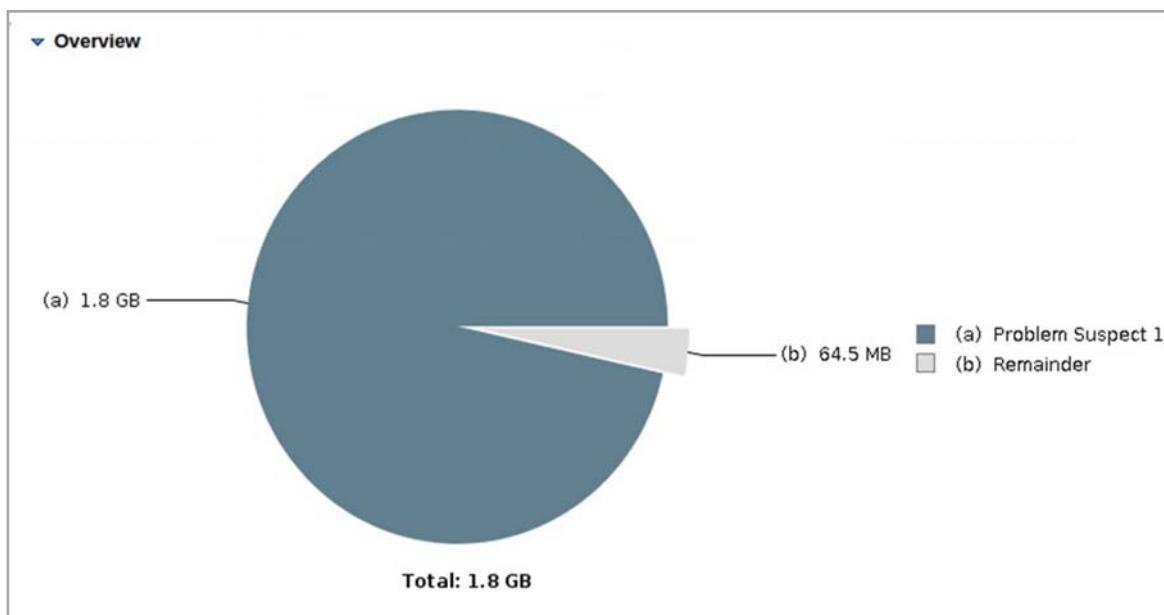
Problem determination

© Copyright IBM Corporation 2016

Figure 7-47. IBM Support Assistant example: Memory visualizer

This image shows an example report of garbage collection and memory visualizer. The memory and heap are depleted in the left side of the graph. This occurs after an out-of-memory event. The yellow line indicates that the JVM restarted, which frees memory. The heap size stays stable at approximately 1.9 GB after reboot.

IBM Support Assistant example: Memory diagnostic



Problem determination

© Copyright IBM Corporation 2016

Figure 7-48. IBM Support Assistant example: Memory diagnostic

The IBM Monitoring and Diagnostic Tools – Interactive Diagnostic Data Explorer is a Java heap analyzer. The diagnostic tool helps identify memory leaks and reduce memory consumption. In this pie chart, one problem suspect is using 1.8 gigabytes with approximately 65 megabytes available. It also was run after the BadApp used too much memory.

Unit summary

- Describe a basic approach for problem determination
- List resources for completing a problem investigation
- Locate relevant log files
- Examine log activity
- Enable tracing on specific components
- Enable High Performance Extensible Logging (HPEL) and use the Log Viewer
- Describe features of the IBM Support Assistant

Problem determination

© Copyright IBM Corporation 2016

Figure 7-49. Unit summary

Review questions

1. What are the two approaches for problem determination?
2. In basic mode logging, what are the two types of JVM log files, and what is the data that they contain?
3. What are the three levels of diagnostic trace detail?
4. What are the three major features of the IBM Support Assistant that are accessible from the Welcome screen?
5. What are some JVM-related problems that a server can experience?



Problem determination

© Copyright IBM Corporation 2016

Figure 7-50. Review questions

Write your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

Review answers

1. What are the two approaches for problem determination?
The answer is: [Analysis and isolation](#)

2. In basic mode logging, what are the two types of JVM log files, and what is the data that they contain?
The answer is: [The SystemOut.log file contains messages from different server components. Mostly informational, these messages log events that occur during the lifetime of the JVM. The SystemErr.log file contains any exceptions and stack traces that server components throw.](#)
3. What are the three levels of diagnostic trace detail?
The answer is: [Fine, finer, and finest](#)
4. What are the three major features of the IBM Support Assistant that are accessible from the Welcome screen?
The answer is: [Find Information, Analyze Problem, Collect, and Send Data](#)
5. What are some JVM-related problems that a server can experience?
The answer is: [Hung threads, OutOfMemory conditions, connect leaks](#)

[Problem determination](#)

© Copyright IBM Corporation 2016

Figure 7-51. Review answers

Exercise: Problem determination

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 7-52. Exercise: Problem determination

Exercise objectives

After completing this exercise, you should be able to:

- Use the administrative console to configure and view log data
- Enable a server to use HPEL
- Enable tracing on application server components
- Use the HPEL Log Viewer to examine log and trace data
- Enable verbose garbage collection for an application server
- Enable memory leak detection for an application server
- Describe features of IBM Support Assistant tools

Unit 8. Introduction to wsadmin and scripting

Estimated time

01:00

Overview

This unit introduces you to the capabilities of wsadmin as an administrative tool for WebSphere Application Server.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server V9 documentation in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Use wsadmin to enter administrative commands
- Create Jython scripts to run wsadmin commands
- Customize the wsadmin environment with profiles and property files
- Use property file-based configurations to modify an environment
- Configure and use command assist to develop wsadmin scripts

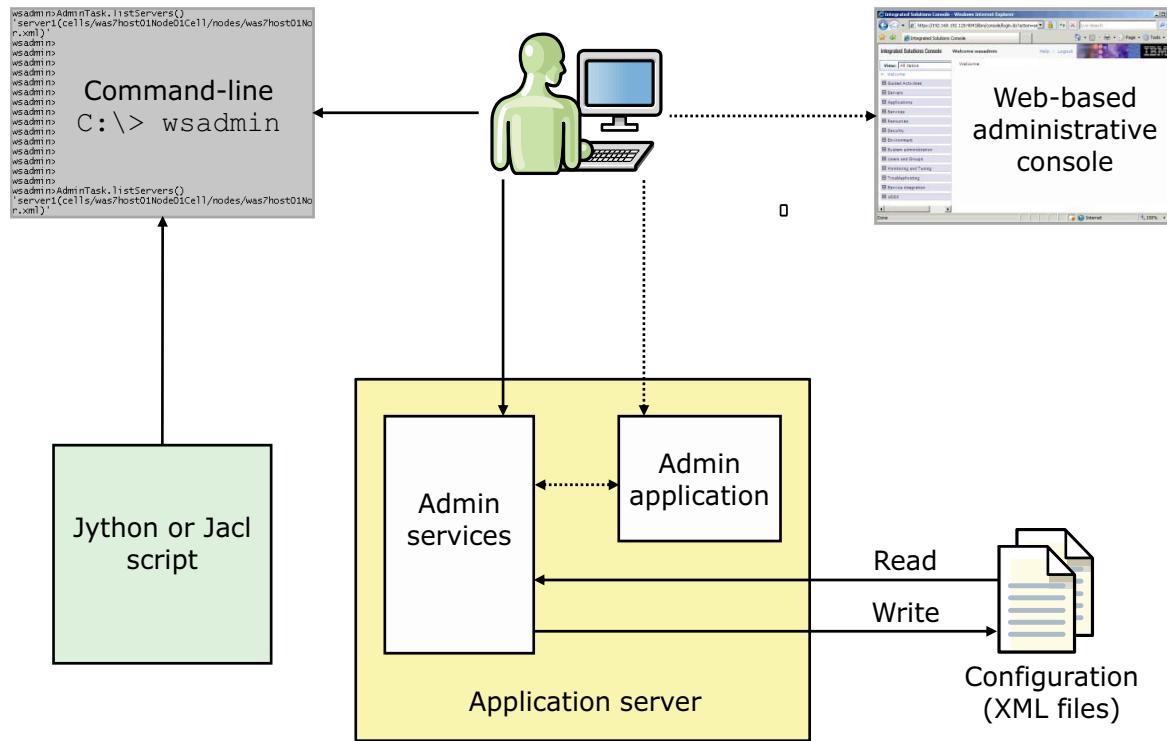
Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-1. Unit objectives



wsadmin versus administration console



Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-2. wsadmin versus administration console

You can use the administrative console or wsadmin tool to manage an application server and the configuration, application deployment, and server runtime operations.

The administrative console is a graphical interface that you can use to manage your applications and do system administration tasks for your WebSphere Application Server environment. The administrative console runs in your web browser.

The wsadmin tool is a command-line client that runs Jython or Jacl scripts that you can use to manage your applications and do system administration tasks for your WebSphere Application Server environment.

Your actions that use the administrative console or wsadmin can access (read) and modify (write) a set of XML configuration files that are used to describe the application server environment.

Scripting benefits

- Automation of routine administration tasks
- Schedule administration tasks
- Support changes in production environments, especially with multiple nodes
- Apply configuration changes or updates to targeted resources in an efficient and consistent manner
- Safer than the administrative console in production environments

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-3. Scripting benefits

There are several benefits for using scripting to manage an application server and the configuration, application deployment, and server runtime operations:

- Automation of routine administration tasks without relying on an operator to run the administrative console
- Schedule administration tasks to run at times when it might be inconvenient to have an operator to run the administrative console
- Support changes in production environments, especially with multiple nodes
- Apply configuration changes and updates to all targeted resources in an efficient and consistent manner
- Using wsadmin is safer than using the administrative console in production environments.

wsadmin

- The wsadmin tool provides:
 - Scripting capabilities
 - Command-line administration
- Support for:
 - Python commands through Jython
 - Tcl commands through Java Command Language (Jacl)
- Modes:
 - Interactive
 - Command-line
 - Script file
- Examples:
 - Start and stop deployment manager, nodes, application servers, enterprise applications, and clusters
 - Configure virtual hosts, JDBC providers, JMS resources
 - Create application servers
 - Create clusters and add members to a cluster

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-4. wsadmin

The WebSphere administrative (wsadmin) scripting program is a powerful, non-graphical command interpreter environment where you can run administrative operations in a scripting language. The wsadmin tool is intended for production environments and unattended operations. The wsadmin tool provides:

- Scripting capabilities
- Command-line administration

The wsadmin tool uses the Bean Scripting Framework (BSF), which supports various scripting languages to configure and control your WebSphere Application Server installation. The wsadmin tool supports:

- Python commands through Jython
- Tcl commands through Java Command Language (Jacl)

The wsadmin launcher makes administrative objects available through language-specific interfaces. Scripts use these objects for application management, configuration, operational control, and communication with MBeans that are running in WebSphere server processes. Three modes are used to start wsadmin:

- Interactive

- Command line
- Script file

You can use the wsadmin tool to do the same tasks that you can use the administrative console for.

You can use the wsadmin tool to manage a WebSphere Application Server V8 installation.

Examples of tasks include:

- Start and stop deployment manager, nodes, application servers, enterprise applications, and clusters
- Configure virtual hosts, JDBC providers, JMS resources
- Create application servers
- Create clusters and add members to a cluster

wsadmin invocation options

- wsadmin invocation options include:

```
wsadmin
-h (help) or -?
-c <command>
-f <script_file_name>
-p <properties_file_name>
-profile <profile_script_name>
-profileName <profile_name>
-lang
-conntype [SOAP | RMI | JSR160RMI | IPC | NONE ]
-host
-user or username
-password
-port
```

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-5. wsadmin invocation options

There are several wsadmin invocation options available. Use the following command-line invocation syntax for the wsadmin scripting client:

- **-h, -help, -?**
Provides syntax help.
- **-c <command>**
Specifies to run a single command. Multiple **-c** options can exist on the command line. They run in the order that you designate. If you start the wsadmin tool with the **-c** option, any changes that you make to the configuration are saved automatically.
- **-f <script_file_name>**
Specifies a script to run. Only one **-f** option can exist on the command line. You can use the **-f** option to run scripts that contain nested Jython scripts. The second level is the user default, `wsadmin.properties`, which is in your home directory.
- **-p <properties_file_name>**
Specifies a properties file. The file that is listed after **-p** represents a Java properties file that the scripting process reads. Three levels of default properties files load before the properties file that you specify on the command line.

- The first level is the installation default, `wsadmin.properties`, which is in the product properties directory.
- The second level is the user default, `wsadmin.properties`, which is in your home directory.
- The third level is the properties file that the environment variable `WSADMIN_PROPERTIES` references.

Multiple `-p` options can exist on the command line. Those options start in the order that you supply them.

- **`-profile <profile_script_name>`**

Specifies a profile script. The profile script runs before other commands, or scripts. If you specify `-c`, then the profile script runs before it starts this command. If you specify `-f`, then the profile script runs before it runs the script. In interactive mode, you can use the profile script to do any standard initialization that you want.

You can specify multiple-profile options on the command line, and they start in the order that you supply them.

- **`-profileName <profile_name>`**

Specifies the profile from which the wsadmin tool runs. Specify this option when one of the following reasons applies:

- You run the wsadmin tool from the `<was_root>/bin` directory, and you do not have a default profile, or you want to run in a profile other than the default profile.
- You are currently in a profile `bin` directory but want to run the wsadmin tool from a different profile.

- **`-lang`**

Specifies the language of the script file, the command, or an interactive shell. The possible languages include Jacl and Jython. The options for the `-lang` argument include `jacl` and `jython`.

This option overrides language determinations that are based on a script file name, a profile script file name, or the `com.ibm.ws.scripting.defaultLang` property. The `-lang` argument has no default value.

If you do not specify the `-lang` argument but you have the `-f <script_file_name>` argument, then the wsadmin tool determines the language, which is based on a target script file name. If you do not specify the `-lang` argument and the `-f` argument, the wsadmin tool determines the language, which is based on a profile script file name if the `-profile <profile_script_name>` argument is specified. If the command line or the property does not supply the script language, and the wsadmin tool cannot determine it, then an error message is generated.

- **`-conntype`**

Specifies the type of connection to use. This argument consists of a string that determines the

type – for example, SOAP – and the options that are specific to that connection type. Possible types include:

- SOAP
- RMI
- JSR160RMI
- IPC
- NONE

For each connection type, you can specify more attributes about the connection.

An example of when `NONE` can be used is when the application server configuration files are so corrupted that the application server cannot be started to access the administrative tasks of the running server.

- `-host`
Specifies a host name to which wsadmin attempts to connect. If this option is not specified, the default `wsadmin.properties` file that is in the properties directory of each profile provides `localhost` as the value of the host property.
- `-user` or `-username`
If security is enabled in the server, this option specifies a user name that the connector uses to connect to the server.
- `-password`
If security is enabled in the server, this option specifies a password that the connector uses to connect to the server.
- `-port`
Specifies a port that the connector uses. The default `wsadmin.properties` file that is in the properties directory of each application server profile provides a value in the port property to connect to the local server.

Jython versus Jacl

Jython syntax is more natural to Java or C programmers

- Jython provides much better support in tools
 - Rational Application Developer (Jython editor, command completion, debugger)
 - WebSphere administrative console (console command assistance)
- wsadmin scripts use Jython 2.7 in version 9

Jacl is deprecated in version 9

- No future development or enhancements for Jacl
- Jacl-to-Jython (Jacl2Jython) conversion assist tool is available

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-6. Jython versus Jacl

The wsadmin tool supports Jython and Jacl scripting languages.

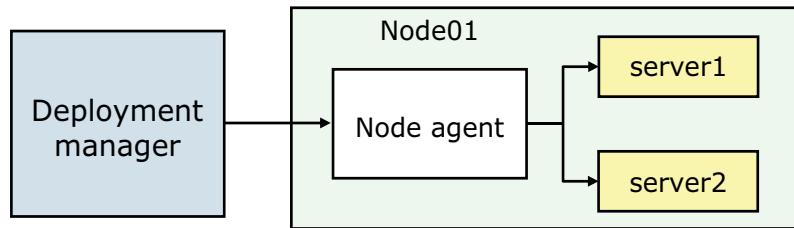
- **Jython scripting language:** Jython is an alternative implementation of Python, and is written entirely in Java.
 - The wsadmin tool uses Jython V2.7 in version 9. Version 8.5.5 used Jython 2.1
 - Jython syntax might seem more natural to Java or C programmers.
 - Future investment and strategic direction focus on the Jython language. Jython has much better support for tools:
 - IBM WebSphere Application Server Developer Tools for Eclipse (Jython editor, command completion, debugger)
 - Administrative console (console command assistance)
 - The product uses a Jython version that does not support Microsoft Windows 2003 or Windows Vista operating systems.
- **Jacl scripting language:** Jacl is deprecated in version 9.

Jacl-to-Jython (Jacl2Jython) is a conversion utility that converts Jacl syntax wsadmin scripts into equivalent Jython syntax wsadmin scripts.

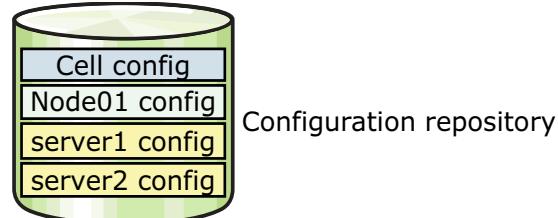
Administrative functions that use wsadmin

- WebSphere Application Server system management separates administrative functions into three categories:

- 1 Running objects in WebSphere Application Server installations



- 2 Configuration of WebSphere Application Server installations (repository)



- 3 Installed applications



[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-7. Administrative functions that use wsadmin

WebSphere Application Server system management separates administration functions into two categories:

- **Running objects in WebSphere Application Server installations:** The wsadmin tool can be used to install, uninstall, and manage applications. You can use either of two methods to install, uninstall, and manage applications.
 - You can use the commands for the **AdminApp** and **AdminControl** objects to start operations on your application configuration.
 - Alternatively, you can use the **AdminApplication** and **BLAManagement** Jython script libraries to do specific operations to configure your enterprise and business-level applications.
- **Configuration of WebSphere Application Server installations (repository):** The wsadmin tool can be used to configure application servers in your environment.
 - You can use the commands for the **AdminConfig** and **AdminTask** objects to retrieve configuration IDs and start operations on the objects to configure the application server.
 - Alternatively, you can use the script library for specific operations to configure your application servers.

An application server configuration provides settings that control how an application server provides services for running applications and their components.

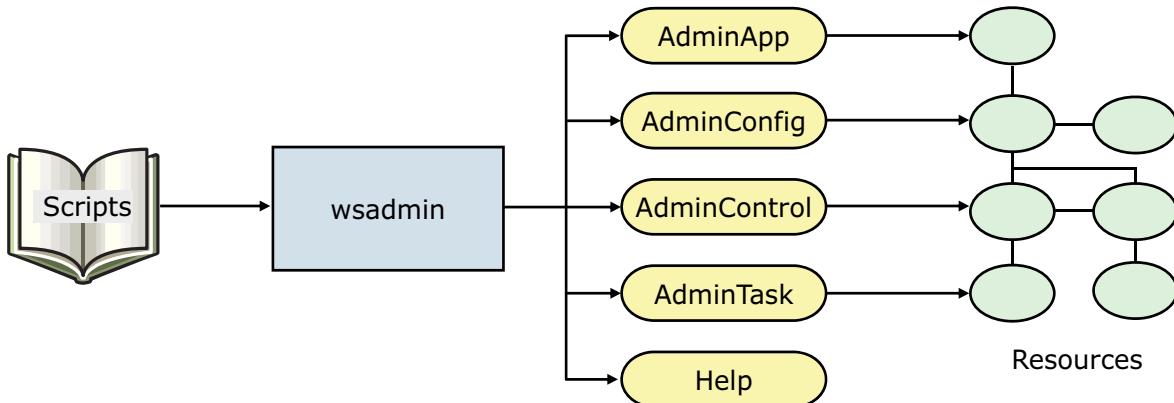
Configuration data is stored in several different XML files, which the server runtime reads when it starts and responds to the component settings stored there. The configuration data includes the settings for the runtime, such as Java virtual machine (JVM) options, thread pool sizes, container settings, and port numbers that the server uses. Other configuration files define Java Platform, Enterprise Edition resources to which the server connects to obtain data that the application logic needs.

Security settings are stored in a separate document from the server and resource configuration. Application-specific configurations, such as deployment target lists, session configuration, and cache settings, are stored in files under the root directory of each application. When viewing the XML data in the configuration files, you can discern the relationship between the configuration objects.

For information about the WebSphere Application Server configuration objects, view the HTML tables in the `<was_root>/web/configDocs` directory. This directory has several subdirectories, one for each configuration package in the model. The `index.html` file ties all of the individual configuration packages together in a top-level navigation tree.

Each configuration package lists the supported configuration classes, and each configuration class lists all of the supported properties. The properties with names that end with the `@` character imply that property is a reference to a different configuration object within the configuration data. The properties with names that end with the `*` character imply that the property is a list of other configuration objects.

Administrative objects in wsadmin



Five administrative objects do different operations:

- **AdminControl:** Work with “live” running objects
- **AdminConfig:** Create or modify WebSphere Application Server “static” configuration
- **AdminApp:** Install, modify, or administer applications
- **AdminTask:** Administrative commands that are more usable and task-oriented
- **Help:** General help

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-8. Administrative objects in wsadmin

The wsadmin tool acts as an interface to Java objects by way of scripts to access resources. The tool uses the same interface (through JMX) as the administrative console to make configuration changes and control servers.

Five objects are available when you use scripts:

- **AdminControl:** Use AdminControl to run operational commands, work with “live” running objects, and do traces or data type conversions.
- **AdminConfig:** Use AdminConfig to run configuration commands to create or modify WebSphere Application Server “static” configurational elements.
- **AdminApp:** Use AdminApp to install, modify, or administer applications.
- **AdminTask:** Use AdminTask to run administrative commands that are easier to use and more task-oriented.
- **Help:** Use Help to obtain general help.



Starting wsadmin

- Interactively

wsadmin

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin
WASX7209I: Connected to process "server1" on node was7host01Node01 using SOAP connector; The type of process is: UnManagedProcess
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

- Command option

wsadmin -c <command>

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -c print(AdminApp.list())
WASX7209I: Connected to process "server1" on node was7host01Node01 using SOAP connector; The type of process is: UnManagedProcess
DefaultApplication
listApp
query
```

- Script file

wsadmin -f <script_file>

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -f "C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin\initialSIBSetup.py"
```

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-9. Starting wsadmin

The wsadmin tool can be started in three ways:

- **Interactively:** wsadmin

Run wsadmin with an option other than `-f` or `-c` or without an option. The wsadmin tool starts and provides an interactive shell with a wsadmin prompt. From the wsadmin prompt, enter any Jacl or Jython command. You can also use the AdminControl, AdminApp, AdminConfig, AdminTask, or Help wsadmin objects to start commands. To leave an interactive scripting session, use the `quit` or `exit` command. The graphic demonstrates wsadmin in an interactive mode:

```
<was_root>\profiles\profile1\bin\wsadmin
```

- **Command option:** wsadmin -c <command>

Run the wsadmin tool with the noninteractive `-c` option. The graphic demonstrates wsadmin by using the command option to run the `list` command for the `AdminApp` object that uses Jython.

```
<was_root>\profiles\profile1\bin\wsadmin -c print(AdminApp.list()) -lang jython
```

- **Script file:** wsadmin -f <script_file>

Run the wsadmin tool with the noninteractive `-f` option, and place the commands that you

want to run into a script file. The graphic demonstrates wsadmin by using the script file option to run a Jython script.

```
<was_root>\profiles\profile1\bin\wsadmin -f c:\Program Files\IBM\WebSphere\  
AppServer\profiles\profile1\bin\initialSIBSetup.jy
```

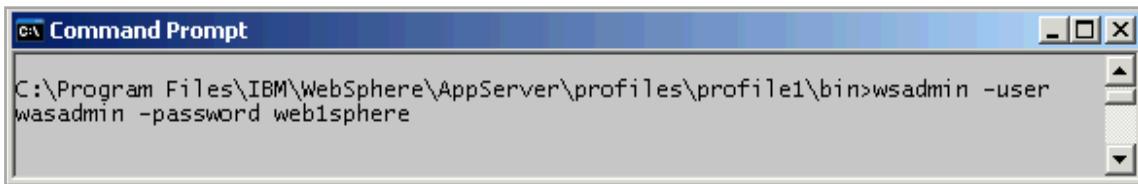
When running `wsadmin` with the `-f` option, you are not required to specify the script language with the `-lang` command-line option. The `wsadmin` command recognizes the script language by looking at the extension of the file name (`.py` for Jython and `.jacl` for Jacl).

Starting wsadmin with security enabled

- If security is enabled, authentication information must be supplied
- Authentication information can be provided in several ways:
 - Prompted



- Command-line parameters



- RMI connections: `sas.client.props` file
- SOAP connections: `soap.client.props` file

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-10. Starting wsadmin with security enabled

WebSphere Application Server console security is enabled by default.

- If security is enabled, authentication information must be supplied.
- Authentication information can be provided in several ways:
 - Prompted:
If security is enabled and you do not provide credentials when you start wsadmin, you are prompted to provide them. The **Login at the Target Server** window opens. Two fields, **User Identity** and **User Password**, must be entered with valid credentials before proceeding.

- Command-line parameters:

You can use command-line parameters to pass authentication credentials.

- `-user` or `-username` can be used interchangeably to provide a user ID.
- `-password` is used to provide a password.

The graphic demonstrates the use of command-line parameters to pass authentication credentials:

```
<was_root>\profiles\profile1\bin>wsadmin -user USERID -password PASSWORD
```

- RMI connections: `sas.client.props` file

For RMI connections, authentication credentials can be provided in the `sas.client.props`

file. The file is in the profile properties directory:

<was_root>\profiles\<profile_name>\properties\

The authentication parameters are:

- com.ibm.CORBA.loginUserId
- com.ibm.CORBA.loginPassword

- SOAP connections: soap.client.props file

For SOAP connections, authentication credentials can be provided in the soap.client.props file. The file is in the profile properties directory:

<was_root>\profiles\<profile_name>\properties\

The authentication parameters are:

- com.ibm.SOAP.loginUserId
- com.ibm.SOAP.loginPassword

wsadmin properties

- Certain default behaviors for wsadmin can be changed by editing:

`<profile_root>\<profile_name>\properties\wsadmin.properties`

- Examples of properties include:

```

- com.ibm.ws.scripting.connectionType=SOAP
- com.ibm.ws.scripting.port=8879
- com.ibm.ws.scripting.host=localhost
- com.ibm.ws.scripting.defaultLang=jython
- com.ibm.ws.scripting.traceFile=
- com.ibm.ws.scripting.validationOutput=
- com.ibm.ws.scripting.traceString=com.ibm.*=all=enabled
- com.ibm.ws.scripting.profiles=
- com.ibm.ws.scripting.emitWarningForCustomSecurityPolicy=true
- com.ibm.ws.scripting.tempdir=
- com.ibm.ws.scripting.validationLevel=

```

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-11. wsadmin properties

Scripting administration uses several Java properties files. Properties files can be used to control your system configurations. Before any properties file is specified on the command line, three levels of default properties files are loaded. These properties files include:

- **Installation default file:** The first level represents an installation default, which is in the `profile_root/properties` directory for each application server profile called `wsadmin.properties`.
- **User default file:** The second level represents a user default, and is in the Java `user.home` property. This properties file is also called `wsadmin.properties`.
- **Properties file:** The third level is a properties file that is pointed to by the `WSADMIN_PROPERTIES` environment variable. This environment variable is defined in the environment where the wsadmin tool starts.

If one or more of these properties files are present, they are interpreted before any properties file that is present on the command line. The three levels of properties files load in the order that they are specified. The properties file that is loaded last overrides the ones that are loaded earlier.

Certain default behaviors for wsadmin can be changed by editing:

`<profile_root>\<profile_name>\properties\wsadmin.properties`

Examples of properties include:

- `com.ibm.com.ws.scripting.connectionType=SOAP`
Determines the connector to use. This value can either be SOAP, JSR160RMI, RMI, IPC, or NONE. The `wsadmin.properties` file specifies SOAP as the connector.
- `com.ibm.com.ws.scripting.port=8880`
Specifies the port to use when attempting a connection. The `wsadmin.properties` file specifies 8879 as the SOAP port for a single-server installation.
- `com.ibm.com.ws.scripting.host=localhost`
Determines the host to use when attempting a connection. If not specified, the default is the local host.
- `com.ibm.com.ws.scripting.defaultLang=jython`
Indicates the language to use when running scripts. The `wsadmin.properties` file specifies Jacl as the scripting language. The supported scripting languages are Jacl and Jython.
If you change this value to Jython, some WebSphere Application Server scripts that depend on the default language of Jacl fail unless rewritten to call `wsadmin` with the `-lang jacl` command-line option.
- `com.ibm.com.ws.scripting.traceFile=`
Determines where trace and log output is directed. The `wsadmin.properties` file specifies the `wsadmin.traceout` file that is in the `<profile_root>/logs` directory of each application server profile as the value of this property.
- `com.ibm.com.ws.scripting.validationOutput=`
Determines where the validation reports are directed. The default file is `wsadmin.valout`, which is in the `<profile_root>/logs` directory of each application server profile.
- `com.ibm.com.ws.scripting.traceString=com.ibm.*=all=enabled`
Turns on tracing for the scripting process. The default has tracing turned off.
- `com.ibm.com.ws.scripting.profiles=`
Specifies a list of profile scripts to run automatically before running user commands, scripts, or an interactive shell.
- `com.ibm.com.ws.scripting.emitWarningForCustomSecurityPolicy=true`
Controls whether the WASX7207W message is emitted when custom permissions are found. The possible values are true and false. The default value is true.
- `com.ibm.com.ws.scripting.tempdir=`
Determines the directory to use for temporary files when installing applications. The Java virtual machine (JVM) API uses `java.io.temp` as the default value.
- `com.ibm.com.ws.scripting.validationLevel=`
Determines the level of validation to use when configuration changes are made from the scripting interface. Possible values are: NONE, LOW, MEDIUM, HIGH, and HIGHEST. The default is HIGHEST.
- `com.ibm.com.ws.scripting.crossDocumentValidationEnabled=`
Determines whether the validation mechanism examines other documents when changes are made to one document. Possible values are true and false. The default value is true.

- `com.ibm.com.ws.scripting.classpath=`
Searches for classes and resources, and is appended to the list of paths.

Profile scripts

- Profile scripts can be used to preset wsadmin with predefined settings and functions
- Run during wsadmin startup
- A profile script can be called in either of the following ways:
 - Using the `-profile` option on the command line
 - Defined in `wsadmin.properties com.ibm.ws.scriptingprofiles=`

```
#-----
# Print whereAMI
#-----
def whereAMI():
    #Print cell and node names
    print "Cell: " + AdminConfig.showAttribute(AdminConfig.list("Cell"), "name")
    print "Node: " + AdminConfig.showAttribute(AdminConfig.list("Node"), "name")
    return

#-----
# Start of main
#-----
print ""
print "Hello, and welcome to wsadmin using jython"
print ""
print "Running global_profile.py Global definitions and settings could be added"
print "here. It would also be possible to extend wsadmin by defining new"
print "customized commands and procedures."
print ""

whereAMI()
print ""
```

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-12. Profile scripts

Profile scripts can be used to preload wsadmin with predefined settings and functions. Profile scripts are run during wsadmin startup. They can be called in one of two ways:

- Use the `-profile` option on the command line.
- Use the `com.ibm.ws.scriptingprofiles=` property to define the profile script in `wsadmin.properties`.

The profile script runs before other commands, or scripts. If you specify `-c`, then the profile script runs before it starts this command. If you specify `-f`, then the profile script runs before it runs the script. In interactive mode, you can use the profile script to do any standard initialization that you want.

You can specify multiple `-profile` options on the command line, and they start in the order that you supply them.

The graphic example identifies a Jython script that is named `global_profile.py` with the following syntax:

```
#-----
# Print whereAMI
#-----
def whereAMI():
    #Print cell and node names
    print "Cell: " + AdminConfig.showAttribute(AdminConfig.list("Cell"), "name")
    print "Node: " + AdminConfig.showAttribute(AdminConfig.list("Node"), "name")
    return
#-----
# Start of main
#-----
print ""
print "Hello, and welcome to wsadmin using jython"
print ""
print "Running global_profile.py Global definitions and settings could be added"
print "here. It would also be possible to extend wsadmin by defining new"
print "customized commands and procedures."
print ""
whereAMI()
print ""
```

The following example output demonstrates that wsadmin started by using the `-profile` option to run the `global_profile.py` script:

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin\wsadmin -profile
c:\software\wsadmin\global_profile.py
WASX7209I: Connected to process "server1" on node was7host01Node01 using SOAP
connector; The type of process is: UnManagedProcess
Hello, and welcome to wsadmin using jython
Running global_profile.py Global definitions and settings could be added
here. It would also be possible to extend wsadmin by defining new
customized commands and procedures.
Cell: was7host01Node01Cell
Node: was7host01Node01
WASX7031I: For help, enter: "print Help.help()"
```

AdminConfig: Managing configurations

- Management configuration scripts use the AdminConfig object to access the repository where configuration information is stored
- Example:

```
wsadmin> AdminConfig.list ('Server')
'server(cells/was8host01Node01Cell/nodes/washost01Node01/servers/server1|
  server.xml#Server_1183122130078
wsadmin>
```

- Use the AdminConfig object to:
 - List configuration objects and their attributes
 - Create configuration objects
 - Modify configuration objects
 - Remove configuration objects
 - Obtain help

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-13. AdminConfig: Managing configurations

Management configuration scripts use the AdminConfig object to access the repository where configuration information is stored.

An example of AdminConfig is:

```
wsadmin>AdminConfig.list ('Server')
```

You can use the AdminConfig object to:

- List configuration objects and their attributes
- Create configuration objects
- Modify configuration objects
- Remove configuration objects
- Obtain help

AdminApp: Managing applications

- Application management scripts use the AdminApp object to manage applications in the application server configuration
- Example:

```
ca Command Prompt - wsadmin -profile C:\software\wsadmin\global_profile.py
wsadmin>print AdminApp.view('ivtApp')
Specifying application options
Specify the various options that are available to prepare and install your application.

Directory to install application: ${APP_INSTALL_ROOT}/${CELL}
Distribute application: Yes
Use Binary Configuration: No
Create MBeans for resources: No
Override class reloading settings for Web and EJB modules: No
Reload interval in seconds:
Validate Input off/warn/fail: off
File Permission: .*\*.dll=755#.*\*.so=755#.*\*.a=755#.*\*.sl=755
```

- Use the AdminApp object to:
 - Install and uninstall applications
 - List installed applications
 - Edit application configurations
 - Obtain help

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-14. AdminApp: Managing applications

Application management scripts use the AdminApp object to manage applications in the application server configuration.

An example of AdminApp is:

```
wsadmin>AdminApp.view ('ivtApp')
```

Running the command produces output specific to the application. In the Jython command, output is easier to read when using the `print` option:

```
wsadmin>print AdminApp.view ('ivtApp')
```

You can use the AdminApp object to:

- Install and uninstall applications
- List installed applications
- Edit application configurations
- Obtain help

AdminControl: Managing running objects

- Operation management scripts use the AdminControl object to communicate with the MBeans that represent running objects
- Example:

```
Terminal
File Edit View Terminal Tabs Help
wsadmin>
wsadmin>traceMBean=AdminControl.completeObjectName("type=TraceService,process=server1,*")
wsadmin>AdminControl.invoke(traceMBean,"appendTraceString","WAS.j2c=finest")
''
wsadmin>
```

- Use the AdminControl object to:
 - List running objects and their attributes
 - Start actions on running objects
 - Obtain dynamic information about MBeans that represent running objects
 - Obtain help

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-15. AdminControl: Managing running objects

Operation management scripts use the AdminControl object to communicate with the MBeans that represent running objects.

An example of AdminControl is:

```
wsadmin>AdminControl.getPort()
```

You can use the AdminControl object to:

- List running objects and their attributes
- Start actions on running objects
- Obtain dynamic information about MBeans that represent running objects
- Obtain help

The number and type of MBeans available to the scripting client depend on the server to which the client is connected. If the client is connected to a deployment manager, then all the MBeans that are running in the deployment manager are visible. All the MBeans running in the node agents that are connected to this deployment manager, and all the MBeans running in the application servers on those nodes, are visible.

AdminTask: Accessing administrative functions

- AdminTask object is used to access a set of administrative commands to provide an alternative way to access configuration commands
- Example:

```
Terminal
File Edit View Terminal Tabs Help
wsadmin>
wsadmin>print AdminTask.listServerPorts("server1")
[[SOAP_CONNECTOR_ADDRESS [[[host was85host] [node was85hostNode01] [server server1] [port 8880] ]]] ]
[[SIP_DEFAULTHOST_SECURE [[[host *] [node was85hostNode01] [server server1] [port 5061] ]]] ]
[[SIP_DEFAULTHOST [[[host *] [node was85hostNode01] [server server1] [port 5060] ]]] ]
[[SIB_ENDPOINT_ADDRESS [[[host *] [node was85hostNode01] [server server1] [port 7276] ]]] ]
[[WC_defaulthost_secure [[[host *] [node was85hostNode01] [server server1] [port 9443] ]]] ]
[[DCS_UNICAST_ADDRESS [[[host *] [node was85hostNode01] [server server1] [port 9353] ]]] ]
[[SIB_MQ_ENDPOINT_SECURE_ADDRESS [[[host *] [node was85hostNode01] [server server1] [port 5578] ]]] ]
```

- Benefits of using AdminTask:
 - Provides more usable and task-oriented commands
 - Runs simple and complex commands
 - Commands grouped based on function
 - Can be run in batch or interactive mode
 - Can be run in connected or local mode

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-16. AdminTask: Accessing administrative functions

The AdminTask object is used to access a set of administrative commands that provide an alternative and easier way to access configuration commands. You can use AdminTask commands to run a single command for administrative actions that might otherwise require multiple commands.

Some examples of AdminControl are:

```
wsadmin>AdminTask.binaryAuditReader( '-interactive' )
wsadmin>AdminTask.listServers( )
```

The benefits of using AdminTask include:

- Provides commands that are more task-oriented and easier to use
- Runs simple and complex commands
- Commands are grouped, based on function
- Can be run in batch or interactive mode
- Can be run in connected or local mode

The administrative commands are discovered dynamically when you start a scripting client. The set of available administrative commands depends on the edition of WebSphere Application Server you install. You can use the AdminTask object commands to access these commands.

Administrative commands are grouped based on their function. You can use administrative command groups to find related commands. For example, the administrative commands that are related to server management are grouped into a server management command group. The administrative commands that are related to the security management are grouped into a security management command group. An administrative command can be associated with multiple command groups because it can be useful for multiple areas of system management. Both administrative commands and administrative command groups have names that uniquely identify them.

Two run modes are always available for each administrative command. These run modes are the batch and interactive modes. When you use an administrative command in interactive mode, you go through a series of steps to collect your input interactively. This process provides users a text-based wizard and a similar user experience to the wizard in the administrative console. You can also use the `help` command to obtain help for any administrative command and the `AdminTask` object.

The administrative commands do not replace any existing configuration commands or running object management commands, but provide a way to access these commands and organize the inputs. The administrative commands can be available in connected or local mode. The set of available administrative commands is determined when you start a scripting client in connected or local mode. If a server is running, do not run the scripting client in local mode because any configuration changes made in local mode are not reflected in the running server configuration, and the reverse is also true. If you save a conflicting configuration, you can corrupt the configuration.

Help within wsadmin

Jython

- print Help.help()
- print Help.AdminConfig()
- print Help.AdminTask()
- print Help.AdminControl()
- print Help.AdminApp()

```
wsadmin>print Help.help()
WASX7028I: The Help object has two purposes:

First, provide general help information for the the objects
supplied by wsadmin for scripting: Help, AdminApp, AdminConfig,
and AdminControl.

Second, provide a means to obtain interface information about
MBeans running in the system. For this purpose, a variety of
commands are available to get information about the operations,
attributes, and other interface information about particular
MBeans.

The following commands are supported by Help; more detailed
information about each of these commands is available by using the
"help" command of Help and supplying the name of the command
as an argument.

attributes      given an MBean, returns help for attributes
operations     given an MBean, returns help for operations
constructors   given an MBean, returns help for constructors
description    given an MBean, returns help for description
notifications  given an MBean, returns help for notifications
```

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-17. Help within wsadmin

You can find general help and dynamic online information about the currently running MBeans with the wsadmin tool. Use the Help object as an aid in writing and running scripts with the AdminControl object.

Some examples of commands available for the Help object are:

- AdminConfig
Use the AdminConfig command to view a summary of each available method for the AdminConfig object.
- AdminTask
Use the AdminTask command to view a summary of help commands and ways to start an administrative command with the AdminTask object.
- AdminControl
Use the AdminControl command to view a summary of the help commands and ways to start an administrative command.
- AdminApp
Use the AdminApp command to view a summary of each available method for the AdminApp object.

- `help`

Use the `help` command to view a summary of all the available methods for the Help object.

Some Jython examples are:

- `print Help.help()`
- `print Help.AdminConfig()`
- `print Help.AdminTask()`
- `print Help.AdminControl()`
- `print Help.AdminApp()`

For complete documentation, see the IBM Knowledge Center. The IBM Knowledge Center also includes a number of examples that demonstrate some useful wsadmin features.

Administrative command help

- You can select from three levels of online help for administrative commands
- Example of command-level help:

```
wsadmin>print AdminTask.help('listServers')
WASX8006I: Detailed help for command: listServers

Description: list servers of specified server type and node name. If node name is not specified, whole cell will be searched. If the server type is not specified servers of all types are returned.

Target object: None

Arguments:
  serverType - The ServerType ie: (APPLICATION_SERVER)
  nodeName - The Node Name
```

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-18. Administrative command help

You can select from three levels of online help for administrative commands.

- Top-level help: Provides general information for the AdminTask object and associated commands.

```
print AdminTask.help()
```

- Second-level help: Provides information about all of the available administrative commands and command groups.

```
print AdminTask.help( '-commands' )
```

- Third-level help: Provides specific help on a command group, a command, or a step. Command group-specific help provides descriptions for the command group that you specify and the commands that belong to the associated group. Command-specific help provides a description of the specified command, and associated parameters and steps.

Step-specific help provides a description of the specified step and the associated parameters. For command and step-specific help, required parameters are marked with an asterisk in the help output.

```
print AdminTask.help( 'listServers' )
```

Important points to remember when using wsadmin

- Commands are case-sensitive
- Running multiple commands in a script file is faster than running individual commands
- Saving configuration changes is a two-step process:
 - The first step validates the changes
 - The second step completes the save
- Save periodically:

```
wsadmin> AdminConfig.save()
' '
wsadmin>
```

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-19. Important points to remember when using wsadmin

When using wsadmin, remember the following important points:

- Commands are case-sensitive.
- Running multiple commands in a script file is faster than running individual commands. For example, `wsadmin -f "script_file_name"` is faster than individual commands with `wsadmin -c`
- Saving configuration changes is a two-step process:
 - The first step validates the changes.
 - The second step saves the changes.
- Save periodically in the script file or interactive mode to persist configuration updates to existing objects:

`AdminConfig.save()`

Scripting: Simple script

```

print "Simple wsadmin scripting example."

cell = AdminConfig.list("Cell")
node = AdminConfig.list("Node")

cellName = AdminConfig.showAttribute(cell, "name")
nodeName = AdminConfig.showAttribute(node, "name")

print ""
print "Cell name is: " + cellName
print "Node name is: " + nodeName

```



```

C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -f
"C:\software\wsadmin\simple_script.py" -username wasadmin -password
*****
WASX7209I: Connected to process "server1" on node washost01Node01 using
SOAP connector: The type of process is: UnManagedProcess
Simple wsadmin scripting example

Cell name is: washost01Node01Cell
Node name is: washost01Node01

```

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-20. Scripting: Simple script

The following is an example of a simple script:

```

print "Simple wsadmin scripting example."
cell = AdminConfig.list("Cell")
node = AdminConfig.list("Node")
cellName = AdminConfig.showAttribute(cell, "name")
nodeName = AdminConfig.showAttribute(node, "name")
print ""
print "Cell name is: " + cellName
print "Node name is: " + nodeName

```

An example of the output from this script is:

```

WASX7209I: Connected to process "server1" on node was7host01Node01 using SOAP
connector: The type of process is: UnManagedProcess
Simple wsadmin scripting example.
Cell name is: was7host01Node01Cell
Node name is: was7host01Node01

```

Scripting: Looping script

```

print "Simple loop scripting example"

appNames = AdminApp.list()
appNamesArray = appNames.split('\r\n')

for appName in appNamesArray:
    print "app Name: " + appName

```



```

C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -f
"C:\software\wsadmin\simple_script.py" -username wasadmin -password
*****
WASX7209I: Connected to process "server1" on node washost01Node01 using
SOAP connector: The type of process is: UnManagedProcess
Simple loop scripting example

app Name: DefaultApplication
app Name: PlantsByWebSphere
app Name: ivtApp
app Name: query

```

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-21. Scripting: Looping script

The following is an example of a looping script:

```

print "Simple loop script"
print ""
appNames = AdminApp.list()
appNamesArray = appNames.split('\r\n')
for appName in appNamesArray:
    print "App Name: " + appName

```

Here is an example of the output from these scripts:

```

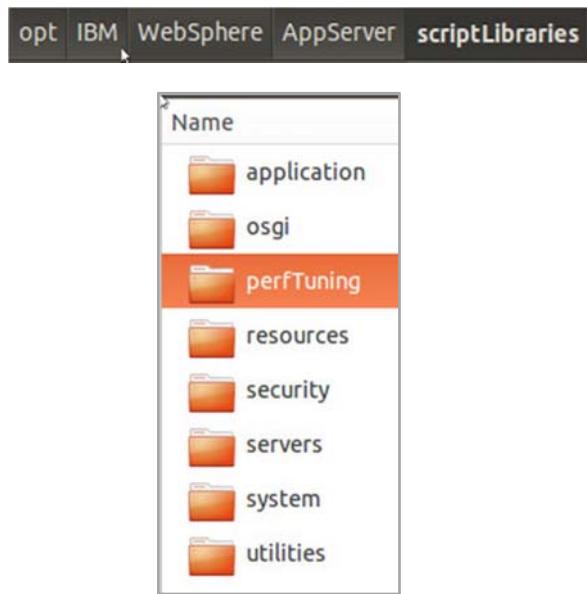
WASX7209I: Connected to process "server1" on node washost01Node01 using SOAP
connector: The type of process is: UnManagedProcess
Simple loop script
App Name: DefaultApplication
App Name: PlantsByWebSphere
App Name: ivtApp
App Name: query

```



Jython script library

- Provides a library of wsadmin Jython scripts for commonly used administrative functions
- Grouped according to administrative function
- One location for learning script syntax
- Supports rapid development of new scripts by combining library scripts with custom code
- IBM Knowledge Center, under Reference
 - `<was_root>/scriptLibraries`



Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-22. Jython script library

The Jython script library provides a set of procedures to automate the most common application server administration functions. For example, you can use the script library to easily configure servers, applications, mail settings, resources, nodes, business-level applications, clusters, authorization groups, and more. You can run each script procedure individually, or combine several procedures to quickly develop new scripts.

The Jython script library helps to eliminate the complexities that are associated with scripting:

- Syntax is complex.
- Incompatible or incorrect parameters are easy to specify.
- Complexity presents a barrier to learning WebSphere.
- Complexity can result in inconsistent and unstable systems that make problem determination difficult.
- wsadmin provides a low level of abstraction that results in complex mappings between operations and problems that users are trying to solve.

Jython script library features include:

- It provides a library of wsadmin Jython scripts for commonly used administrative functions:
 - Application management

- Configuration
- Runtime operations
- Scripts are grouped according to administrative function. Examples of grouped functions include:
 - Utility scripts
 - JDBC query scripts
 - Cluster administration scripts
 - Cluster configuration scripts
 - Server administration scripts
 - Server configuration scripts
- Users can learn script syntax from the script library, rather than by referencing external documentation.
- It supports rapid development of new scripts by combining library scripts with custom code that the user develops.

The Jython script library is in the `<was_root>/scriptLibraries` directory. Information about the script library is available in the IBM Knowledge Center under Reference.

How to use the Jython script library

- The Jython script library is used in three ways:

- Run scripts in interactive mode with the wsadmin tool

```
wsadmin>AdminServerManagement.createApplicationServer("profile1",
"server1", "default")
```

- Use a text editor to combine several scripts

```
# My Custom Jython Script - file.py
AdminServerManagement.createApplicationServer("profile1",
"server1", "default")
AdminServerManagement.createApplicationServer("profile2",
"server2", "default")

# Use one of them as the first member of a cluster
AdminClusterManagement.createClusterWithFirstMember("cluster1",
"APPLICATION_SERVER", "profile1", "server1")

# Install an application
AdminApplication.installAppWithClusterOption("DefaultApplication",
"..\\installableApps\\DefaultApplication.ear", "cluster1")

# Start all servers and applications on the node
AdminServerManagement.startAllServers("profile1")
```

- Use the Jython scripting library code as sample syntax to write custom scripts

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-23. How to use the Jython script library

The scripting library provides a set of procedures to automate the most common application server administration functions. Each script in the script library demonstrates suggested practices for writing wsadmin scripts.

The script library code is in the `<was_root>/scriptLibraries` directory. Within this directory, the scripts are organized into subdirectories according to function, and further organized according to version. For example, the `<was_root>/scriptLibraries/application/V80` subdirectory contains procedures that complete application management tasks that are applicable to version 8.0 and later of the product.

The Jython script library can be used in three ways:

- Interactively:

Run scripts from the Jython script library in interactive mode with the wsadmin tool. You can start the wsadmin tool, and run individual scripts that are included in the script library:

```
wsadmin>AdminServerManagement.createApplicationServer("myNode", "myServer",
"default")
```

- Use a text editor to combine scripts.

Use a text editor to combine several scripts from the Jython script library, as the following example demonstrates:

```
#  
# My Custom Jython Script - file.py  
  
AdminServerManagement.createApplicationServer("profile1", "server1", "default")  
AdminServerManagement.createApplicationServer("profile2", "server2", "default")  
# Use one of them as the first member of a cluster  
AdminClusterManagement.createClusterWithFirstMember("cluster1",  
"APPLICATION_SERVER", "profile1", "server1")  
# Install an application  
AdminApplication.installAppWithClusterOption("DefaultApplication",  
"..\\installableApps\\DefaultApplication.ear", "cluster1")  
# Start all servers and applications on the node  
AdminServerManagement.startAllServers("profile1")
```

- As sample scripts:

Use the Jython scripting library code as sample syntax to write custom scripts.

Configuration repository: The issues

- The repository consists of multiple files in XML and other formats
- The configuration files are spread across many directories
- Configuration objects are complex
- Some configuration objects repeatedly stored in multiple files
- Example: Properties for a JDBC provider

```
<resources.jdbc:JDBCProvider xmi:id="JDBCProvider_1183122153343" name="Derby JDBC Provider">
  <classpath>${DERBY_JDBC_DRIVER_PATH}/derby.jar</classpath>
  <factories xmi:type="resources.jdbc:DataSource" xmi:id="DataSource_1183122153625" name="D
    <propertySet xmi:id="J2EEResourcePropertySet_1183122153625">
      <resourceProperties xmi:id="J2EEResourceProperty_1183122153625" name="databaseName" t
      <resourceProperties xmi:id="J2EEResourceProperty_1183122153626" name="shutdownDatabase
      <resourceProperties xmi:id="J2EEResourceProperty_1183122153627" name="dataSourceName"
      <resourceProperties xmi:id="J2EEResourceProperty_1183122153628" name="description" ty
      <resourceProperties xmi:id="J2EEResourceProperty_1183122153629" name="connectionAttrib
      <resourceProperties xmi:id="J2EEResourceProperty_1183122153630" name="createDatabase"
    </propertySet>
    <connectionPool xmi:id="ConnectionPool
  </factories>
</resources.jdbc:JDBCProvider>
```

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-24. Configuration repository: The issues

In previous releases of WebSphere Application Server, users used wsadmin, the administrative console, and Java APIs to query and modify configurator objects. Now users who work with the WebSphere Application Server configuration repository are confronted with several issues:

- The repository consists of multiple files in XML and other formats.
- The configuration files are spread across many directories.
- Some files contain complex objects that are associated with the WebSphere Common Configuration Model.
- Some configuration objects are repeatedly stored in multiple files.

Properties-based file configuration is a tool to help users deal more easily with these issues. A new set of wsadmin commands is available that can extract and apply properties files to configuration objects.

The graphic shows JDBCProvider object content from the configuration XML file named resources.xml.

Properties file based configuration: A solution

- Properties files are more human readable
 - Properties files consist of name and value pairs
 - Decouples configuration data from changes in the underlying configuration model between releases
 - Can be used with configuration archives
 - Differences between configuration environments are easier to identify

```
wsadmin>AdminTask.extractConfigProperties('-propertiesFileName  
    jdbcprovider.props -configData Server=server1 filterMechanism  
    SELECTED SUBTYPES -selectedSubTypes [JDBCProvider]')
```

```
ResourceType=JDBCProvider
ImplementingResourceType=JDBCProvider
ResourceId=Cell=!(cellName):Node=!(nodeName):Server
#
#Properties
#
classpath=$(DERBY_JDBC_DRIVER_PATH)/derby.jar
name=Derby JDBC Provider (XA)
implementationClassName=org.apache.derby.jdbc.EmbeddedDriver
nativepath={}
description=Built-in Derby JDBC Provider (XA)
providerType=Derby JDBC Provider (XA) #readonly
xa=true #boolean
```

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-25. Properties file based configuration: A solution

Using the `PropertiesBasedConfiguration` command group for the AdminTask object, you can extract the configuration attributes and values from your environment to properties files. You can use this feature for various purposes, including:

- To modify your existing configuration in one location, instead of configuring multiple administrative console panels or running many commands
 - To improve the application development lifecycle

Properties file-based configuration offers the following benefits:

- Properties files are more human readable than configuration files, which are in XML or other formats.
 - Properties files consist of a standard name-value pair format.
 - The properties file-based configuration decouples configuration data from changes in the underlying configuration model between releases.
 - Automating the configuration process is made easier.
 - The necessity to write complex wsadmin scripts is eliminated.

Many users of WebSphere Application Server try to create their own automated methods for modifying an application server configuration, often by directly manipulating files in the

configuration repository. However, custom solutions are tightly coupled to the underlying file structure, which can change from release to release.

- Properties file-based configuration can be used with, but is not a replacement for, configuration archives.

Properties files can be used with configuration archives to replicate a configuration by:

- Importing a configuration archive
- Using properties file configuration to customize the environment

At most, the properties files-based configuration commands extract commonly used configuration attributes from the configuration repository only. However, a backup that is made by using configuration archives can contain an exact copy of the configuration, which can be applied to another system to exactly replicate the configuration information.

If you want to replicate the configuration of one system onto another, with some customizations, you can use configuration archives to gain an exact replica of the configuration. Then, follow with properties file-based configuration to make the required customizations.

- Differences between configuration environments are easier to identify since properties files are simple text files. It is easy to use standard text editors to compare properties files from different configuration environments to help identify potential configuration problems.

WebSphere Application Server derives configuration information from the configuration repository, not from configuration properties files. To update the configuration repository to reflect the information in a configuration properties file, you must use wsadmin commands to apply the properties files to the configuration.

The graphic depicts the use of the `wsadmin extractConfigProperties` command to create a properties file with content based on information that is contained in the configuration XML file.

Properties file configuration content

- Each object is defined in a separate section:
 - Resource type and identifier
 - Configuration information
- Example: Properties for a JDBC provider

```
# SubSection 1.0 # JDBCProvider attributes
#
  ResourceType=JDBCProvider
ImplementingResourceType=JDBCProvider
ResourceId=Cell=!{cellName}:Node=!{nodeName}:Server=!
  {serverName}:JDBCProvider=ID#JDBCProvider_1183122153343
#
# Properties
#
  classpath={$DERBY_JDBC_DRIVER_PATH}/derby.jar} #Configuration information
name=Derby JDBC Provider
implementationClassName=
  org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource
nativepath={}
description=Derby embedded non-XA JDBC Provider
providerType=Derby JDBC Provider #readonly
xa=false #boolean
```

1

Resource type and identifier

2

Configuration information

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-26. Properties file configuration content

Configuration properties files contain a series of name-value pairs. Each configuration object is defined in two separate sections:

1. Resource type and identifier

The first section of the example defines a resource type and a resource identifier. The identifier is often in a format that includes the cell, node, and server names, ending with a string that contains the resource type and a large number.

In the example, the resource type is JDBC provider.

2. Configuration information

Name-value pairs are used to describe the configuration information.

The following code is an example of a JDBC provider configuration object:

```
#  
# SubSection 1.0 # JDBCProvider attributes  
#  
ResourceType=JDBCProvider  
ImplementingResourceType=JDBCProvider  
ResourceId=Cell=!{cellName}:Node=!{nodeName}:Server=!{serverName}:JDBCProvider=  
ID#JDBCProvider_1183122153343  
#  
#  
# Properties  
#  
classpath={$DERBY_JDBC_DRIVER_PATH}/derby.jar}  
name=Derby JDBC Provider  
implementationClassName=org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource  
nativepath={}  
description=Derby embedded non-XA JDBC Provider  
providerType=Derby JDBC Provider #readonly  
xa=false #boolean
```

Properties file configuration commands

- `extractConfigProperties`: Extracts configuration data into a properties file

```
wsadmin>AdminTask.extractConfigProperties('-propertiesFileName  
server1.props -configData Server=server1')
```

- `validateConfigProperties`: Verifies that the properties in the properties file are valid

```
wsadmin>AdminTask.validateConfigProperties('-propertiesFileName  
server1.props')
```

- `applyConfigProperties`: Applies properties in a specific properties file

```
wsadmin>AdminTask.applyConfigProperties('-propertiesFileName  
app.props')
```

- `deleteConfigProperties`: Deletes objects in your configuration

```
wsadmin>AdminTask.deleteConfigProperties('-propertiesFileName  
thread.props')
```

- `createPropertiesFileTemplates`: Creates template properties files

```
wsadmin>AdminTask.createPropertiesFileTemplates('-propertiesFileName  
app.props -configType Application')
```

[Introduction to wsadmin and scripting](#)

© Copyright IBM Corporation 2016

Figure 8-27. Properties file configuration commands

Properties file-based configuration uses the following five commands:

- `extractConfigProperties`

The `extractConfigProperties` command extracts configuration data in the form of a properties file. The system exports the most commonly used configuration data and attributes, converts the attributes to properties, and saves the data to a file. You can specify the resource of interest with the `target` object or the `configData` parameter. Use the `configData` parameter to specify a server, node, cluster, or application instance. If no configuration object is specified, the command extracts the profile configuration data. An example is:

```
AdminTask.extractConfigProperties(' -propertiesFileName server1.  
props -configData Server=server1')
```

One common scenario for using properties files to work with your configuration is to extract a properties file, which is based on your current environment. Modify the extracted file, and then apply the updated properties to your configuration. You can use any text editor to modify the properties file.

- validateConfigProperties

The `validateConfigProperties` command verifies that the properties in the properties file are valid and can be successfully applied to the new configuration. Here is an example:

```
AdminTask.validateConfigProperties( '-propertiesFileNameserver1.props' )
```

It is a good practice to validate the properties file before applying it to your configuration. The following two steps validate a properties file before applying it to the configuration:

- Use the `validateConfigProperties` command to validate the properties file.
- Use the `applyConfigProperties` command and the `-validate` option to apply the properties and validate the file simultaneously.

- applyConfigProperties

The `applyConfigProperties` command applies properties in a specific properties file to the configuration. If specific properties do not exist, the system adds attributes or configuration data to the configuration. If the properties exist in the configuration, the system sets the new values for the attributes. An example is:

```
wsadmin>AdminTask.applyConfigProperties( '-propertiesFileNameapp.props' )
```

- deleteConfigProperties

The `deleteConfigProperties` command deletes properties in your configuration as designated in a properties file. The system removes the attributes or configuration data that corresponds to each property in the properties file. In the properties file, you must add the flag `DELETE=true` to the section that contains the resource identifier for the object that you want to delete. Then, run the `deleteConfigProperties` command on your properties file. An example is:

- In the properties file:

```
#  
# SubSection 1.0.1.4 # Thread pools  
#  
ResourceType=ThreadPool  
ImplementingResourceType=Server  
ResourceId=Cell!=!{cellName}:Node!=!{nodeName}:Server!=!  
{serverName}:ThreadPoolManager=  
ID#ThreadPoolManager_1:ThreadPool-myThreadPool  
DELETE=true  
#
```

- The command to delete the thread pool would be:

```
wsadmin>AdminTask.deleteConfigProperties  
( '-propertiesFileNamethread.props' )
```

If you run the `deleteConfigProperties` command before you add the `DELETE=true` attribute and value to the properties file, the command resets each property to the default value. The system completely removes properties that do not have default values.

- createPropertiesFileTemplates

Use the `createPropertiesFileTemplates` command to create template properties files to

use for creating or deleting specific object types. The command stores the template properties file in the properties file that the `propertiesFileName` parameter identifies. An example is:

```
wsadmin>AdminTask.createPropertiesFileTemplates(  
' -propertiesFileNameapp.props -configType Application')
```

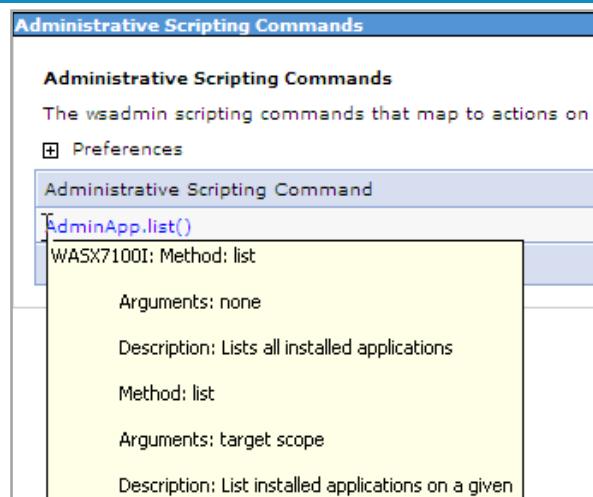
There are four different types of properties file templates that you can create with the `createPropertiesFileTemplates` command:

- Server
- ServerCluster
- Application
- AuthorizationGroup

The templates are properties files that contain the required parameters to create a configuration object of a specific type. When you create a template for the application configuration type, you must provide a resource ID and information about the application that you are deploying. The properties file template contains comments and instructions for how to modify and use the template.

Command assistance

- Works in concert with the administrative console
 - Last run commands are made available to Rational Application Developer
 - Commands can be pasted directly to Jython scripts



- Administrative console access
 - Under **Help**, click **View administrative scripting command for last action**
 - The last command that ran is displayed
 - Place the cursor over the command to get command information
 - Command can be copied into a Jython script



Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-28. Command assistance

Using command assistance, you can see wsadmin scripting commands that correspond to actions in the administrative console. Seeing these commands might help you develop the commands necessary to administer WebSphere Application Server from the wsadmin utility. You can view wsadmin scripting commands in the Jython language for the last action that runs in the administrative console. Working in concert with the administrative console, administrative command assistance provides the following capabilities:

- The last commands are made available to Rational Application Developer. When you do server operations in the administrative console, the administrative command assistance tool captures and displays the wsadmin commands that are issued.
- Commands can be pasted directly to Jython scripts. You can transfer the output from the administrative command view directly to a text editor, such as the Jython editor, enabling you to develop Jython scripts that are based on actual console actions.

To access command assistance from the administrative console:

1. Under **Help**, click **View administrative scripting command for last action**. The last command is shown.
2. Place the cursor over the command to get command information.
3. The command can be copied into a Jython script.

Unit summary

- Use wsadmin to enter administrative commands
- Create Jython scripts to run wsadmin commands
- Customize the wsadmin environment with profiles and property files
- Use property file-based configurations to modify an environment
- Configure and use command assist to develop wsadmin scripts

Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-29. Unit summary

Review questions

1. Which of the following list is not one of the five Java objects that perform different operations?
 - A. AdminConfig
 - B. AdminControl
 - C. AdminTask
 - D. Help
 - E. AdminStart

2. What is the default protocol type for wsadmin?
 - A. SOAP
 - B. RMI
 - C. None

3. The default behaviors for wsadmin can be changed by editing which file?



Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-30. Review questions

Write your answers here:

- 1.

- 2.

- 3.

Review answers

1. Which of the following list is not one of the five Java objects that perform different operations?

- A. AdminConfig
- B. AdminControl
- C. AdminTask
- D. Help
- E. AdminStart

The answer is E.

2. What is the default protocol type for wsadmin?

- A. SOAP
- B. RMI
- C. None

The answer is A.

3. The default behaviors for wsadmin can be changed by editing which file?

The answer is: wsadmin.properties



Introduction to wsadmin and scripting

© Copyright IBM Corporation 2016

Figure 8-31. Review answers

Exercise: Using wsadmin

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 8-32. Exercise: Using wsadmin

Exercise objectives

After completing this exercise, you should be able to:

- Use wsadmin to run administrative commands interactively and with scripts
- Create a simple administrative script
- Use console command assistance
- Use property file based configuration to modify your settings

Unit 9. WebSphere security

Estimated time

01:30

Overview

In this unit, you learn basic security concepts and architecture that apply to WebSphere Application Server. You learn how to configure administrative security and application security.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server V9 documentation in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Explain basic security concepts
- Describe WebSphere Application Server security
- Describe enhancements to certificate management
- Configure fine-grained administrative security
- Configure application security
- Describe auditing features and functions

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-1. Unit objectives

Topics

- WebSphere security basics
- WebSphere user registries
- Administrative security
- Application security
- Java 2 security
- Security auditing

[WebSphere security](#)

© Copyright IBM Corporation 2016

Figure 9-2. Topics

9.1. WebSphere security basics

WebSphere security basics

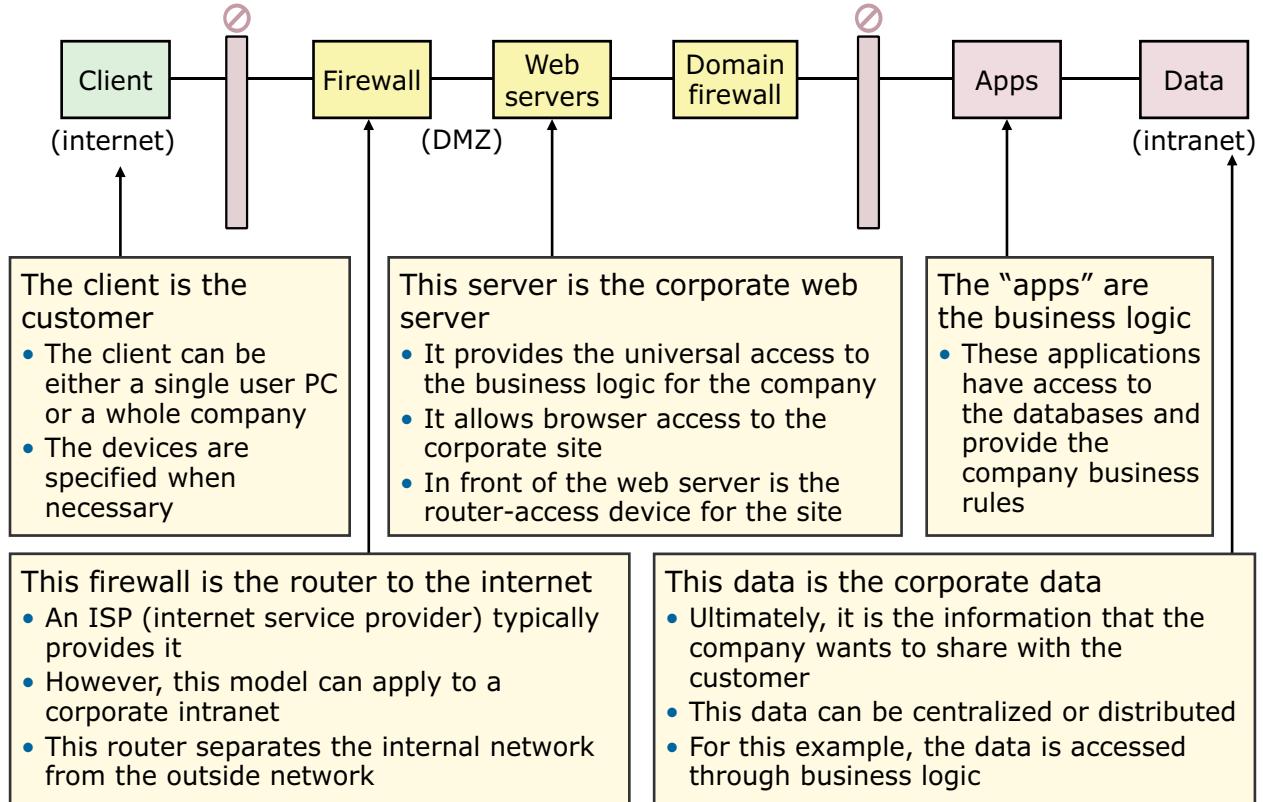
© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-3. WebSphere security basics



Basic security end-to-end model



WebSphere security

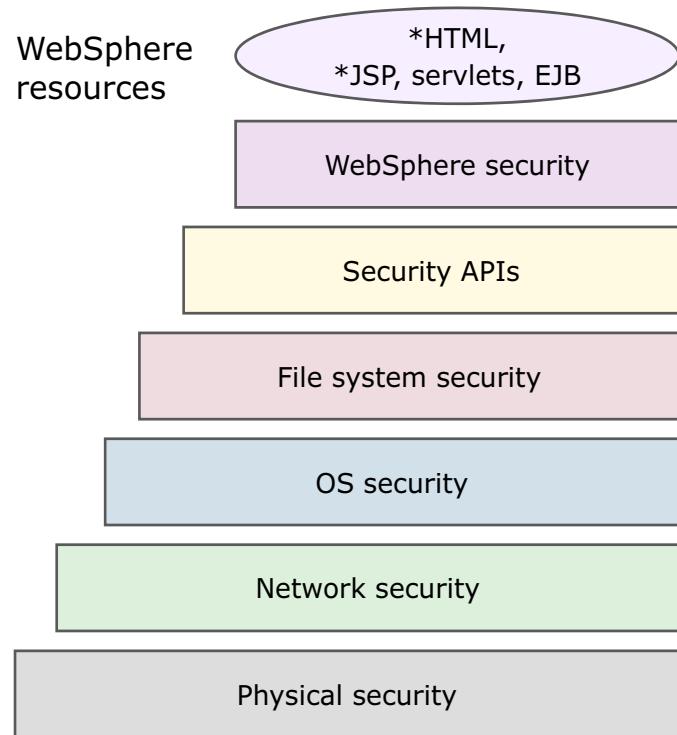
© Copyright IBM Corporation 2016

Figure 9-4. Basic security end-to-end model

With an e-business application, a general topology must be secured from end to end. This process involves securing many different parts.

WebSphere Application Server security overview

- Security can be applied at different levels



[WebSphere security](#)

© Copyright IBM Corporation 2016

Figure 9-5. WebSphere Application Server security overview

Many levels are involved in securing an environment. WebSphere provides only part of the total security that must be applied.

Physical security refers to the protection of the hardware itself. Is it kept in a safe and secure area? Who has physical access to it?

Network security can involve setting up firewalls to protect an intranet or a DMZ where the web servers are going to run.

Operating system security is the security infrastructure of the underlying operating system. It provides certain security services to the WebSphere security application. These services include the file system security support to secure sensitive files in WebSphere product installation. The WebSphere system administrator can configure the product to obtain authentication information directly from the operating system user registry, for example, the NT Security Access Manager (SAM).

File system security is especially a concern about protecting your configuration files and key ring files.

Security APIs are as follows:

- JVM: The JVM security model provides a layer of security above the operating system layer.

- CORBA security: Any calls that are made among secure ORBs are over a Secure Association Service (SAS) or CSIV2 layer that sets up the security context and the necessary quality of protection. After the session is established, the call is passed up to the enterprise bean layer. This layer is for a **distributed** platform only.
- Java EE security: The security collaborator enforces Java EE-based security policies and supports Java EE security APIs.
- WebSphere security: WebSphere security enforces security policies and services in a unified manner on access to web resources and enterprise beans. It consists of WebSphere security technologies and features to support the needs of a secure enterprise environment.

WebSphere security service: Wide view

- Security service runs locally in each process (deployment manager, node agent, and application server)
 - Security workload not bottlenecked to a single process
 - Security service failure affects only a single process
- Separation of authentication mechanism and user registry

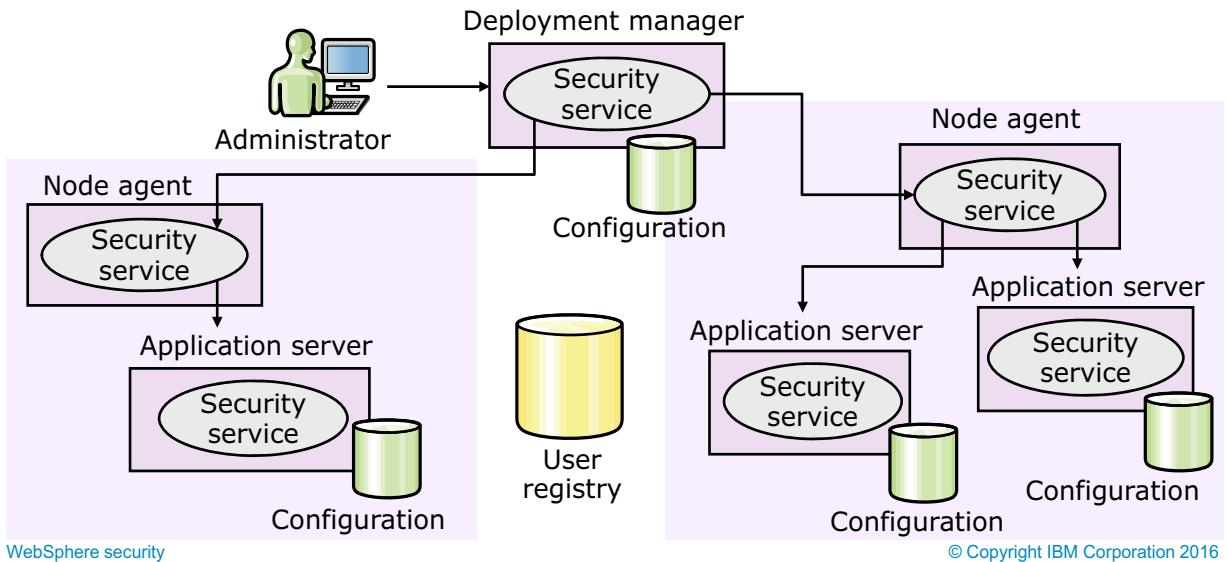


Figure 9-6. WebSphere security service: Wide view

The security service runs within each of the managed processes so that it does not have a single point of failure (other than possibly the user registry, but that is a different topic).

Types of security

- Administrative security
 - Protects things such as administrative console, wsadmin, scripts
- Application security
 - Protects access to the applications
- Java 2 security
 - Protects the local systems



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-7. Types of security

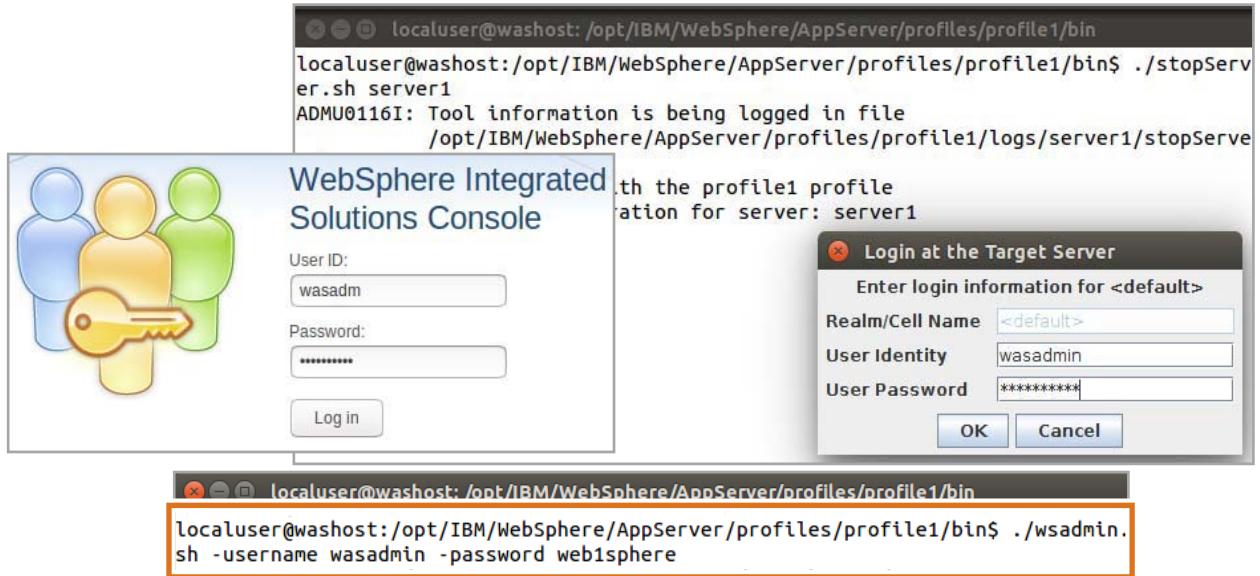
Within WebSphere, a number of different types of security can be configured. These types of security are covered in more detail during this lecture. They include:

- Administrative security
- Application security
- Java 2 security



Administrative security

- Protects administrative console, scripts, wsadmin, and others
- Access can be restricted through:
 - Administrative roles
 - Fine-grained access



WebSphere security

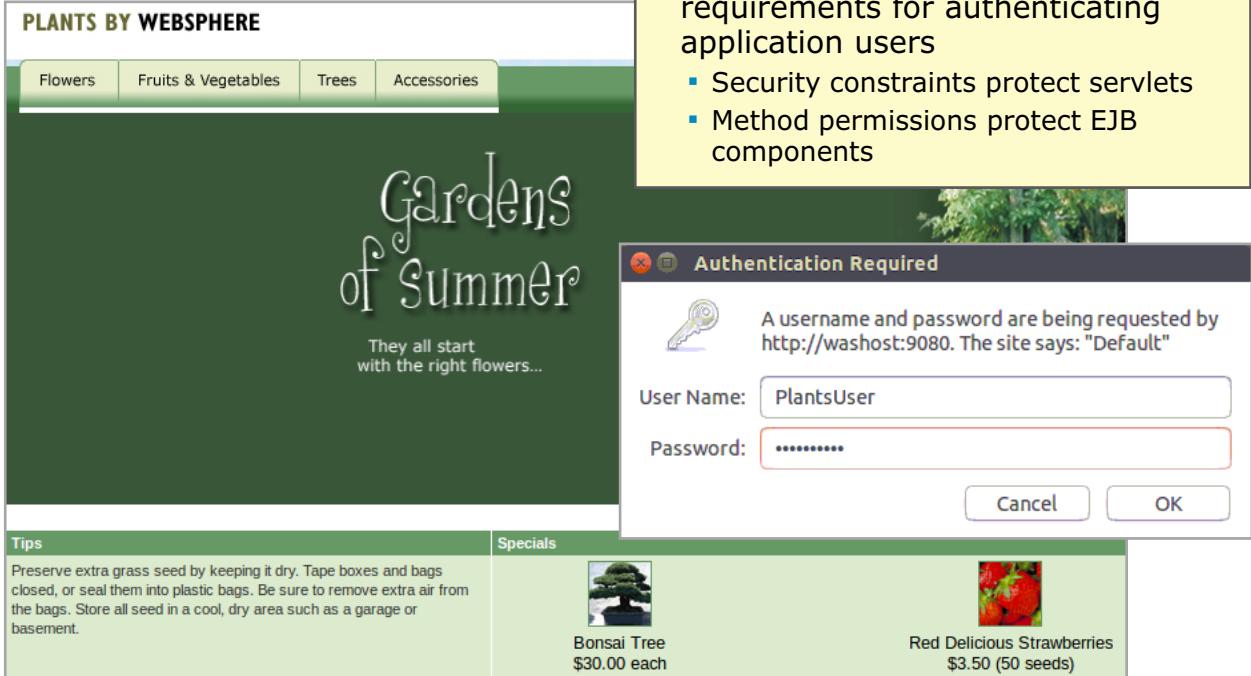
© Copyright IBM Corporation 2016

Figure 9-8. Administrative security

Administrative security allows the administrator to restrict access to the administrative interfaces, including the administrative console, the administrative scripts, and wsadmin.

IBM Training 

Application security



- Enables security for the applications in your environment
- Provides application isolation and requirements for authenticating application users
 - Security constraints protect servlets
 - Method permissions protect EJB components

 Authentication Required

A username and password are being requested by <http://washost:9080>. The site says: "Default"

User Name:

Password:

[WebSphere security](#)

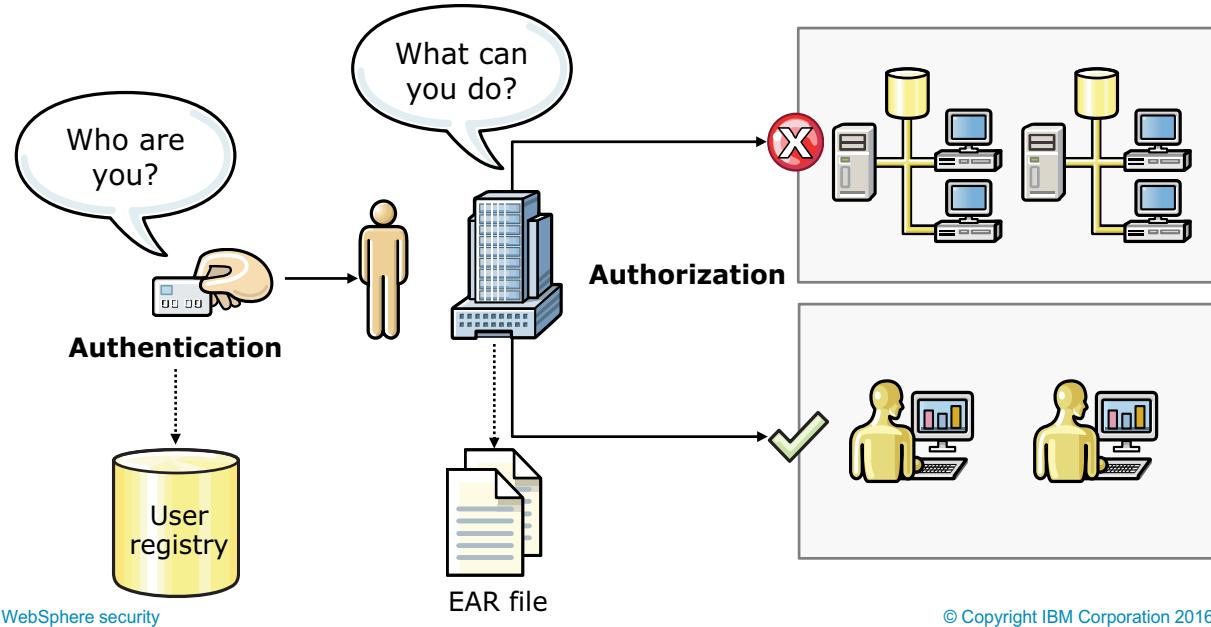
© Copyright IBM Corporation 2016

Figure 9-9. Application security

Application security allows the administrator to restrict access to the enterprise applications. This restricted access is done by defining security roles, security constraints, and method permissions. Then, the security roles are mapped to the users and groups in the environment.

Authentication and authorization: What is the difference?

- The distinction between authentication and authorization is important
 - Authentication → Who are you?
 - Authorization → When authenticated, what are you allowed to do?



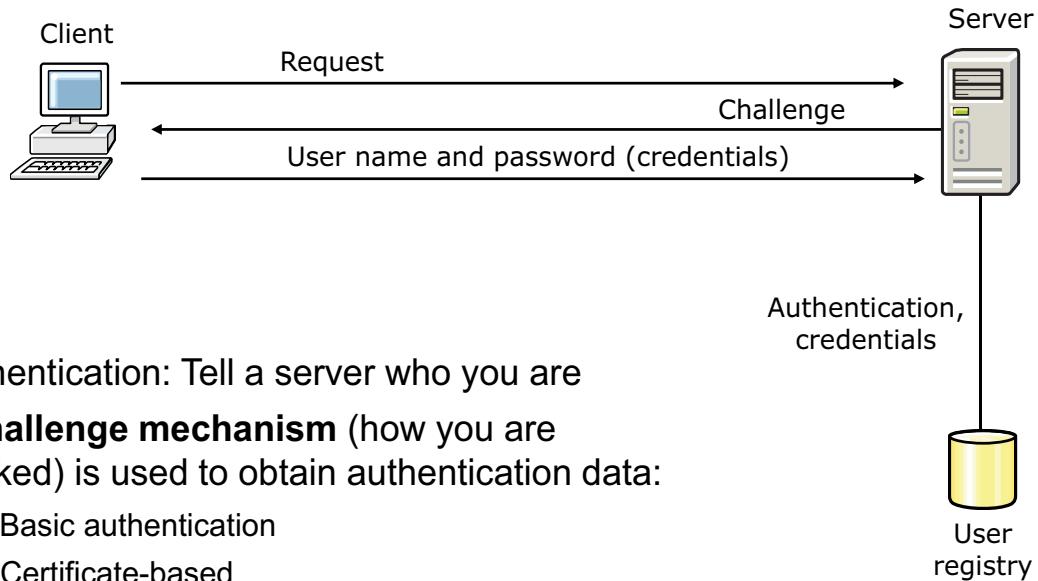
WebSphere security

© Copyright IBM Corporation 2016

Figure 9-10. Authentication and authorization: What is the difference?

Authentication information can be found in a user registry. Authorization information can be found within the EAR file. The WebSphere security service is responsible for making sure that protected resources are accessible only by authenticated and correctly authorized users.

Challenge mechanism: Authentication basic steps



Authentication: Tell a server who you are

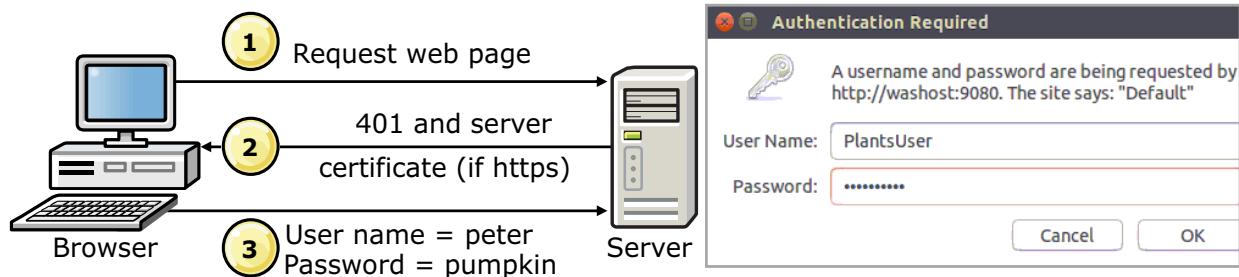
- **Challenge mechanism** (how you are asked) is used to obtain authentication data:
 - Basic authentication
 - Certificate-based
 - Form-based authentication (preferred)
- Challenge mechanism is defined in the deployment descriptor
- Credentials are validated against user registry

Figure 9-11. Challenge mechanism: Authentication basic steps

Authentication in its simplest form is rather straightforward.

- A request is made.
- A challenge is returned.
- A user ID and password are sent.
- The server checks a user registry to see whether the information is valid.

Challenge mechanism: Basic authentication



- Warning: Password is not encrypted, merely encoded
 - Make sure that this channel is over HTTPS
- Warning: Basic authentication token is sent across in the HTTP header
 - Danger: The server has no way to remove the token during a logout
 - If you walk away from a public browser, even if you log off, your credentials are still stored in the browser
 - Another person can walk up and be automatically logged in to your site
 - To remove the basic authentication token, close the browser or explicitly tell the browser to delete authentication credentials

WebSphere security

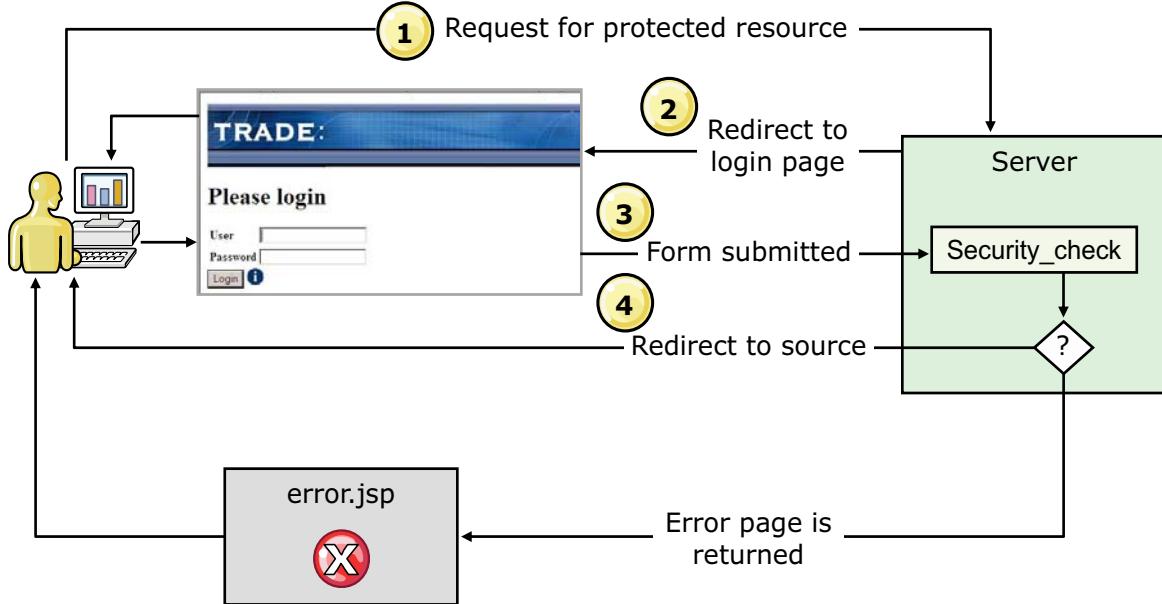
© Copyright IBM Corporation 2016

Figure 9-12. Challenge mechanism: Basic authentication

HTTP basic authentication is not a secure authentication mechanism. Basic authentication sends user names and passwords over the internet as text that is Base64 encoded, and the target server is not authenticated. This form of authentication can expose user names and passwords. If someone can intercept the transmission, the user name and password information can easily be decoded. However, when a secure transport mechanism such as SSL, or security at the network level such as the IPsec protocol or VPN strategies, is used with basic authentication, some of these concerns can be alleviated.

Challenge mechanism: Form-based authentication

- Defined within the application
- Suggested approach



WebSphere security

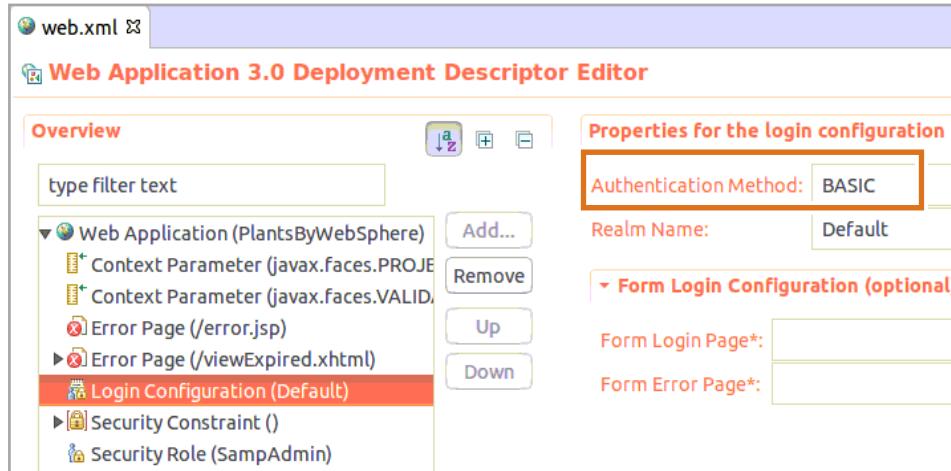
© Copyright IBM Corporation 2016

Figure 9-13. Challenge mechanism: Form-based authentication

- A client requests access to a protected resource.
- If the client is unauthenticated, the server redirects the client to a login page.
- The client submits the login form to the server.
- The server attempts to authenticate the user. If authentication succeeds, the authenticated user's principal is checked to ensure that it is in a role that is authorized to access the resource. If the user is authorized, the server redirects the client to the resource that uses the stored URL path. If authentication fails, the client is forwarded or redirected to an error page.

Defining the challenge type

- Challenge types are set in the EAR deployment descriptors
 - The default that Rational Application Developer uses is basic authentication
 - Form-based is defined by adding a login configuration



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-14. Defining the challenge type

The challenge type is defined within the EAR. Using Rational Application Developer, it can be defined within the deployment descriptor. The default type is basic authentication.

9.2. WebSphere user registries

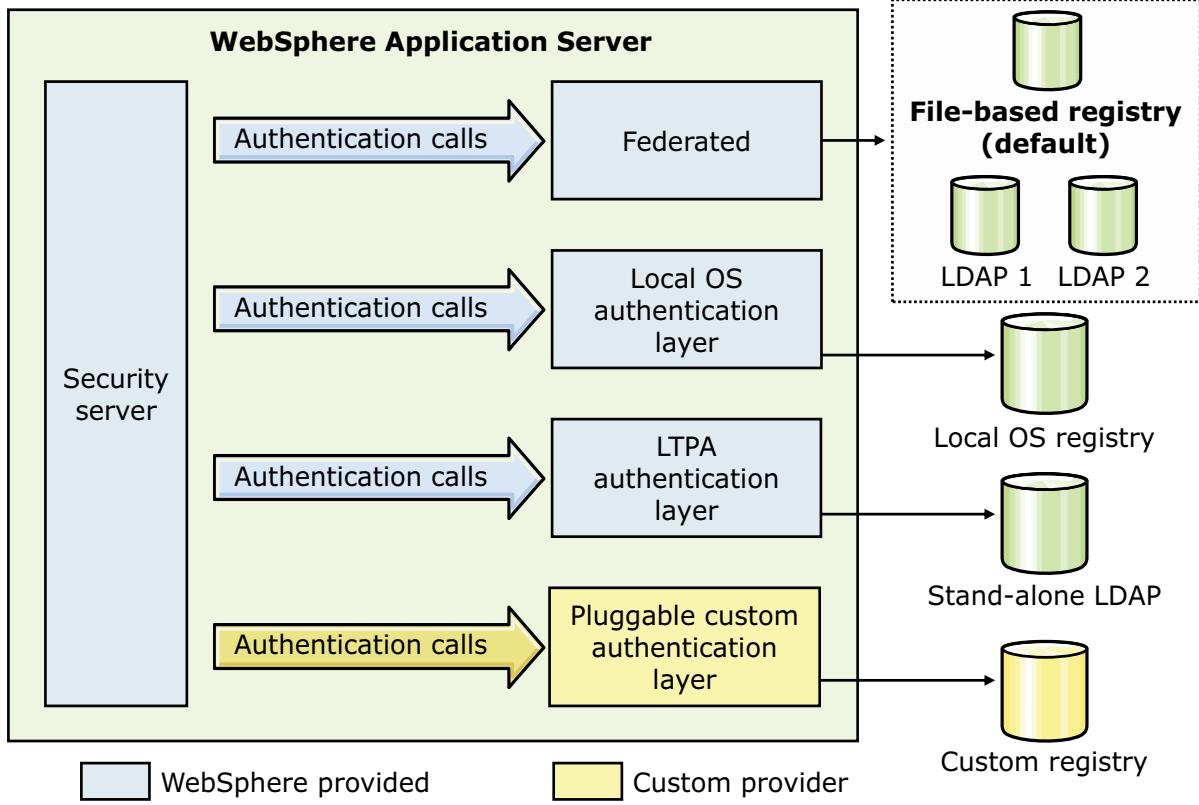
WebSphere user registries

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-15. WebSphere user registries

Registries and authentication mechanisms



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-16. Registries and authentication mechanisms

The user registries that are supported include local operating system, stand-alone LDAP, custom user registry, and federated registry. The federated registry effectively combines multiple repositories into a single view. It can support multiple LDAP servers, file-based repository, database repository, and custom repositories.

Global security

Use this panel to configure administration and the default application security policy. This se security policy for user applications. Security domains can be defined to override and custo

[Security Configuration Wizard](#) [Security Configuration Report](#)

Administrative security

Enable administrative security [Administrative authentication](#)

Application security

Enable application security

Java 2 security

Use Java 2 security to restrict application access to local resources
 Warn if applications are granted custom permissions
 Restrict access to resource authentication data

User account repository

Realm name: defaultWIMFileBasedRealm

Current realm definition: Federated repositories

Available realm definitions:

- Federated repositories
- Federated repositories**
- Local operating system
- Standalone LDAP registry
- Standalone custom registry

Configure... Set as current

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-17. Defining user registries

Defining which user registry is used and how it is configured can be done through the console, through a manual interface, or through a wizard. Generally speaking, the wizard is considered too simplistic to be used for configuring anything but the simplest configurations.



Manual security configuration

User account repository

Realm name
defaultWIMFileBasedRealm

Current realm definition
Federated repositories

Available realm definitions

- Federated repositories
- Federated repositories** (highlighted)
- Local operating system
- Standalone LDAP registry
- Standalone custom registry

Configure...

Global security > Federated repositories

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. This can consist of identities in the file-based repository that is built into the system, in one or more external repositories, both the built-in repository and one or more external repositories.

General Properties

* Realm name
defaultWIMFileBasedRealm

* Primary administrative user name

Server user identity

Automatically generated server identity

Server identity that is stored in the repository
Server user ID or administrative user on a Version 6.0.x node

Ignore case for authorization

Allow operations if some of the repositories are down

Repositories in the realm:

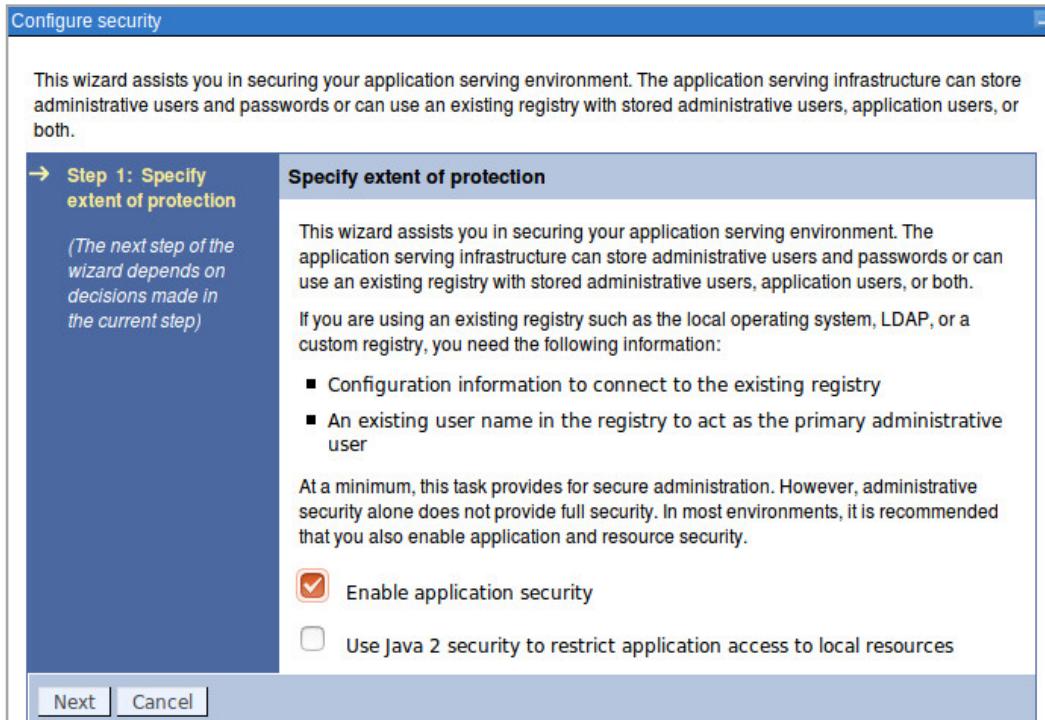
Add repositories (LDAP, custom, etc)...	Use built-in repository	Remove	
Select	Base Entry	Repository Identifier	Repository Type
<small>You can administer the following resources:</small>			
<input type="checkbox"/>	o=defaultWIMFileBasedRealm	InternalFileRepository	File

WebSphere security

Figure 9-18. Manual security configuration

Security wizard: Step 1

- Not detailed enough to be used for anything but simple environments



WebSphere security

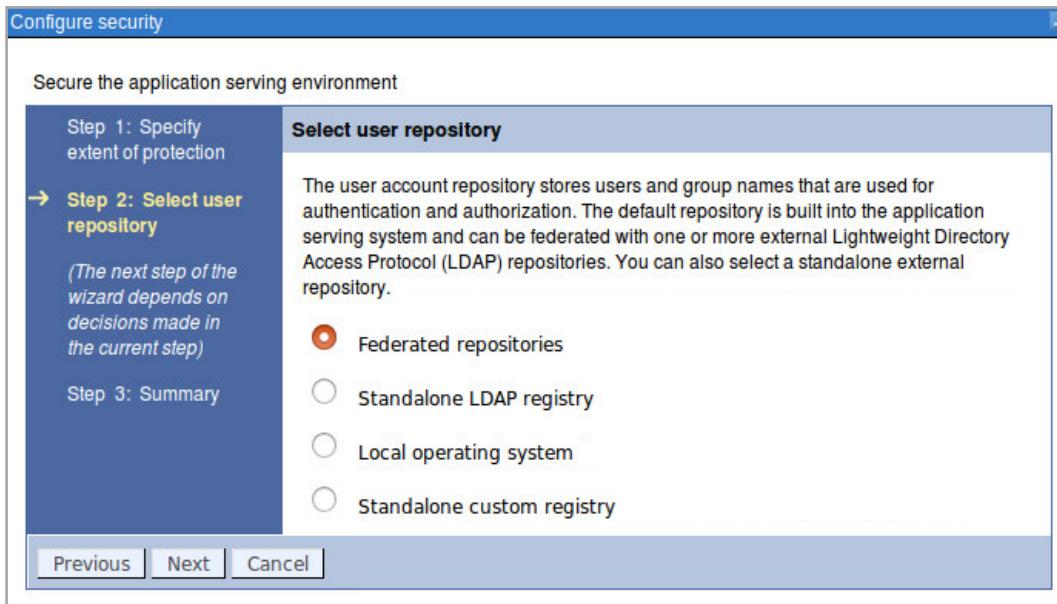
© Copyright IBM Corporation 2016

Figure 9-19. Security wizard: Step 1

Although the wizard that is used to configure security can set up simple environments, it is typically used for only the most basic configurations.

This diagram shows step 1 of the security wizard.

Security wizard: Step 2



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-20. Security wizard: Step 2

This diagram shows step 2 of the security wizard. The user repository is being selected.

Security wizard: Step 3

Configure security

Secure the application serving environment

Step 1: Specify extent of protection Step 2: Select user repository → Step 3: Configure federated repository Step 4: Summary	Configure federated repository <p>A secure, file-based user repository is built into the system for storing administrative users or environments with a small number of users. The file-based user repository can be federated with one or more external LDAP repositories. If this is the first time security has been enabled using this repository, provide a new user name and password to act as an administrator. If security was previously enabled using this repository, provide the name of a user with administrator privileges that is in the built-in repository.</p> <p>Note: Use this panel to configure a federated repository with a built-in, file-based repository in the realm. To configure a federated repository with a non file-based repository in the realm, you must use the User accounts repository section on the Global security panel.</p> <p>* Primary administrative user name <input type="text" value="wasadmin"/></p> <p>Password <input type="password" value="*****"/></p> <p>Confirm password <input type="password" value="*****"/></p>
--	---

[Previous](#) [Next](#) [Cancel](#)

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-21. Security wizard: Step 3

This diagram shows step 3 of the security wizard and specifies the user name and password for the primary administrative user.



Security wizard: Step 4

Configure security

Secure the application serving environment

Step 1: Specify extent of protection	Summary	
Step 2: Select user repository	Displays the list of values that are selected during the wizard and are used to enable security.	
Step 3: Configure federated repository		
→ Step 4: Summary		
Options	Values	
Enable administrative security	true	
Enable application security	true	
Use Java 2 security to restrict application access to local resources	false	
User repository	Federated repositories	
Primary administrative user name	wasadmin	

Previous | Finish | Cancel

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-22. Security wizard: Step 4

This diagram shows the summary of the security wizard.

User registry support

- WebSphere Application Server supports some user registries

Local OS	LDAP
NT Domain, NT WorkGroup, Windows	IBM Tivoli Directory Server
AIX	IBM SecureWay Directory Server
Solaris	Sun Java System Directory Server
HP-UX	IBM Lotus Domino
Linux	Microsoft Active Directory
OS/400	Novell eDirectory
	Custom (requires addition configuration)

[WebSphere security](#)

© Copyright IBM Corporation 2016

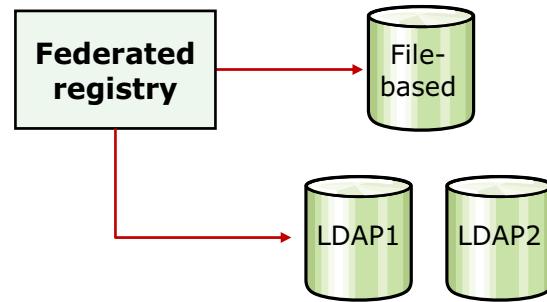
Figure 9-23. User registry support

Although there are supported LDAP servers, other LDAP servers can also be used by defining the appropriate schema mappings through the advanced LDAP properties.

Generally, avoid local OS registries, particularly in distributed, non-domain environments.

Federated repositories

- The installation wizard and profile management tools have a default of enabling administrative security
 - The default repository type is a file-based federated repository
- Federated repositories provide for the use of multiple repositories with WebSphere Application Server
- Can be:
 - File-based
 - Single LDAP
 - Custom registry
 - Database
 - Multiple LDAPS
 - Subtree of an LDAP
- Defined and theoretically combined under a single realm
- All of the user repositories that are configured under the federated repository are invisible to WebSphere Application Server



[WebSphere security](#)

© Copyright IBM Corporation 2016

Figure 9-24. Federated repositories

This slide covers the federated repositories and what can be used:

- File-based
- Single LDAP
- Custom registry
- Database
- Multiple LDAPS
- Subtree of an LDAP

The VMM (Virtual Member Manager) provides federation capabilities.



Custom registry: Configuration

[Global security > Standalone custom registry](#)

Specifies a custom registry that implements the UserRegistry interface in the com.ibm.websphere.security package. For backward compatibility, the application server also supports a custom registry that implements the CustomRegistry interface in the com.ibm.websphere.security package. When security is enabled and any of the properties or panel are changed, go to the Security > Global security panel. Click Apply or OK to validate the changes.

General Properties		Related Items						
* Primary administrative user name <input type="text"/>		Trusted authentication realms - inbound						
Server user identity <ul style="list-style-type: none"> <input checked="" type="radio"/> Automatically generated server identity <input type="radio"/> Server identity that is stored in the repository Server user ID or administrative user on a Version 6.0.x node <input type="text"/> <p>Password <input type="password"/></p>								
* Custom registry class name <input type="text" value="com.ibm.websphere.security.FileRegistrySample"/>		Configured from administrative console: <ul style="list-style-type: none"> • Security > Global security • Select Standalone custom registry from Available realm definitions • Click Configure <ul style="list-style-type: none"> ▪ User name and password must exist ▪ Class name must be implemented and in class path 						
Custom properties <input type="button" value="New"/> <input type="button" value="Delete"/> <table border="1"> <thead> <tr> <th>Select</th> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> </tbody> </table>			Select	Name	Value	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
Select	Name	Value						
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>						

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-25. Custom registry: Configuration

The custom registry allows custom implementation of user registry.

Some possible implementations include:

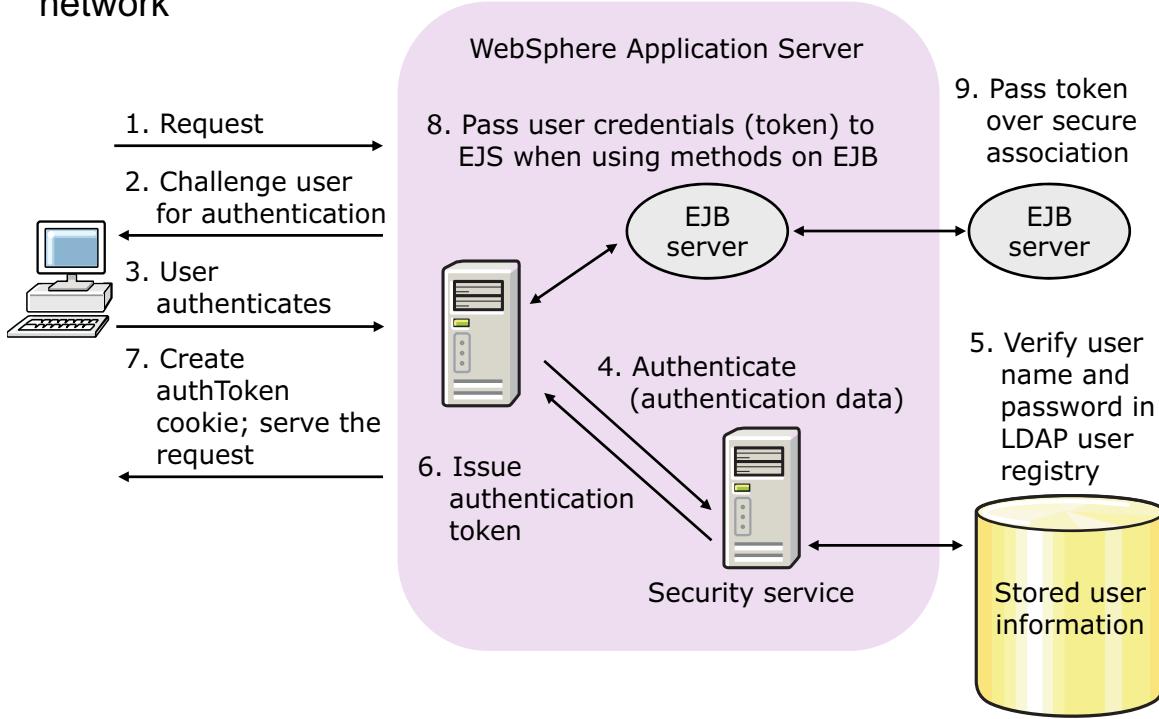
- Database
- Flat file
- OS-based, with more custom logic
- Use other, not directly supported, registries

WebSphere provides:

- Base types
 - Implementing classes extend the `com.ibm.websphere.security.UserRegistry` class
- Working sample implementation: `com.ibm.websphere.security.FileRegistrySample`

Authentication mechanism: LTPA

- Allows the identity of a user to be passed around the distributed network



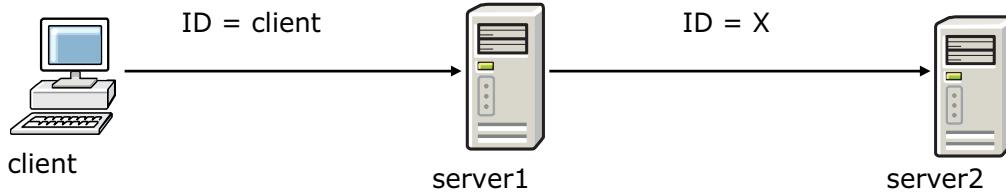
WebSphere security

© Copyright IBM Corporation 2016

Figure 9-26. Authentication mechanism: LTPA

Regardless of what registry type is used, an LTPA token is generated and then shared with the client in the form of a cookie. The token, or the information in the token, can be used throughout WebSphere to pass the user's identity information around.

LTPA provides delegation



X can run as:

Option 1. Client

Option 2. Server 1

Option 3. Specified identity

- Using an LTPA token supports delegation

A request for work can pass its security context, which contains its credentials. As the call proceeds through the servers to its final destination, credentials can be changed. The options are to keep the client's credentials, switch to the credentials of the server, or some other specified identity.

LTPA provides single sign-on (SSO)

Global security > Single sign-on (SSO)

Specifies the configuration values for single sign-on.

General Properties

- Enabled
- Requires SSL
- Domain name
- Interoperability mode
-
- Web inbound security attribute propagation
- Set security cookies to HTTPOnly to help prevent cross-site scripting attacks

Apply OK Reset Cancel

- As soon as clients have a valid LTPA token, they do not need to reauthenticate within a cell (until the LTPA token expires)
- SSO is on by default
- Issues cookies to web browser to track user authentication information
- Provides for SSO within or even between WebSphere cells

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-28. LTPA provides single sign-on (SSO)

It is also possible to configure multiple cells to share LTPA tokens, thus creating SSO for multiple cells.

9.3. Administrative security

Administrative security

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-29. Administrative security

Administrative security

Turning on administrative security enables many features, including:

- Authentication of HTTP and IIOP clients
- Administrative console security
- Naming security
- Use of SSL transports
- Role-based authorization checks of servlets, EJB components, and MBeans
- Propagation of identities (RunAs)
- The common user registry



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-30. Administrative security

Administrative security not only protects the administrative tools, but also enables a number of other security features:

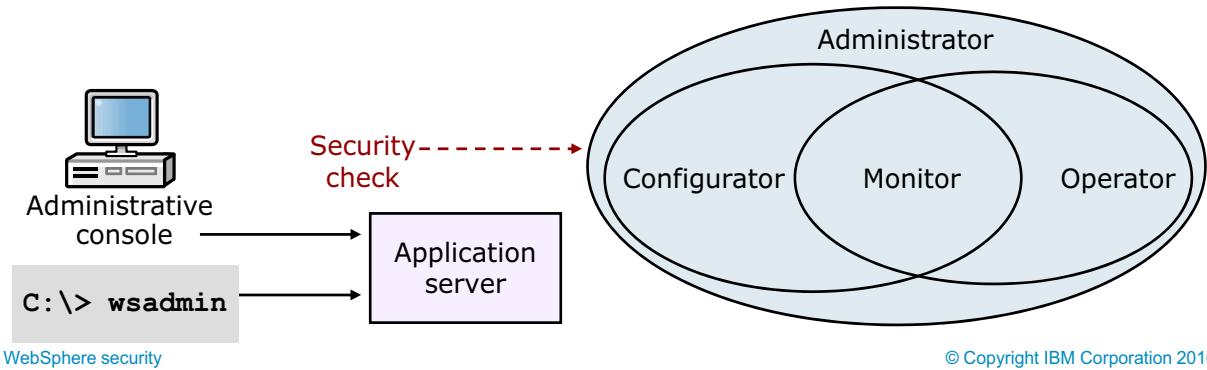
- Authentication of HTTP and IIOP clients
- Administrative console security
- Naming security
- Use of SSL transports
- Role-based authorization checks of servlets, EJBs, and MBeans
- Propagation of identities (RunAs)
- The common user registry

Whichever user ID is used to run the application server process has implicit access as a console administrator user.

Console security

Defines which roles have access to the administrative tools

- **Monitor:** Least privileged, allows a user to view the WebSphere configuration and current application server state
- **Configurator:** Monitor privilege plus the ability to change the WebSphere configuration
- **Operator:** Monitor privilege plus the ability to change runtime state, such as starting or stopping servers
- **Administrator:** Operator, configurator, and iscadmins privilege, plus more privileges that are granted solely to the administrator role



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-31. Console security

Additional privileges for the administrator role include:

- Modifying the primary administrative user and password
- Mapping users and groups to the administrator role
- Enabling or disabling administrative and Java 2 security

More console security roles

- **iscadmins** (Integrated Solutions Console)
 - Available only for administration console users
 - Allows a user to manage users and groups in the federated repositories
- **Deployer**
 - Allows a user to change configuration and runtime state on applications that use wsadmin
- **Admin Security Manager**
 - Allows a user to map users to administrative roles by using wsadmin
 - When restricted access to resource authentication data is in effect, users can also manage authorization groups
- And others

WebSphere security

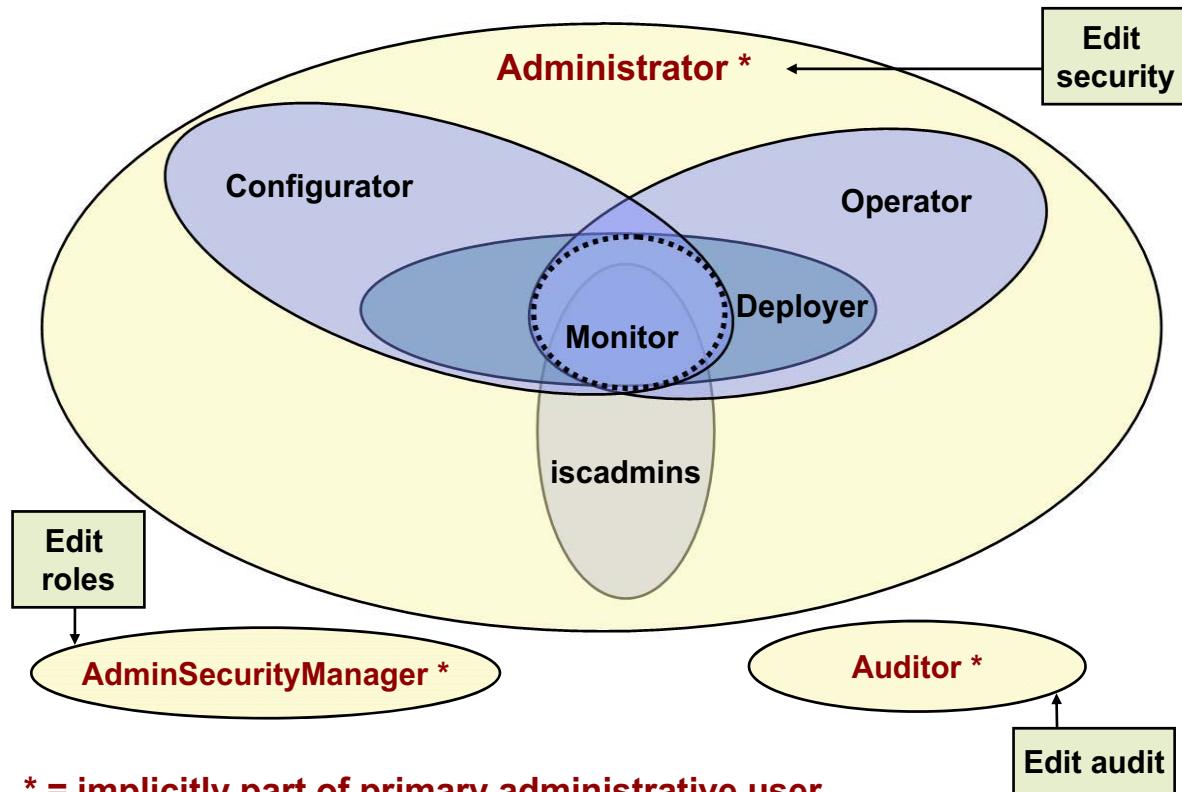
© Copyright IBM Corporation 2016

Figure 9-32. More console security roles

There are more console security roles available; these include iscadmins, deployer, and AdminSecurityManager.



Administrative roles



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-33. Administrative roles

This graphic shows a representation of the various administrative security roles that are available, and how they overlap.

Console security: Creating users and groups

- To set up console security
 - Turn on administrative security
 - Create console users and groups**
 - Done in active user registry



The diagram shows two screenshots of the administrative console. The top screenshot is titled 'Manage Users' and shows a search interface with fields for 'Search by' (User ID), 'Search for' (*), and 'Maximum results' (100). Below the search bar is a message: '0 users matched the search criteria.' A 'Create...' button is highlighted with an orange box. The bottom screenshot is titled 'Manage Groups' and shows a similar search interface with fields for 'Search by' (Group name), 'Search for' (*), and 'Maximum results' (100). Below the search bar is a message: '0 groups matched the search criteria.' A 'Create...' button is also highlighted with an orange box. Both screens show a table with columns for User ID, Group name, First name, Last name, and Email address, though no data is present.

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-34. Console security: Creating users and groups

This diagram shows the creation of users and groups through the administrative console.

Console security: Mapping users and groups

- To set up console security
 - Turn on administrative security
 - Create console users and groups
 - **Map users and groups to administrative roles**



WebSphere security

The screenshot shows the 'Administrative user roles' configuration page. At the top, it says 'Administrative user roles > User'. Below that is a list of roles: Admin Security Manager, Administrator, Auditor, and Configurator. A 'Search and Select Users' section follows, with a 'Search string' input field containing an asterisk (*) and a 'Search' button. Below this are fields for 'Maximum results to display' (set to 20) and two lists: 'Available' and 'Mapped to role'. The 'Available' list contains two items, each with a blue circular icon with a black arrow pointing right. The 'Mapped to role' list also contains two items, each with a blue circular icon with a black arrow pointing right.

© Copyright IBM Corporation 2016

Figure 9-35. Console security: Mapping users and groups

This diagram shows the mapping of users to specific console security roles. The interface for mapping administrative groups is virtually the same.

9.4. Application security

Application security

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

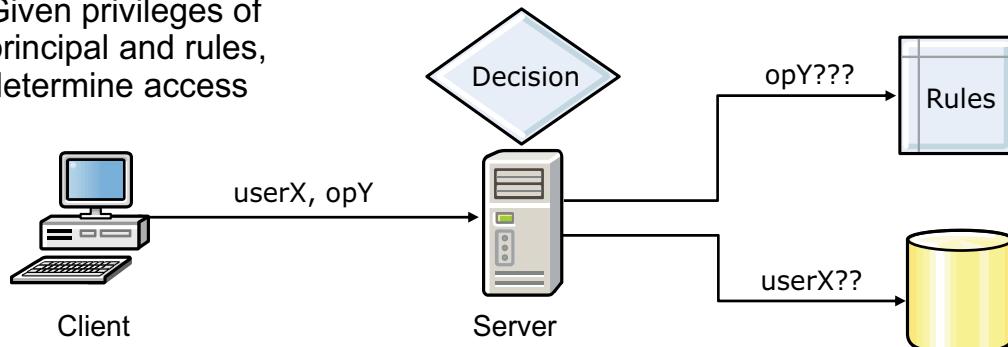
Figure 9-36. Application security

Authorization

Authorization involves granting trusted principals permission to perform actions on resources (web pages, servlets, JSP pages, and EJB components)

Control access to resources

- Security lookup (by server)
 - Determine security privileges for principal
 - Access information that is stored in registry
- Rule enforcement (by server)
 - Obtain rules from registry
 - Given privileges of principal and rules, determine access



WebSphere security

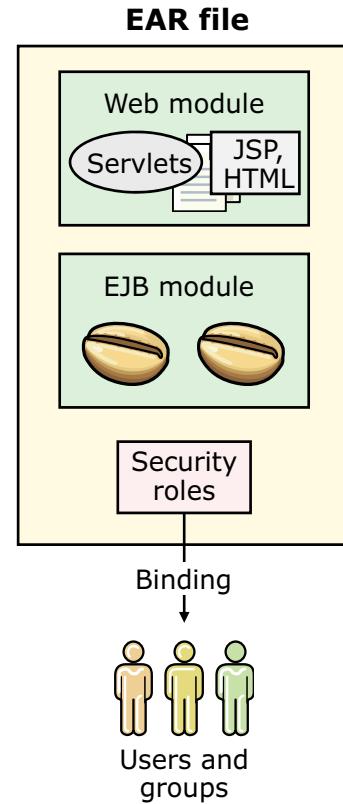
© Copyright IBM Corporation 2016

Figure 9-37. Authorization

Java EE security can provide the rules engine through the WebSphere Application Server. Alternatively, another authorization service (such as Tivoli Access Manager) can be used.

Security roles: Application authorization

- Use security roles for authorization
 - Specify security at an abstract level without knowledge of actual users and groups
- Security roles are then applied to the web and EJB application components
 - Web URLs or EJB methods
- Binding of the users and groups to the security roles is generally done at the application installation time
 - Can also be done post-installation



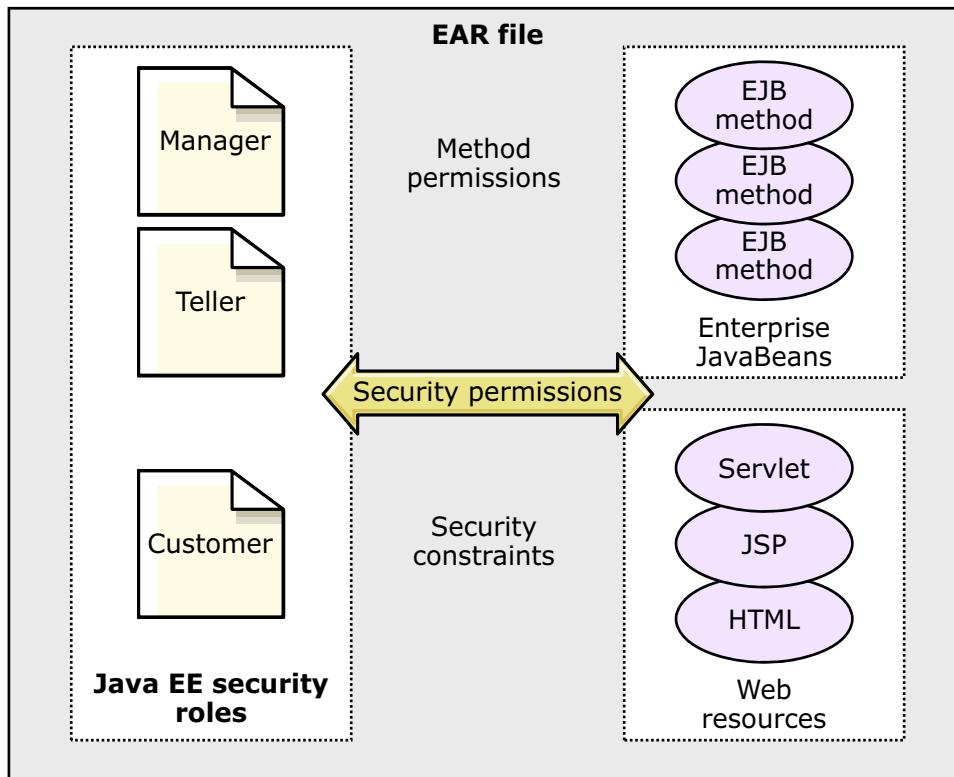
WebSphere security

© Copyright IBM Corporation 2016

Figure 9-38. Security roles: Application authorization

Java EE security is concerned with controlling access to application resources, not system resources.

Securing Java EE application artifacts: Part 1



WebSphere security

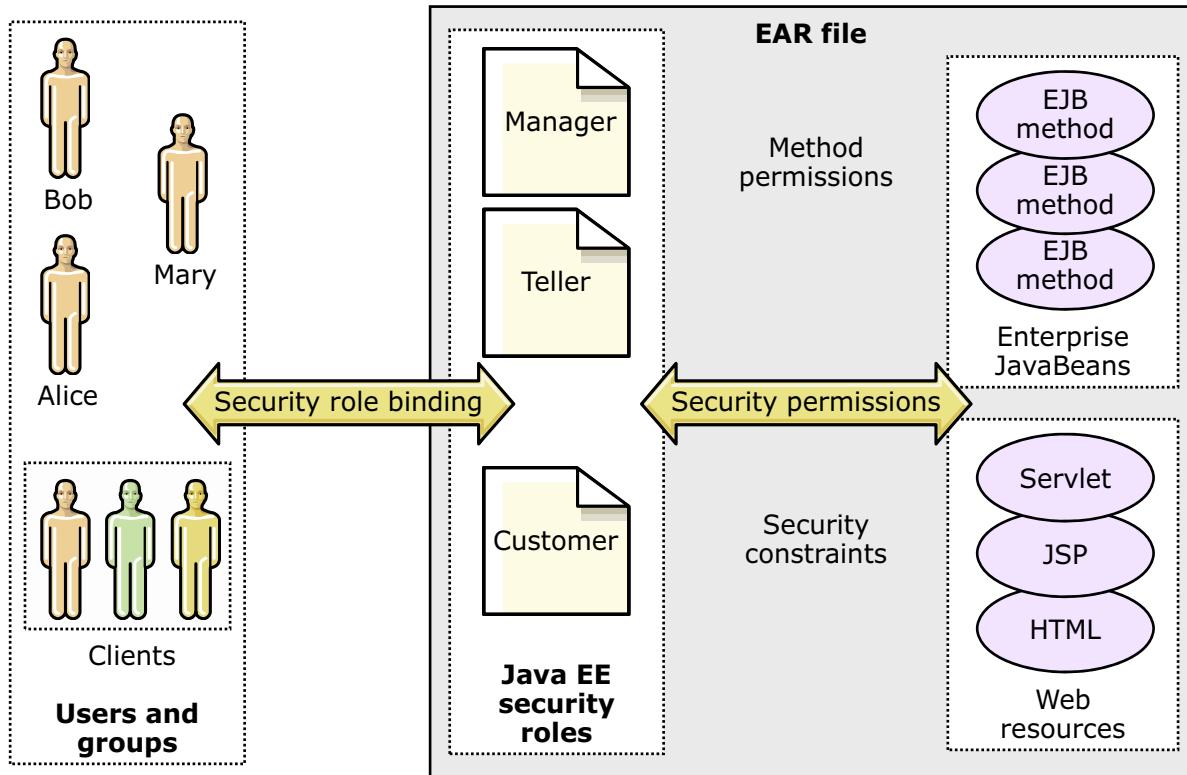
© Copyright IBM Corporation 2016

Figure 9-39. Securing Java EE application artifacts: Part 1

To control who can do what to an application, several XML objects can be added to the EAR file. It is worth noting that these XML objects do not affect the application code itself.

The first step is to create security roles. These roles are merely XML objects that the runtime uses to define which users and groups have access to the application. The second step is to map the security roles to the application code. These mappings are also just XML objects. These mappings are called security constraints (for web container objects) and method permissions (for EJBs). Again, these XML objects do not affect the application code; they are merely part of the deployment descriptors.

Securing Java EE application artifacts: Part 2



WebSphere security

© Copyright IBM Corporation 2016

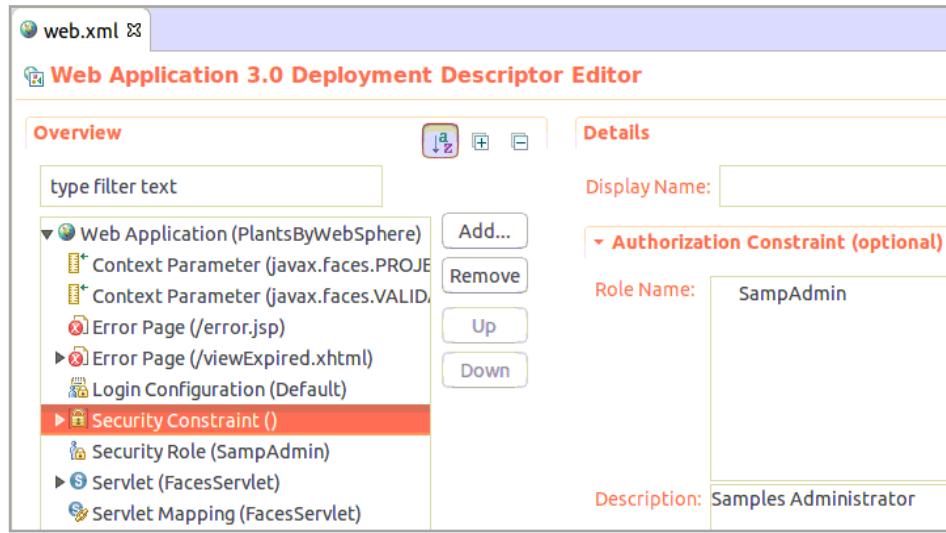
Figure 9-40. Securing Java EE application artifacts: Part 2

During the deployment process (or afterward), the security roles are mapped to the users in groups in the environment. This mapping means that the runtime is able to understand which users and groups in its environment must access the various objects in the application code.

In this example, the security role that is called Manager is mapped to various application methods (perhaps the method that is used to create a customer bank account). The actual user who is called Bob is mapped to the security role Manager during deployment. Therefore, when a user attempts to access the method for creating a bank account, the runtime is aware of a constraint on that method. It checks to make sure that the user is mapped to the Manager security role. If not, the user gets an authorization failure.

Applying application security

- Application security can be applied to resources within an EAR
 - Security roles are defined in the application deployment descriptor
 - Servlets and JSP pages are protected with security constraints, which are mapped to the security roles
 - EJB components are protected with method permissions, which are mapped to the security roles
- The security roles are then mapped to actual users and groups during installation of the application



WebSphere security

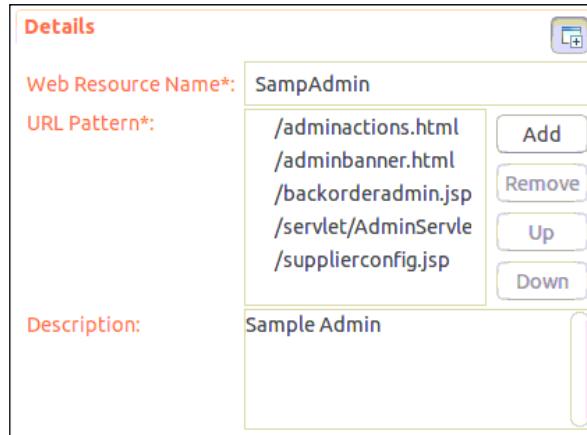
© Copyright IBM Corporation 2016

Figure 9-41. Applying application security

The security roles are defined within the EAR file. This screen shows the SampAdmin security role within the PlantsByWebSphere application.

Creating security constraints

- After the security role is created, it can be mapped in a security constraint to protect web application artifacts



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-42. Creating security constraints

This screen shows the constraints for the PlantsByWebSphere application, which is mapped to the SampAdmin security role.

Using the console to map security roles

- The mapping of users and groups to security roles can take place during or after application installation
 - After installation, use the administrative console
 - Go to the application
 - Under Detailed Properties, select **Security role to user/group mapping**

Select	Role	Special subjects	Mapped users	Mapped groups
<input type="checkbox"/>	SampAdmin	None		

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-43. Using the console to map security roles

As soon as the application is deployed, the console can be used to map the security roles to the actual user and groups that exist within that environment. This mapping can be done at deployment time or any time thereafter through the security role to user and group mapping functions.

9.5. Java 2 security

Java 2 security

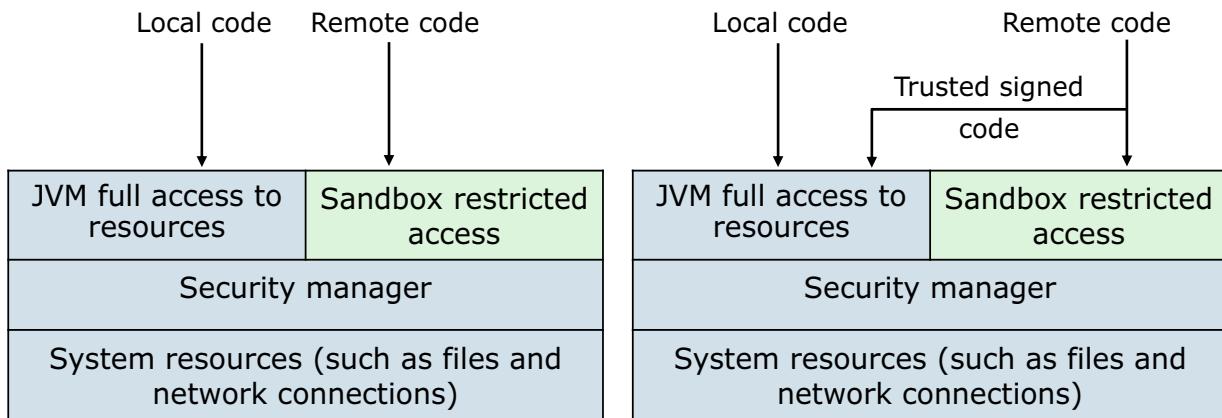
© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-44. Java 2 security

Java security model

- Java 1.0 (sandbox model):
 - Downloaded code (untrusted) runs in a sandbox (restricted environment)
 - Application code (local Java classes) has full access to resources (trusted and no protection)
- Java 1.1 (signed code):
 - Extends 1.0 sandbox model
 - Introduces signed code
 - If the public key that is used to verify the signature is trusted, digitally signed remote code is treated like local code



WebSphere security

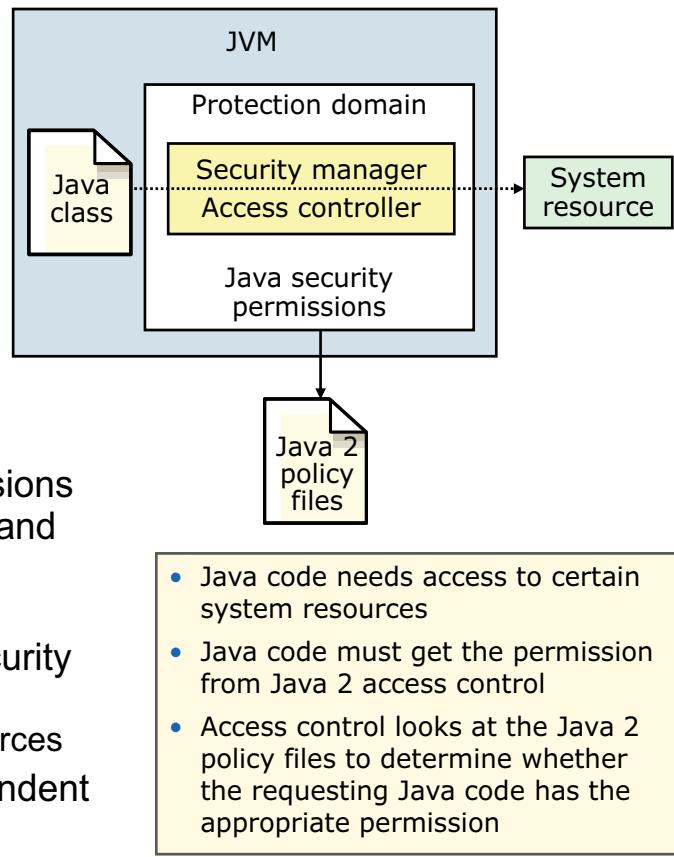
© Copyright IBM Corporation 2016

Figure 9-45. Java security model

This slide introduces the concept of the Java sandbox and what it can do. Initially, in Java 1.0, it was not possible to have something that ran within the sandbox access anything on the local system. This restriction was designed to protect the local system from the code that ran on it (this protection is different from application security, which protects the running code from clients who might attempt to use it). With Java 1.1, it was possible to sign code and trust it as a result. This feature allowed the administrator to have code that was running access certain parts of the local system.

Java security overview

- Protects the system from the applications
- Provides an access control mechanism to manage the application access to system level resources
 - File I/O, network connections (sockets), property files
 - Policy-based
- Policies define a set of permissions available from various signers and code locations
 - Stored in policy files
- All Java code runs under a security policy
 - Grants access to certain resources
- Can be turned on or off independent of administrative security



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-46. Java security overview

Orthogonal to Java EE role-based security, Java 2 security is about protecting system resources. It is policy-based (several .policy files control it) and provides fine-grained access control to system resources, such as:

- File I/O
- Sockets
- Properties

To find Java 2 access exceptions, look for the string `java.security.AccessControlExceptions` in the `SystemOut.log` or `SystemError.log` file.



Enabling Java 2 security

Global security

Use this panel to configure administration and the default application security policy. This security configuration panel provides a central location to define security policy for user applications. Security domains can be defined to override and customize the security settings.

[Security Configuration Wizard](#) [Security Configuration Report](#)

Administrative security

- Enable administrative security
 - [Administrative user roles](#)
 - [Administrative group roles](#)
 - [Administrative authentication](#)

Application security

- Enable application security

Java 2 security

- Use Java 2 security to restrict application access to local resources
 - Warn if applications are granted custom permissions
 - Restrict access to resource authentication data

- Can be enabled and disabled independent of administrative and application security
- Java 2 security provides a policy-based, fine-grained access control mechanism that increases overall system integrity
- Java 2 security checks for permissions before allowing access to certain protected system resources

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-47. Enabling Java 2 security

When Java 2 security is enabled in WebSphere, the security manager component by default throws a `java.security.AccessControl` exception when a permission violation occurs. This exception, if not handled, often causes a runtime failure. This exception is also logged in the `SystemOut.log` file.

However, when the JVM `com.ibm.websphere.java2secman.norethrow` property is set and has a value of true, the security manager does not throw the AccessControl exception: it is only logged.

Note: This property is intended for a sandbox or debug environment only since it instructs the security manager not to throw the AccessControl exception. By not rethrowing the exception, Java 2 security is not truly enforced. Do not use this property in a production environment where a relaxed Java 2 security environment weakens the very integrity that Java 2 security is intended to produce.

The JVM parameter that is entered on the command line starts the server, usually in the script `startServer`. Enter as: `-Dcom.ibm.websphere.java2secman.norethrow=true`

Look in the log for the next line to verify that the previous is in place:

SecurityManager W SECJ0381I: Warning, the `com.ibm.websphere.java2secman.norethrow` property is true. The WebSphere Java 2 Security Manager is not rethrowing

AccessControl exceptions. Do not use this debug setting in a production environment. See the Knowledge Center for Java 2 Security debugging features.

9.6. Security auditing

Security auditing

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-48. Security auditing

Security auditing

- The security auditing subsystem has two primary goals:
 - Confirm the effectiveness and integrity of the existing security configuration
 - Identify areas where improvement to the security configuration might be needed
- The security auditing subsystem can capture the following types of auditable events:
 - Authentication
 - Authorization
 - Principal and credential mapping
 - Audit policy management
 - User registry and identity management
 - Delegation
 - Administrative configuration management

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-49. Security auditing

Security auditing is a new feature in WebSphere Version 7. Auditing can log various information, including:

- Authentication
- Authorization
- Principal-credential mapping
- Audit policy management
- User registry and identity management
- Delegation
- Administrative configuration management

Enable security auditing

- Configuration is necessary before auditing can be enabled
 - Create an audit-specific set of console users or groups and map to Auditor role
 - Define notification mechanism (log file, email)
 - Enable monitoring
- Enable auditing

Security auditing

Security auditing provides a means to gather and store auditable event records to help assure the integrity of the business computing environment.

General Properties		Related Items
<input checked="" type="checkbox"/> Enable security auditing		<ul style="list-style-type: none"> ■ Event type filters ■ Audit service provider ■ Audit event factory configuration ■ Audit encryption key stores and certificates ■ Audit record encryption configuration ■ Audit record signing configuration ■ Audit monitor
Audit subsystem failure action <input type="button" value="No warning"/>		
Primary auditor user name <input type="button" value="wasadmin"/>		
<input type="checkbox"/> Enable verbose auditing		
<input type="button" value="Apply"/> <input type="button" value="Reset"/>		

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-50. Enable security auditing

Enabling security auditing requires some configuration settings. This diagram shows security auditing as it is enabled.



Viewing audit data

- Audit data can be viewed as:
 - Text
 - An HTML report (through wsadmin)

Audit Records

File:///tmp/basicAuditReport.html

Audit Records

Hostname washost . ReportTime Aug 19, 2016, 05:09:02

```

BinaryAudit_washost...stNode01_server1.log x
*****
 Start Display Current Environment *****
WebSphere Platform 9.0.0.0 [ND 9.0.0.0 gm1621.04] [JAVA8 8.0.3.0 pxa6480sr3-20160428_01]
\washostNode01\server1 and process id 39781
Host Operating System is Linux, version 4.2.0-27-generic
Java version = 1.8.0, Java Runtime Version = pxa6480sr3-20160428_01 (SR3), Java Com
was.install.root = /opt/IBM/WebSphere/AppServer
user.install.root = /opt/IBM/WebSphere/AppServer/profiles/profile1
Java Home = /opt/IBM/WebSphere/AppServer/java/8.0/jre
ws.ext.dirs = /opt/IBM/WebSphere/AppServer/java/8.0/lib:/opt/IBM/WebSphere/AppServer/
AppServer/classes:/opt/IBM/WebSphere/AppServer/lib:/opt/IBM/WebSphere/AppServer/ins
ext:/opt/IBM/WebSphere/AppServer/web/help:/opt/IBM/WebSphere/AppServer/deploytool/i
Classpath = /opt/IBM/WebSphere/AppServer/profiles/profile1/properties:/opt/IBM/WebS
AppServer/lib/startup.jar:/opt/IBM/WebSphere/AppServer/lib/bootstrap.jar:/opt/IBM/W
ebsphere/AppServer/lib/lmproxy.jar:/opt/IBM/WebSphere/AppServer/lib/urlprotocols.j
batchboot.jar:/opt/IBM/WebSphere/AppServer/deploytool/itp/batch2.jar:/opt/IBM/WebS
Java Library path = /opt/IBM/WebSphere/AppServer/lib/native/linux/x86_64:/opt/IBM/
compressedrefs:/opt/IBM/WebSphere/AppServer/java/8.0/jre/lib/amd64:/opt/IBM/WebSp
nullDllsdir:/usr/lib:
Orb Version = IBM Java ORB build orb80-20160421.02
Current trace specification = *info
***** End Display Current Environment *****
Seq = 0 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeRea
SessionId = N/A | RemoteHost = null | RemoteAddr = null | RemotePort = null | ProgN

```

Record Number	Event Type
0	SECURITY_RESOURCE_ACCESS
CreationTime=Fri Aug 19 04:45:14 EDT 2016	Action=preinvoke MBean Server (module):getState
RegistryType=WIMUserRegistry	Domain=null
RemoteAddr=null	RemotePort=null
RegistryUserName=null	AppUserName=server:washostNode01Cell_washostNode01_server1
FirstCaller=server:washostNode01Cell_washostNode01_server1	CallerList=null
ResourceName=getState()	ResourceType=SM_MBEAN
1	SECURITY_RESOURCE_ACCESS
CreationTime=Fri Aug 19 04:45:14 EDT 2016	Action=preinvoke MBean Server (module):getState
RegistryType=WIMUserRegistry	Domain=null
RemoteAddr=null	RemotePort=null
RegistryUserName=null	AppUserName=server:washostNode01Cell_washostNode01_server1

WebSphere security © Copyright IBM Corporation 2016

Figure 9-51. Viewing audit data

With security auditing enabled, it is possible to view the security logs as either text files or as HTML reports (generated through wsadmin).

Securing audit records

- Access to audit configurations is restricted
 - To change audit settings, Auditor access is required (Administrator access is not sufficient)
- Audit data can be digitally protected
 - Can be digitally signed
 - Can be encrypted with a separate audit certificate

Security auditing > Audit record encryption configuration

By encrypting the audit records, only a user given the Auditor role

General Properties

Enable encryption

The Audit keystore containing the encryption certificate.

AuditKeyStore

Certificate in keystore

Certificate alias

Security auditing > Audit record signing configuration

Signing audit records provides a means of tamper-proofing the recording of the auditable events.

General Properties

Enable signing

Managed keystore containing the signing certificate

NodeDefaultKeyStore ((cell):washostNode01Cell:(node):washostNode01)

Certificate in keystore

Certificate alias

(none)

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-52. Securing audit records

Unit summary

- Explain basic security concepts
- Describe WebSphere Application Server security
- Describe enhancements to certificate management
- Configure fine-grained administrative security
- Configure application security
- Describe auditing features and functions

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-53. Unit summary

Review questions

1. Which type of security restricts access to the application?
 - A. Administrative security
 - B. Application security
 - C. Java 2 security
 - D. File system security
2. Which type of security restricts access to the operating system?
 - A. Administrative security
 - B. Application security
 - C. Java 2 security
 - D. File system security
3. Which type of security restricts access to the console?
 - A. Administrative security
 - B. Application security
 - C. Java 2 security
 - D. File system security



WebSphere security

© Copyright IBM Corporation 2016

Figure 9-54. Review questions

Write your answers here:

- 1.
- 2.
- 3.

Review answers

1. Which type of security restricts access to the application?
 - A. Administrative security
 - B. Application security
 - C. Java 2 security
 - D. File system security

The answer is B.
2. Which type of security restricts access to the operating system?
 - A. Administrative security
 - B. Application security
 - C. Java 2 security
 - D. File system security

The answer is C.
3. Which type of security restricts access to the console?
 - A. Administrative security
 - B. Application security
 - C. Java 2 security
 - D. File system security

The answer is A.

WebSphere security

© Copyright IBM Corporation 2016

Figure 9-55. Review answers



Exercises:

**Configuring WebSphere
Application Server security**

**Configuring application
security**

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-56. Exercises: Configuring WebSphere Application Server security, Configuring application security

Exercise objectives

After completing Exercise 7, you should be able to:

- Enable WebSphere Application Server security
- Configure administrative security by configuring access to administrative functions
- Configure fine-grained administrative security

After completing Exercise 8, you should be able to:

- Define Java EE security roles
- Define access for resources in an application
- Enable and verify application security

Unit 10. Performance monitoring

Estimated time

00:45

Overview

This unit describes performance monitoring methods and tools that are available through the administrative console.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server V9 documentation in IBM Knowledge Center, Monitoring, and Tuning Performance:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Describe performance monitoring and tuning methods
- Use the Tivoli Performance Viewer to monitor application server resources
- Use the performance servlet to generate performance data
- Configure the Request Metrics tool to generate performance data about the end-to-end request flow
- Use Performance Advisors to generate suggested tuning actions

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-1. Unit objectives

Topics

- Performance tuning and monitoring
- Request metrics
- Performance advisors

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-2. Topics

10.1. Performance tuning and monitoring

Performance tuning and monitoring

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 10-3. Performance tuning and monitoring

The need for performance monitoring and tuning

- How well a website responds while receiving heavy user traffic is an essential factor in the overall success of an organization
- Poor performance results in:
 - Escalated support costs
 - Loss of customer confidence
 - Loss of revenue
- Performance problems can be anywhere in the application server environment
 - Monitoring ensures that applications are running as expected and, if not, determines why and where the problem lies
- WebSphere Application Server can function with default settings but:
 - Improving throughput, and reducing server response times, requires more tuning

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-4. The need for performance monitoring and tuning

The goal of performance monitoring is to collect runtime statistics on your application and its environment to quantify their performance behavior. You can use it to determine whether your application meets its performance objectives and helps to identify any performance bottlenecks.

Tuning performance suggested practices

- Plan for performance
- Take advantage of performance functions (for example, use the dynamic cache service)
- Obtain performance advice from the advisors
- Tune the environment
- Troubleshoot performance problems

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-5. Tuning performance suggested practices

Tuning WebSphere Application Server is a critical part of getting the best performance from your website. But tuning WebSphere Application Server involves analyzing performance data and determining the optimal server configuration. This determination requires considerable knowledge about the various components in the application server and their performance characteristics. The performance advisors encapsulate this knowledge and analyze the performance data. The advisors provide configuration recommendations to improve the application server performance. Therefore, the performance advisors provide a starting point for tuning the application server. Keep in mind the following suggestions:

- Take advantage of performance functions.
- Obtain performance advice from the advisors.
- Tune the environment.
- Troubleshoot performance problems.

Performance terminology

- **Response time** measures an **individual** user's average wait for a request
- Response time includes:
 - Processing time
 - Transit time
 - Wait time in queues
- **Throughput** measures activities that are completed in a unit of time
 - Example: Website pages that are served per second
- **Bottleneck** defines a choke point in the system such as multiple threads that are waiting for some task to complete
- Bottlenecks result when users are queued waiting for a shared resource
 - Processor
 - Data source connections
 - Disk I/O
- **Load** is user activity against a website
 - Users arriving, logging in, sending requests
 - Requests per second, pages per hour

[Performance monitoring](#)

© Copyright IBM Corporation 2016

Figure 10-6. Performance terminology

Some other performance-related terms are path length, scalability, and capacity.

Path length refers to the number of steps that an action takes. Reducing the path length speeds up a website or application. Path length reduction can be achieved by speeding up the steps and reducing the number of steps an activity takes.

Scalability defines how easily a site can expand, sometimes with little warning, to support increased load. Load can come from many sources: new markets, normal growth, and extreme peaks in activity.

Capacity describes how much load the site can support. Discovering the website capacity is the result of performance and load testing.

Tuning parameter hot list

- Review hardware and software requirements
- Install the current refresh pack, fix pack, and the interim fixes
- Check hardware configuration and settings
- Tune the operating system
- Set the minimum and maximum Java virtual machine (JVM) heap sizes
- Tune WebSphere Application Server data sources and connection pools
- Enable the pass by reference option
- Tune related components, for example, the database
- Disable functions that are not required
- Review the application design

[Performance monitoring](#)

© Copyright IBM Corporation 2016

Figure 10-7. Tuning parameter hot list

This hot list contains recommendations that improve performance or scalability, or both, for many applications.

WebSphere Application Server provides several tunable parameters and options to match the application server environment to the requirements of your application.

- **Review the hardware and software requirements.**

For correct functionality and performance, it is critical to satisfy the minimum hardware and software requirements. See the IBM WebSphere Application Server supported hardware, software, and APIs website, which details hardware and software requirements.

- **Install the most current refresh pack, fix pack, and suggested interim fixes.**

The list of suggested updates is maintained on the support site.

- **Check hardware configuration and settings.**

Verify that network interconnections and hardware configuration are set up for peak performance.

- **Tune the operating systems.**

Operating system configuration plays a key role in performance. For example, adjustments such as TCP/IP parameters might be necessary for your application.

- **Set the minimum and maximum Java virtual machine (JVM) heap sizes.**

Many applications need a larger heap size than the default for best performance. It is also advisable to select an appropriate GC policy that is based on the characteristics of the application.

- **Tune WebSphere Application Server JDBC data sources and associated connection pools.**

The JDBC data source configuration might have a significant performance impact. For example, the connection pool size and prepared statement cache must be sized based on the number of concurrent requests that are processed and the design of the application.

- **Enable the pass by reference option.**

Use applications that can take advantage of the pass by reference option to avoid the cost of copying parameters to the stack.

- **Ensure that the transaction log is assigned to a fast disk.**

Some applications generate a high rate of writes to the WebSphere Application Server transaction log. Locating the transaction log on a fast disk or disk array can improve response time.

- **Tune related components, for example, database.**

In many cases, some other component, for example a database, needs adjustments to achieve higher throughput for your entire configuration.

- **Disable functions that are not required.**

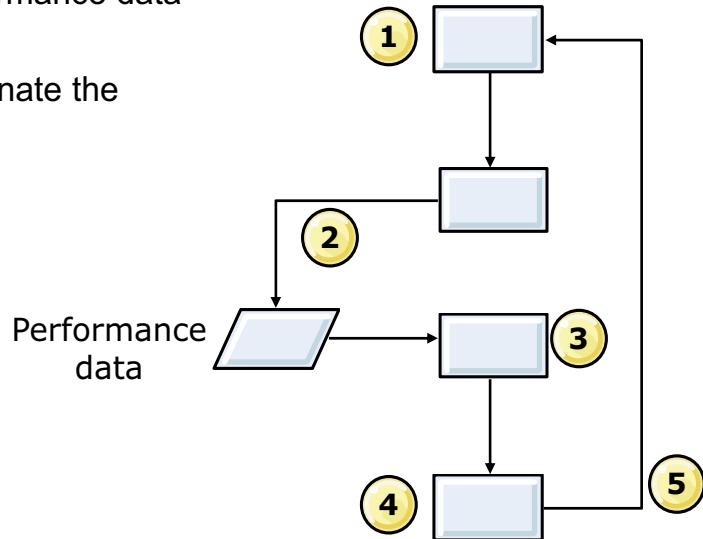
For example, if your application does not use the Web Services Addressing (WS-Addressing) support, disabling this function can improve performance. **Attention:** Use this property with care because applications might require WS-Addressing MAPs to function correctly. Setting this property also disables WebSphere Application Server support for the following specifications, which depend on the WS-Addressing support: Web Services Atomic Transactions, Web Services Business Agreement, and Web Services-Notification. To disable the support for WS-Addressing, see “Enabling Web Services Addressing support for JAX-RPC applications.”

- **Review your application design.**

You can track many performance problems back to the application design. Review the design to determine whether it causes performance problems.

Solving performance problems

- An iterative process:
 1. Load test the system
 2. Monitor and collect performance data
 3. Identify bottlenecks
 4. Tune parameters to eliminate the most severe bottleneck
 5. Repeat



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-8. Solving performance problems

Your application and its runtime environment must also be tuned optimally. This process entails conducting many iterations of a monitor, tune, and test cycle. In short, monitoring, performance testing, and tuning are essential tasks for ensuring a well-performing, application-serving environment.

This process is often iterative because when one bottleneck is removed, some other part of the system now constrains the performance. For example, replacing slow hard disks with faster ones might shift the bottleneck to the processor of a system.

Measuring performance and collecting data

- Establish a benchmark:
 - A **benchmark** is a standard set of application functions to run
 - Use the benchmark to test the application under expected loads
 - Record throughput and response time under normal load and peak load

- Two types of performance data:
 - WebSphere Application Server Performance Monitoring Infrastructure (PMI) provides performance data that you can use to tune application server performance
 - With the Request Metrics tool, you can track individual transactions, recording the processing time in each of the major WebSphere Application Server components

[Performance monitoring](#)

© Copyright IBM Corporation 2016

Figure 10-9. Measuring performance and collecting data

Begin by choosing a *benchmark*, a standard set of operations to run. This benchmark exercises application functions that are experiencing performance problems. Complex systems frequently need a warm-up period to cache objects and optimize code paths. System performance during the warm-up period is much slower than after the warm-up period. The benchmark must be able to generate work that warms up the system before recording the measurements that are used for performance analysis. Depending on the system complexity, a warm-up period can range from a few thousand transactions to longer than 30 minutes.

If the performance problem under investigation occurs only when many clients use the system, then the benchmark must also simulate multiple users. Another key requirement is that the benchmark must be able to produce repeatable results. If the results vary more than a few percent from one run to another, consider the possibility that the initial state of the system might not be the same for each run. It might also be that the measurements are made during the warm-up period, or that the system is running more workloads.

Several tools facilitate benchmark development. The tools range from tools that merely invoke a URL to script-based products that can interact with dynamic data that the application generates. IBM Rational has tools that can generate complex interactions with the system under test and simulate thousands of users. Producing a useful benchmark requires effort and must be part of the development process. Do not wait until an application goes into production to determine how to measure performance.

The benchmark records throughput and response time results in a form to allow graphing and other analysis techniques. The performance data that WebSphere Application Server Performance Monitoring Infrastructure (PMI) provides helps to monitor and tune the application server performance. Request metrics are another source of performance data that WebSphere Application Server provides. Request metrics allow a request to be timed at WebSphere Application Server component boundaries, enabling a determination of the time that is spent in each major component.

WebSphere performance tools (1 of 3)

- WebSphere provides integrated tools to monitor and tune system and application performance:
- Tivoli Performance Viewer
 - With Tivoli Performance Viewer, administrators can monitor the overall health of WebSphere Application Server
 - Accessed from within the administrative console
- Performance advisors
 - Analyze collected performance data and provide configuration recommendations to improve the application server performance
 - Output can be viewed in Tivoli Performance Viewer or in administrative console runtime messages

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-10. WebSphere performance tools (1 of 3)

These tools are explained in greater detail in the subsequent sections.

WebSphere performance tools (2 of 3)

- Request Metrics (tool)
 - With request metrics, you can track individual transactions, recording the processing time in each of the major WebSphere Application Server components
 - Output is viewed in standard logs or by using an Application Response Measurement (ARM)-based tool
- Performance servlet
 - Provides simple retrieval of performance data in XML format
 - Accessed through a browser

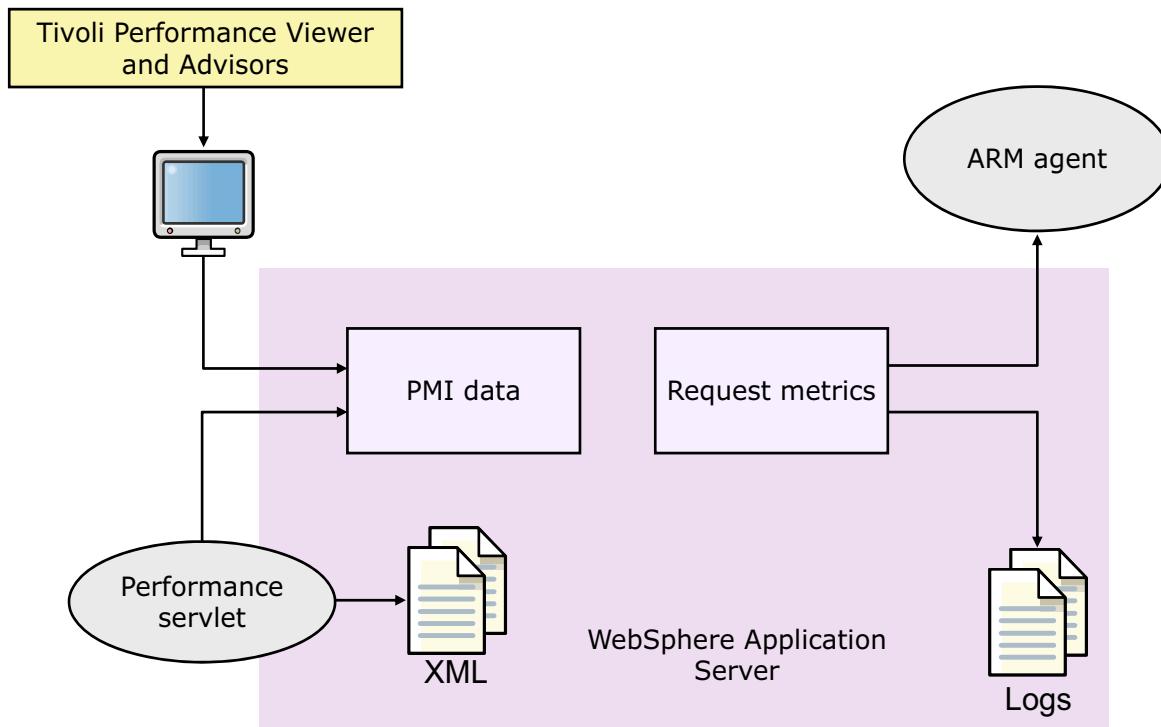
Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-11. WebSphere performance tools (2 of 3)

These tools are explained in greater detail in the subsequent sections.

WebSphere performance tools (3 of 3)



ARM= Application Response Measurement

Performance monitoring

© Copyright IBM Corporation 2016

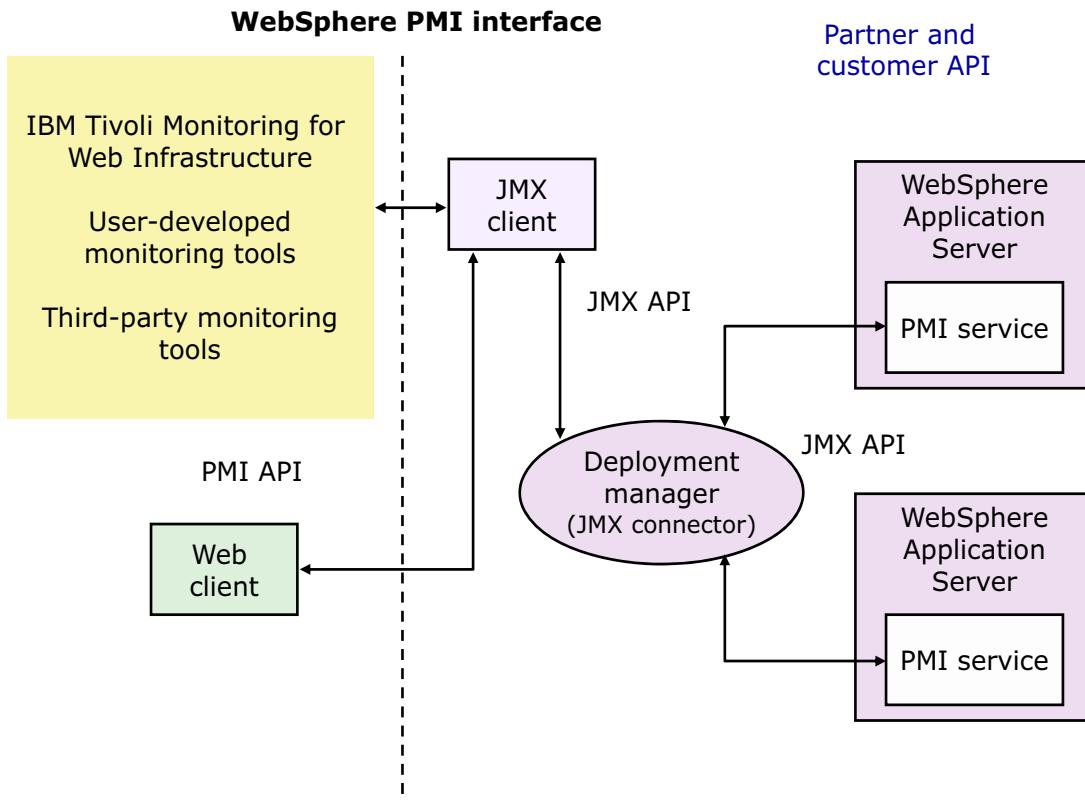
Figure 10-12. WebSphere performance tools (3 of 3)

WebSphere provides integrated tools to monitor and tune system and application performance:

- Tivoli Performance Viewer
 - Gives administrators the ability to monitor the overall health of WebSphere Application Server
 - Accessed from within the administrative console
- Request metrics (tool)
 - Gives you the ability to track individual transactions, recording the processing time in each of the major WebSphere Application Server components
 - Output is viewed in standard logs or by using an Application Response Measurement (ARM)-based tool
- Performance advisors
 - Analyze collected performance data and provide configuration recommendations to improve the application server performance
 - Output is viewed in Tivoli Performance Viewer or in administrative console runtime messages

- Performance servlet
 - Provides simple retrieval of performance data in XML format
 - Accessed through a browser

PMI architecture



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-13. PMI architecture

The Performance Monitoring Infrastructure (PMI) uses a client/server architecture.

The figure shows the overall PMI architecture. On the right side, the server updates and keeps PMI data in memory. The left side displays a web client, a Java client, and a Java Management Extensions (JMX) client that retrieves the performance data. This data consists of counters such as servlet response time and data connection pool usage. The data points are then retrieved by using a web client, a Java client, or a JMX client. WebSphere Application Server contains Tivoli Performance Viewer, a Java client that displays and monitors performance data.

The server collects performance data from various WebSphere Application Server components. A client retrieves performance data from one or more servers and processes the data. WebSphere Application Server supports the Java Platform, Enterprise Edition (Java EE) Management Reference Implementation (JSR-77).

PMI counters are enabled, based on a monitoring or instrumentation level. The levels are None, Basic, Extended, All, and Custom. These levels are specified in the PMI module XML file. Enabling the module at a certain level includes all the counters at that level plus counters from levels below that level. As a result, enabling the module at the extended level enables all the counters at that level plus all the Basic level counters.

JSR-077 defines a set of statistics for Java EE components as part of the Statistic Provider interface. The PMI monitoring level of Basic includes all of the JSR-077 specified statistics. PMI is set to monitor at a Basic level by default.

Types of performance data

- System resources such as processor usage
- WebSphere pools and queues, such as a database connection pool
- Customer application data, such as average servlet response time
- You can also view data for other products or customer applications that implement custom PMI by using the Tivoli Performance Viewer

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-14. Types of performance data

Tivoli Performance Viewer is used to help manage configuration settings by viewing the various graphs or by using the Tivoli Performance Advisor. For example, by looking at the summary chart for thread pools, you can determine whether the thread pool size must be increased or decreased by monitoring the percent usage. After configuration settings are changed based on the data that is provided, you can determine the effectiveness of the changes. To help with configuration settings, use the Tivoli Performance Advisor. The Advisor assesses various data while your application is running, and provides advice about configuration settings to improve performance.

PMI data collection settings

- None
 - All statistics are unavailable
- Basic
 - Statistics that are specified in Java EE specification, plus top statistics like processor usage and live HTTP sessions, are enabled
 - This set is enabled by default and provides basic performance data about runtime and application components (**up to 2% more processor usage**)
- Extended
 - Basic set, plus key statistics from various WebSphere Application Server components like WLM and dynamic caching, are enabled
 - This set provides detailed performance data about various runtime and application components (**up to 3% more processor usage**)
- All
 - All statistics are enabled (**up to 6% more processor usage**)
- Custom
 - Statistics are enabled individually

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-15. PMI data collection settings

PMI uses statistics sets to specify the type and amount of performance data to collect.

PMI counters are enabled, based on a monitoring or instrumentation level. The levels are None, Basic, Extended, All, and Custom. These levels are specified in the PMI module XML file. Enabling the module at a certain level includes all the counters at that level plus counters from levels below that level. So, enabling the module at the extended level enables all the counters at that level plus all the Basic level counters.

Using the administrative console to enable PMI

Application servers > server1 > Performance Monitoring Infrastructure (PMI)

Use this page to configure Performance Monitoring Infrastructure (PMI)

Configuration

General Properties

Enable Performance Monitoring Infrastructure (PMI)

Use sequential counter updates

Currently monitored statistic set

- None
No statistics are enabled.
- Basic
Provides basic monitoring, including Java EE and the top 38 statistics.
- Extended
Provides extended monitoring, including the basic level of monitoring plus workload monitor, performance advisor, and Tivoli resource models.
- All
All statistics are enabled.
- Custom
Provides fine-grained control to selectively enable statistics.

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-16. Using the administrative console to enable PMI

- Click **Servers > Server Types > WebSphere Application Servers > <server_name>**
- On the Configuration tab, under Performance, click **Performance Monitoring Infrastructure (PMI)**
- Select the **Enable Performance Monitoring Infrastructure (PMI)** check box
- Select the statistics set

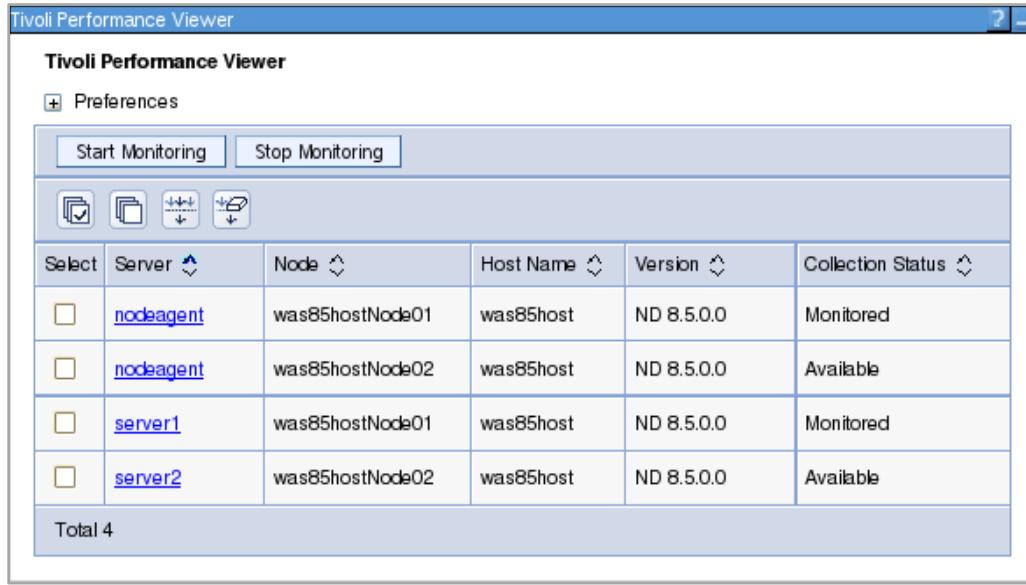
Click **Servers > Server Types > WebSphere Application Servers > <server_name>**.

On the Configuration tab, under Performance, click **Performance Monitoring Infrastructure (PMI)**.

Select the **Enable Performance Monitoring Infrastructure (PMI)** check box.

Start monitoring

- After enabling PMI, select the server and click **Start Monitoring** on the Tivoli Performance Viewer page
 - In the administrative console, select **Monitoring and Tuning > Performance Viewer > Current activity**



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-17. Start monitoring

- Deprecated feature:** Tivoli Performance Viewer displays graphics in either the Scalable Vector Graphics (SVG) format or as a static image in the JPG format. If you do not have the Adobe SVG browser plug-in, you are prompted to download and install it. If you select not to install the plug-in (by selecting Cancel), Tivoli Performance Viewer displays the static image.

Installing the Adobe SVG plug-in is advantageous for several reasons. First, the SVG format provides interactive graphics that provide more information when you hover your mouse over a point, line, or legend item. You can also use the SVG format to click a point and see details for it. Second, using the SVG format provides a performance benefit because the work to display the SVG image is done on the client side. When viewing a static image, the application server must convert the SVG image into a static image, which is a processor-intensive and memory-intensive operation. If your browser is Internet Explorer 7, the Adobe SVG installation prompt might be inaccessible. To resolve the problem, you can reinstall Adobe SVG.

- For users who are migrating from version 7:** Beginning with version 8, the Tivoli Performance Viewer graph uses Dojo technology for plotting the performance activity rather than the Scalable Vector Graphics (SVG) format. The Dojo format provides a better user experience and is more processor and memory efficient for the application server. The SVG format is still supported but is deprecated in version 8 of this product. To use the SVG format and image format, set the JVM property to false; for example:

```
com.ibm.websphere.tpv.DojoGraph=false
```

If the property is set to false, Dojo is disabled, and Tivoli Performance Viewer uses the SVG format to display interactive graphics; or it uses the JPG format to display non-interactive graphics.

When you specify to use the SVG format by setting

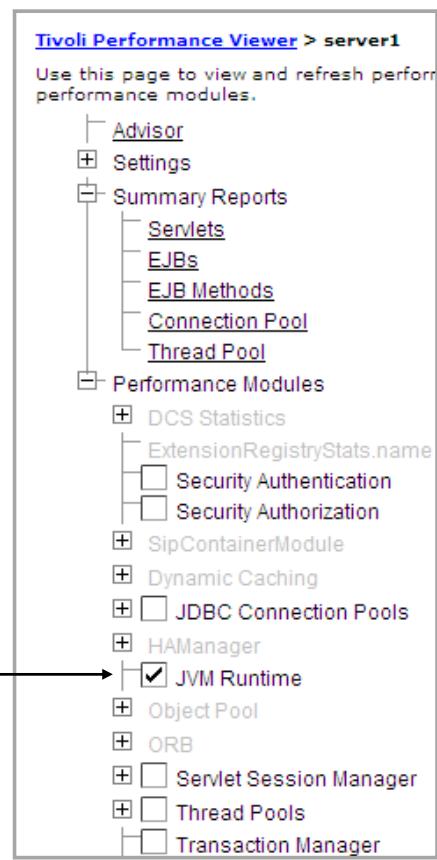
`com.ibm.websphere.tpv.DojoGraph=false`, if you do not have the Adobe SVG browser plug-in, you are prompted to download and install it. If you select not to install the plug-in (by selecting Cancel), Tivoli Performance Viewer displays the static image. If your browser is Internet Explorer 7, the Adobe SVG installation prompt might be inaccessible. To resolve the problem, you can reinstall Adobe SVG. By default, this property is set to a value of true to use the Dojo format.



Tivoli Performance Viewer (1 of 4)

- Select one or more performance modules to monitor from the navigation page
- Click **View Module(s)**
- The performance data is dynamically displayed in a chart and table
- Note: The disabled modules become active when you enable the Extended or All PMI statistics sets

JVM runtime module is selected



Performance monitoring

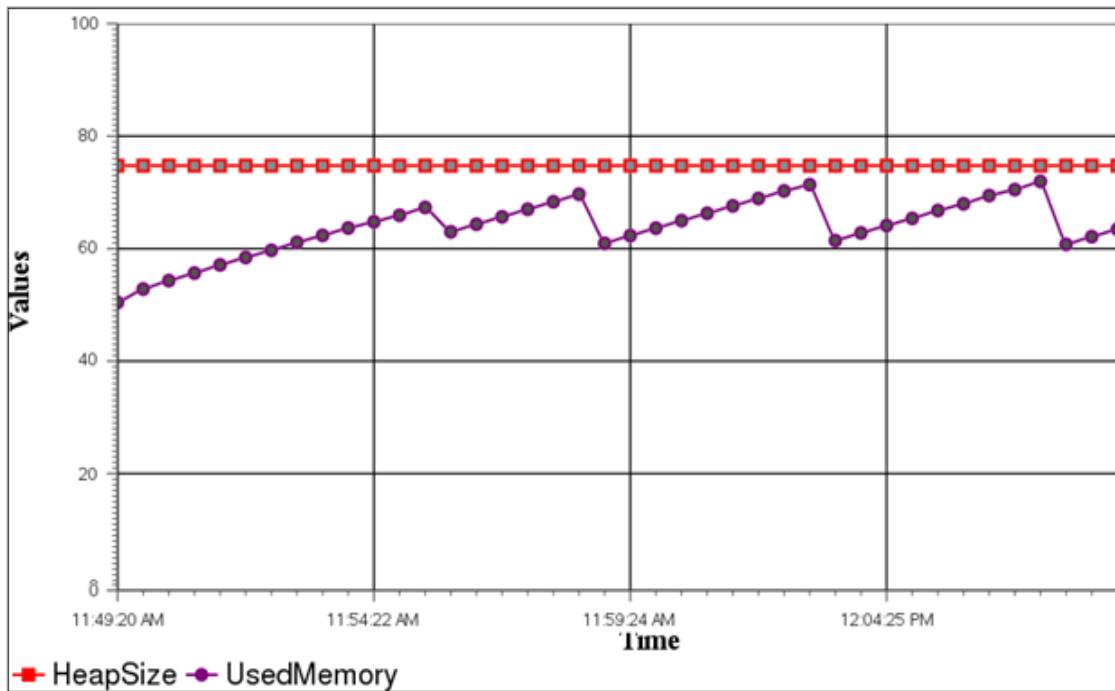
© Copyright IBM Corporation 2016

Figure 10-18. Tivoli Performance Viewer (1 of 4)

The screen capture on this slide shows the navigation tree in Tivoli Performance Viewer where you can select which components to monitor. In this case, the JVM Runtime module is selected.

Tivoli Performance Viewer (2 of 4)

- Chart view of JVM metrics



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-19. Tivoli Performance Viewer (2 of 4)

Statistics for the selected modules are displayed as a line graph. You can select which metrics you want to display in the graph, and you can optionally show the legend. In this example, only the heap size and used memory metrics are displayed for the JVM runtime. The sawtooth pattern of the used memory graph is typical of a steady state JVM. The periodic reductions in used memory correspond to JVM garbage collections, which return unused memory to the heap.

Tivoli Performance Viewer (3 of 4)

- Chart view controls
 - Reset To Zero
 - Clear Buffer
 - View Table
 - Show or Hide Legend



The screenshot shows a software window titled 'Tivoli Performance Viewer'. At the top, there are four buttons: 'Reset To Zero' (highlighted in blue), 'Clear Buffer', 'View Table', and 'Hide Legend'. Below the buttons is a table with the following columns: 'Select', 'Marker', 'Name', 'Value', 'Scale', 'Update', and 'Scaled Value'. The table displays five rows of data for 'JVM Runtime' metrics:

Select	Marker	Name	Value	Scale	Update	Scaled Value
<input checked="" type="checkbox"/>		HeapSize ?	86528.0	<input type="text" value="0.0010"/>		86.52801
<input type="checkbox"/>		FreeMemory ?	31048.0	<input type="text" value="0.0010"/>		31.048002
<input checked="" type="checkbox"/>		UsedMemory ?	55479.0	<input type="text" value="0.0010"/>		55.479004
<input type="checkbox"/>		UpTime ?	1765.0	<input type="text" value="0.01"/>		17.65
<input type="checkbox"/>		ProcessCpuUsage ?	0.0	<input type="text" value="1.0"/>		0.0

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-20. Tivoli Performance Viewer (3 of 4)

The **Reset To Zero** button sets a new baseline by using the current counter readings at the instant the button is clicked. Future data points are plotted on the graph relative to their position at the time **Reset To Zero** is clicked. Data points that are gathered before the time **Reset To Zero** is clicked are not displayed, although they are still held in the Tivoli Performance Viewer buffer. If **Undo** **Reset To Zero** is clicked again, Tivoli Performance Viewer displays all data that is recorded from the original baseline, not from the **Reset To Zero** point.

Click **Clear Buffer** to remove the PMI data from a table or chart.

Tivoli Performance Viewer (4 of 4)

- Table view

Time	JVM Runtime HeapSize	JVM Runtime FreeMemory	JVM Runtime UsedMemory
12:29:29 PM	86528.00	30979.00	55548.00
12:28:59 PM	86528.00	32317.00	54210.00
12:28:29 PM	86528.00	18502.00	68025.00
12:27:59 PM	86528.00	19513.00	67014.00
12:27:29 PM	86528.00	20768.00	65759.00
12:26:59 PM	86528.00	22197.00	64330.00
12:26:29 PM	86528.00	23452.00	63075.00

Performance monitoring

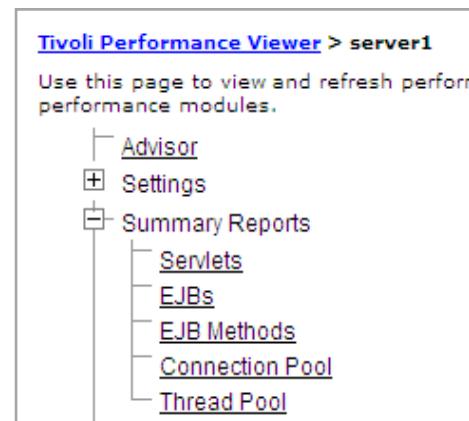
© Copyright IBM Corporation 2016

Figure 10-21. Tivoli Performance Viewer (4 of 4)

Statistics can also be viewed in a table format, by clicking **View Table** in the Tivoli Performance Viewer.

Summary reports

- View a statistics report by selecting one of the summary reports
- Servlets
 - Lists all servlets that are running in the current application server
- EJB components
 - Lists all EJB components that are running in the server
 - Amount of time that is spent in their methods
 - Number of EJB invocations
 - Total time that is spent in each EJB
- EJB methods
 - Details about methods
- Connection pool
 - Lists all data source connections that are defined in the application server and show their usage over time
- Thread pool
 - Shows the usage of all thread pools in the application server over time



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-22. Summary reports

The following summary reports help you to analyze various resources in the application server:

- The **Servlet** summary lists all servlets that are running in the current application server. Use the servlet summary view to quickly find the most time-intensive servlets and the applications that use them, and to determine which servlets are used most often. You can sort the summary table by any of the columns.

Tip: Sort by **Average Response Time** to find the slowest servlet or JSP page. Sort by **Total Requests** to find the servlet or JSP that is used the most. Sort by **Total Time** to find the most costly servlet or JSP.

- The **Enterprise JavaBeans** (EJB) summary lists the following information: all enterprise beans that are running in the server, the amount of time that is spent in their methods, the number of EJB invocations, and the total time that is spent in each enterprise bean.

`total_time = number_of_invocations * time_in_methods`

Sort the various columns to find the most expensive enterprise bean. Also, if the PMI counters are enabled for individual EJB methods, you can select a check box next to the EJB name to see statistics for each of the methods.

Tip: Sort by **Average Response Time** to find the slowest enterprise bean. Sort by **Method Calls** to find the enterprise bean that is used the most. Sort by **Total Time** to find the most costly enterprise bean.

- The **EJB method** summary shows statistics for each EJB method. Use the EJB method summary to find the most costly methods of your enterprise beans.

Tip: Sort by **Average Response Time** to find the slowest EJB method. Sort by **Method Calls** to find the EJB method that is used the most. Sort by **Total Time** to find the most costly EJB method.

- The **connection pool** summary lists all data source connections that are defined in the application server and shows their usage over time.

When the application is experiencing normal to heavy usage, the pools that the application uses must be nearly fully used. Low utilization means that resources are being wasted by maintaining connections or threads that are never used. Consider the order in which work progresses through the various pools. If the resources near the end of the pipeline are underused, it might mean that resources near the front are constrained or that more resources than necessary are allocated near the end of the pipeline.

- The **thread pool** summary shows the usage of all thread pools in the application server over time.

When the application is experiencing normal to heavy usage, the pools that the application uses must be nearly fully used. Low utilization means that resources are being wasted by maintaining connections or threads that are never used. Consider the order in which work progresses through the various pools. If the resources near the end of the pipeline are underused, it might mean that resources near the front are constrained or that more resources than necessary are allocated near the end of the pipeline.

Example: Servlet Summary Report

- Use the servlet summary to:
 - Find the servlets that use the most time and the applications that use them
 - Determine which servlets are called most often
- You can sort the summary table by any of the columns

Servlets Summary Report					
More information about this page					
<input type="button" value="Start Logging"/>					
Name	Application	Total Requests	Avg Resp Time (ms)	Total Time (ms)	Time
FacesServlet	PlantsByWebSphere.war	131	93.374	12,232	1:28:40 PM
ples.pbw.war.ImageServlet	PlantsByWebSphere.war	26	58.846	1,530	1:28:40 PM
Snoop Servlet	DefaultWebApplication.war	3	7.667	23	1:28:40 PM
/HitCount.jsp	DefaultWebApplication.war	2	6.5	13	1:28:40 PM

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-23. Example: Servlet Summary Report

In this screen capture, the servlets report is shown with the total requests column sorted. You can see which two servlets in the PlantsByWebSphere application are used most frequently.

Performance servlet overview

- Provides performance data output as an XML document, as the provided document type definition (DTD) describes
 - The DTD is located inside the `PerfServletApp.ear` file
- Deployed exactly as any other servlet:
 1. Deploy the servlet on a single application server instance within the domain
 2. After the servlet deploys, you can start it to retrieve performance data for the entire domain; start the performance servlet by accessing the following default URL:
`http://<hostname>/wasPerfTool/servlet/perfservlet`

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-24. Performance servlet overview

The servlet provides a way to use an HTTP request to query the performance metrics for an entire WebSphere Application Server administrative domain. Because the servlet provides the performance data through HTTP, issues such as firewalls are trivial to resolve.

The PerfServlet provides the performance data output as an XML document, as described in the provided document type definition (DTD). In the XML structure, the leaves of the structure provide the actual observations of performance data and the paths to the leaves that provide the context.

The performance servlet EAR file `PerfServletApp.ear` is in the `WAS_HOME/installableApps` directory, where `WAS_HOME` is the installation path for WebSphere Application Server.



Performance servlet output

```

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9080/wasPerfTool/servlet/perfservlet
http://localhost:90...servlet/perfservlet
<Stat name="Snoop Servlet">
  - <Stat name="URLs">
    - <Stat name="/snoop">
      <CountStatistic ID="15" count="3" lastSampleTime="1312564964205"
      name="URIRequestCount" startTime="1312564957039" unit="N/A"/>
      <RangeStatistic ID="16" highWaterMark="1" integral="24.0"
      lastSampleTime="1312566246451" lowWaterMark="0"
      mean="1.861313528957385E-5" name="URICurrentRequests"
      startTime="1312564957039" unit="N/A" value="0"/>
      <TimeStatistic ID="17" lastSampleTime="1312564964209" max="10" min="4"
      name="URIServiceTime" startTime="1312564957039" totalTime="23"
      unit="MILLISECOND"/>
      <TimeStatistic ID="19" lastSampleTime="1312564957039" max="0" min="0"
      name="AsyncContext Response Time" startTime="1312564957039"
      totalTime="0" unit="MILLISECOND"/>
    </Stat>
    <CountStatistic ID="15" count="3" lastSampleTime="1312564964205"
    name="URIRequestCount" startTime="1312559331856" unit="N/A"/>
    <RangeStatistic ID="16" highWaterMark="1" integral="128.0"
    lastSampleTime="1312566246451" lowWaterMark="0"
    mean="1.851156596856852E-5" name="URICurrentRequests"
    startTime="1312559331855" unit="N/A" value="0"/>
  </Stat>

```

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-25. Performance servlet output

This screen capture shows sample output that the performance servlet generates, which is displayed in a browser. The section that is shown here represents a request to the Snoop Servlet.

Tip: The PerfServlet is a sample monitoring tool that uses WebSphere Application Server administration and monitoring interfaces to extract and display performance data. Using the PerfServlet is not intended for real-time performance monitoring in production environments or for use in large topologies. For these environments, you might use the Tivoli Performance Viewer or IBM Tivoli Composite Application Manager for WebSphere Application Server.

Specific tips for the PerfServlet include the following practices:

- **PerfServlet resource usage:** The PerfServlet is not designed to run concurrently. Being a single threaded servlet, it would collect the data sequentially from available servers. This single threaded operation can cause higher response times when the PerfServlet is used in larger deployments.
- **PerfServlet in large deployments:** By default, when the PerfServlet is first initialized, it retrieves the list of nodes and servers within the cell in which it is deployed. Because collecting this data requires system processing time, the PerfServlet holds this information as a cached list. To force the servlet to refresh its configuration, you can use the option "refreshconfig=true". However, using this option is not suggested unless required because

this option adds extra resource usage to the PerfServlet processing. Use option, node, and server, if you are looking for performance data of a specific server.

- **PerfServlet response time:** How responsive the PerfServlet is depends on the following factors: number of application servers that exist in the cell and number of resources that are configured in the cell (including applications).
- **PerfServlet alternative:** If you are looking for an alternative to using the PerfServlet to capture data programmatically, see the Perf MBean programming interfaces documentation. The documentation can be found under the **Reference > Programming Interfaces > MBean interfaces** section of the WebSphere Application Server documentation in IBM Knowledge Center.

10.2. Request metrics

Request metrics

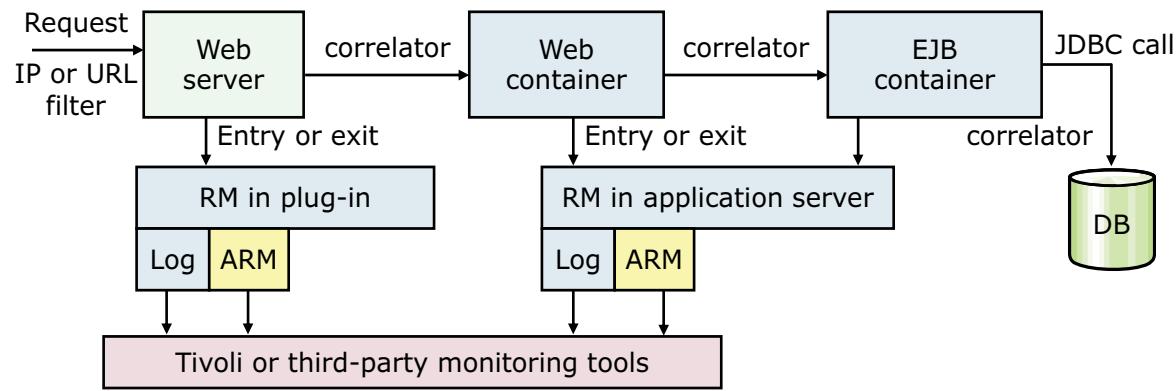
© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 10-26. Request metrics

Request metrics (RM) overview

- This tool is a tracing facility to measure the amount of time a request spends in each component that is traversed during its execution
- Captured information includes:
 - Elapsed time in the web server
 - Response time of started components in the web and EJB containers
 - Response time of related JDBC calls
- Writes trace records to `SystemOut.log` or sends metrics to an Application Response Measurement (ARM) agent



Performance monitoring

© Copyright IBM Corporation 2016

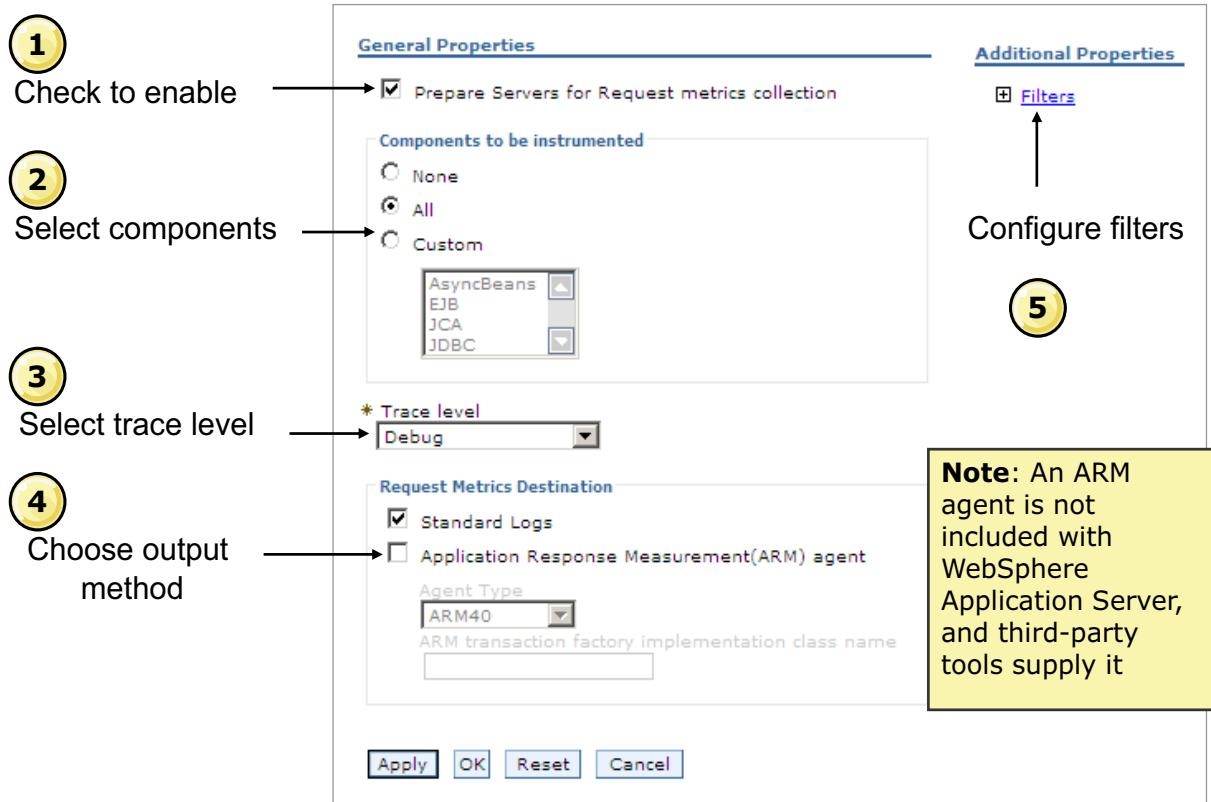
Figure 10-27. Request metrics (RM) overview

Request metrics allow you to monitor the transaction flow and analyze the response time of the components that are involved in processing it. This analysis can help you target performance problem areas and debug resource constraint problems. For example, it can help determine whether a transaction spends most of its time in the web server plug-in, the web container, the Enterprise JavaBeans (EJB) container, or the back-end database. The response time that is collected for each level includes the time that is spent at that level and the time that is spent in the lower levels. For example, if the total response time for the servlet is 130 milliseconds, and it includes 38 milliseconds from the enterprise beans and JDBC calls, then 92 ms can be attributed to the servlet process.

An ARM agent is not included with WebSphere Application Server, but third-party tools can provide it.



Enabling request metrics collection



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-28. Enabling request metrics collection

In the administrative console, select **Monitoring and Tuning > Request Metrics** and select the check box to **Prepare Servers for Request metrics collection**.

Trace level specifies how much trace data to accumulate for a particular transaction. **Trace level** and **Components to be instrumented** work together to control whether a request is instrumented or not. The trace level can be set to one of the following values:

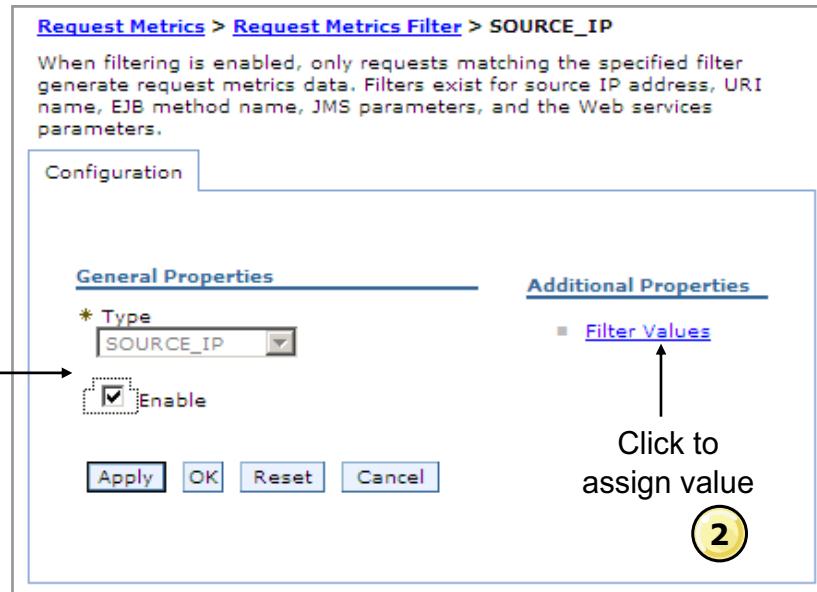
- **None:** No instrumentation.
- **Hops:** Generates instrumentation information about process boundaries only. When this setting is selected, you see the data at the application server level, not the level of individual components such as enterprise beans or servlets.
- **Performance_debug:** Generates the data at Hops level and the first level of the intra-process servlet and Enterprise JavaBeans (EJB) call (for example, when an inbound servlet forwards to a servlet and an inbound EJB calls another EJB). Other intra-process calls like naming and service integration bus (SIB) are not enabled at this level.
- **Debug:** Provides detailed instrumentation data, including response times for all intra-process calls. Note: Requests to servlet filters are instrumented only at this level.
- **Standard logs:** Enables the request metrics logging feature. Select this check box to trigger the generation of request metrics logs in the `SystemOut.log` file. Note: Since enabling the request

metrics logging feature increases processor usage, it is suggested that you use this feature together with filters so that only selected requests are instrumented.

- **Application Response Measurement (ARM) agent:** Allows request metrics to call an underlying Application Response Measurement (ARM) agent. Before enabling ARM, you must install an ARM agent and configure it to the appropriate class path and path, following the instructions of the ARM provider.

Isolating performance for specific types of requests

- Click **Monitoring and Tuning > Request Metrics > Filters**
- Select a filter type (for example, SOURCE_IP)
- Assign filter values



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-29. Isolating performance for specific types of requests

The request metrics filters are enabled according to your configuration. For example, if you enabled source IP, only requests whose source IP matches the one specified in the filter are instrumented.

Note: Filters are checked only for edge transactions. An edge transaction is the transaction that first enters an instrumented system. For example, if a servlet calls an Enterprise JavaBeans component, the servlet is the edge transaction. The servlet must not be instrumented at the web server plug-in, and the URI and SOURCE_IP filters must be checked for the servlet request. However, when the request comes to the EJB container, the EJB filter is not checked because it is no longer an edge transaction.

You must regenerate the web server plug-in configuration file after modifying the request metrics configuration.

Example request metrics data

- Request metrics data from a `SystemOut.log` file

```
[8/5/11 16:12:31:338 EDT] 00000029 PmiRmArmWrapp I PMRM0003I:  
parent:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32874,event=1 -  
current:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32878,event=1 type=JDBC  
detail=java.sql.PreparedStatement.executeQuery() elapsed=0

[8/5/11 16:12:31:346 EDT] 00000029 PmiRmArmWrapp I PMRM0003I:  
parent:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32874,event=1 -  
current:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32879,event=1 type=JDBC  
detail=javax.resource.spi.XAResource.end(Xid, int) elapsed=0

[8/5/11 16:12:31:350 EDT] 00000029 PmiRmArmWrapp I PMRM0003I:  
parent:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32874,event=1 -  
current:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32880,event=1 type=JDBC  
detail=javax.resource.spi.XAResource.commit(Xid, boolean) elapsed=0

[8/5/11 16:12:31:366 EDT] 00000029 PmiRmArmWrapp I PMRM0003I:  
parent:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32874,event=1 -  
current:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32881,event=1 type=JDBC  
detail=javax.resource.spi.ManagedConnection.cleanup() elapsed=0
```

Figure 10-30. Example request metrics data

The example of request metrics data that is shown on this slide details the use of the prepared statement cache to make an SQL call. You can trace the steps that are involved and timings for this database transaction.

10.3. Performance advisors

Performance advisors

© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 10-31. Performance advisors

Performance advisors overview

- WebSphere provides two separate advisors:
 - Performance and Diagnostic Advisor – disabled by default
 - Tivoli Performance Viewer Advisor
- Both provide configuration advice that is based on collected PMI data on a per server basis
 - Advisors do not compare counters among different application servers
- Provides advice that is based on basic rules for tuning WebSphere Application Server
 - Rules are IBM-defined and not configurable
- Advisors do not automatically tune WebSphere based on advice
 - Administrator must manually apply recommendations
 - Suggested settings must be checked against baseline performance to verify improvement: tune, test, monitor

[Performance monitoring](#)

© Copyright IBM Corporation 2016

Figure 10-32. Performance advisors overview

The advisors provide advice on the following application server resources: thread pools, persisted session sizes, cache sizes, and JVM heap size.

For example, consider the data source statement cache. It optimizes the processing of *prepared statements* and *callable statements* by caching those statements that are not used in an active connection. (Both statements are SQL statements that essentially run repeatable tasks without the costs of repeated compilation.) If the cache is full, an old entry in the cache is discarded to make room for the new one. The best performance is generally obtained when the cache is large enough to hold all of the statements that are used in the application. The PMI counter, prepared statement cache discards, indicates the number of statements that are discarded from the cache.

The performance advisors check this counter and provide recommendations to minimize the cache discards. Another example is thread or connection pooling. The idea behind pooling is to use an existing thread or connection from the pool instead of creating an instance for each request. Because each thread or connection in the pool consumes memory and increases the context-switching cost, the pool size is an important configuration parameter. A pool that is too large can hurt performance as much as a pool that is too small. The performance advisors use PMI information about current pool usage, minimum or maximum pool size, and the application server processor utilization to suggest efficient values for the pool sizes.

The advisors can also issue diagnostic advice to help in problem determination and health monitoring. For example, if your application requires more memory than is available, the diagnostic adviser tells you to increase the size of the heap for the application server.

Performance and Diagnostic Advisor (1 of 5)

- Performance advice:
 - Object Request Broker (ORB) service thread pools
 - Web container thread pools
 - Connection pool size
 - Persisted session size and time
 - Prepared statement cache size
 - Session cache size
 - Memory leak detection
- Diagnostic advice:
 - Connection factory diagnostic messages
 - Data source diagnostic messages
- Connection usage diagnostic messages
 - Detection of connection use by multiple threads
 - Detection of connection use across components

Performance

- [Performance Monitoring Infrastructure \(PMI\)](#)
- [Performance and Diagnostic Advisor Configuration](#)

Performance monitoring

© Copyright IBM Corporation 2016

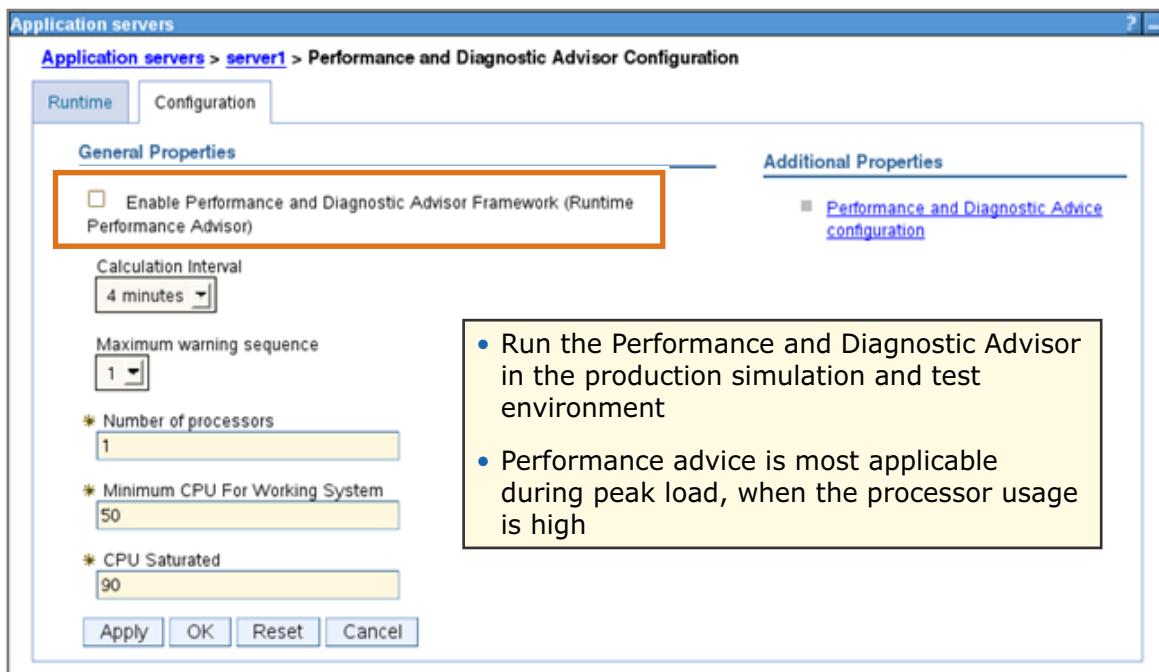
Figure 10-33. Performance and Diagnostic Advisor (1 of 5)

The Performance and Diagnostic Advisor runs in the Java virtual machine (JVM) process of the application server; therefore, the performance cost is minimal.

To access the Performance and Diagnostic Advisor Configuration, click **Servers > Server Types > WebSphere application servers > server_name > Performance and Diagnostic Advisor Configuration**.

Performance and Diagnostic Advisor (2 of 5)

- Click **Servers > Server Types > WebSphere application servers > server_name > Performance and Diagnostic Advisor**



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-34. Performance and Diagnostic Advisor (2 of 5)

The Performance and Diagnostic Advisor analyzes PMI data and receives notifications about performance and diagnostic information from components. Use this page to specify settings for the Performance and Diagnostic Advisor. Performance issues can be related to memory leaks in the system. Use the Memory Dump Diagnostic for Java tool, which is a separate memory leak analysis utility, for detecting memory leaks.

The Performance and Diagnostic Advisor Framework is disabled by default. Each time that you enable it for an application server, you see the warning message:

Run the Performance and Diagnostic Advisor in the Production Simulation and Test environment. Performance advice is most applicable during peak load, when the processor utilization is high.

Performance and Diagnostic Advisor (3 of 5)

- Advisor configuration pane (on both configuration and runtime tabs)
- Select advice and click Start or Stop

<input type="button" value="Start"/> <input type="button" value="Stop"/>					
	<input checked="" type="checkbox"/>	Advice name	Advice applied to component	Advice type	Performance impact
You can administer the following resources:					
	<input checked="" type="checkbox"/>	Thread Max Connections exceeded Diagnostic Alert	J2C Connection Manager	Diagnostic	High
	<input type="checkbox"/>	LTC Nesting Threshold Exceeded Alert	J2C Connection Manager	Diagnostic	High
	<input type="checkbox"/>	Serial Reuse Violation Diagnostic Alert	J2C Connection Manager	Diagnostic	High
	<input type="checkbox"/>	Session Cache Size with Overflow Disabled	Web Container Session Manager	Performance	Low

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-35. Performance and Diagnostic Advisor (3 of 5)

Advice type categorizes the primary intent of a piece of Advice.

Use Advice type for grouping, and then enable or disable sets of advice that is based on your performance goal. Advice has the following types:

- Performance: Performance advice provides tuning recommendations, or identifies problems with your configuration from a performance perspective.
- Diagnostic: Diagnostic advice provides automated logic and analysis that relates to problem identification and analysis. These types of advice are issued when the application server encounters unexpected circumstances.

Performance impact generalizes the negative effect on performance that an alert might incur.

The performance impact of a particular piece of advice is highly dependent upon the scenario that is run and upon the conditions that are met. The performance categorization of alerts is based on worst case scenario measurements. The performance categorizations are:

- **Low:** Advice has minimal negative effect on performance. Advice might be run in test and production environments. Cumulative negative effect on performance is within run-to-run variance when all advice of this type is enabled.

- **Medium:** Advice has measurable but low negative effect on performance. Advice might be run within test environments, and might be run within production environments if deemed necessary. Cumulative negative effect on performance is less than 4% when all advice of this type is enabled.
- **High:** Advice impact is high or unknown. Advice might be run during problem determination tests and functional tests. It is not run in production simulation or production environments unless deemed necessary. Cumulative negative effect on performance might be significant when all advice of this type is enabled.

Performance and Diagnostic Advisor (4 of 5)

- Tuning advice can be viewed in Runtime Events
- Click any TUNE message link for details

Timestamp	Message Originator	Message
Aug 9, 2011 11:47:00 AM EDT	com.ibm.ws.performance.tuning.serverAlert.TraceResponse	TUNE0204W: Decreasing the size of the Web Containe
Aug 9, 2011 11:47:00 AM EDT	com.ibm.ws.performance.tuning.serverAlert.TraceResponse	TUNE0204W: Decreasing the size of the Web Containe
Aug 9, 2011 11:47:00 AM EDT	com.ibm.ws.performance.tuning.serverAlert.TraceResponse	TUNE0204W: Decreasing the size of the ORB thread p
Aug 9, 2011 11:46:42 AM EDT	com.ibm.ws.performance.tuning.serverAlert.TraceResponse	TUNE9001W: Heap utilization patterns indicate tha

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-36. Performance and Diagnostic Advisor (4 of 5)

Tuning advice is provided as messages written to the runtime events. The `TUNExxxx` messages are typically at the Warning level.

Examples include:

- `TUNE0201W`: The rate of discards from the prepared statement cache is high. Increase the size of the prepared statement cache for the data source.
- `TUNE0207W`: Utilization of the connection pool is high. Performance might be improved by increasing the `maxPoolSize` for data source `{DS_name}`. Try setting the minimum size to `{value}`, and the maximum size to `{value}`.
- `TUNE0220W`: The Java virtual machine is spending a considerable amount of time in garbage collection. Consider increasing the heap size.

A complete list is available in the WebSphere Application Server V9 documentation in IBM Knowledge Center under **Reference > Messages > TUNE**.

IBM Training 

Performance and Diagnostic Advisor (5 of 5)

Runtime Events

Runtime Events > Message Details

Use this page to view runtime events that propagate from the server.

General Properties

Message

TUNE9001W: Heap utilization patterns indicate that you may have a memory leak
Additional explanatory data follows. Data values for free memory between 8/9/11 11:44 AM and 8/9/11 11:46 AM were consistently below minimum required percentage.

Message type

Runtime warning

Explanation

Over a period of time the amount of free memory appears to be decreasing or there is consistently insufficient free memory in the heap, indicating that you may have a memory leak.

User action

Use tooling to further analyze your memory usage over time. Refer to the information center for more information about diagnosing out-of-memory errors and Java heap memory leak.

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-37. Performance and Diagnostic Advisor (5 of 5)

The message is:

TUNE9001W: Heap utilization patterns indicate that you might have a memory leak. Additional explanatory data follows. Data values for free memory between 8/9/11 11:44 AM and 8/9/11 11:46 AM were consistently less than the minimum required percentage.

Explanation: Over time the amount of free memory seems to be decreasing or there is consistently insufficient free memory in the heap, indicating that you might have a memory leak.

User action: Use tools to further analyze your memory usage over time. For more information about diagnosing out-of-memory errors and Java heap memory leaks, see the IBM Knowledge Center.

Tivoli Performance Viewer advisor

- Performance advice:
 - ORB service thread pools
 - Web container thread pools
 - Connection pool size
 - Persisted session size and time
 - Prepared statement cache size
 - Session cache size
 - Dynamic cache size
 - Java virtual machine (JVM) heap size
 - DB2 Performance Configuration wizard

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-38. Tivoli Performance Viewer advisor

The performance advisor in Tivoli Performance Viewer provides advice on using collected Performance Monitoring Infrastructure (PMI) data to help tune systems for optimal performance and provide recommendations on inefficient settings. Obtain the advice by selecting the performance advisor in Tivoli Performance Viewer.

In a Network Deployment environment, the performance advisor in Tivoli Performance Viewer runs within the JVM of the node agent and can provide advice on resources that are more expensive to monitor and analyze. In a stand-alone application server environment, the performance advisor in Tivoli Performance Viewer runs within the application server JVM. The Tivoli Performance Viewer advisor requires that you enable performance modules, counters, or both.

Examples of performance advice

- Data sources
 - Situation: The prepared statement discard rate is too high, and heap space is available
 - Advice provided: Increase statement cache size
- Thread pools (ORB, web container, data source)
 - Situation: The number of connections is low (at the minimum)
 - Advice provided: Decrease pool size
 - Situation: All data source connections are heavily used, and heap space is available
 - Advice provided: Increase maximum pool size
 - Situation: The size of the pool is fluctuating a lot (high variance), possibly indicating batch processing, and wasted resources
 - Advice provided: Decrease pool size
- JVM heap size
 - Situation: Heap size is too small (less than 256 MB)
 - Advice provided: Increase the heap size to a value greater than 256 MB

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-39. Examples of performance advice

More examples of advice that the performance advisors would give in certain situations include:

- **Unbounded thread pools**
 - Situation: Threads added to an unbounded pool are not pooled.
 - Advice: If the average number of threads is higher than the pool size, then the pool must be increased to allow better pooling.
- **Sessions**
 - Situation: Read/write time or size is too large.
 - Advice: Warn of application problem.
 - Situation: The number of live sessions is greater than the session cache, and memory is available.
 - Advice: Increase session cache.
 - Situation: Requests are turned down because there is no room for new sessions.
 - Advice: Either too many active sessions are present, or the cache size is too small.
- **DB2 Performance Configuration wizard**

- Situation: A DB2 database is detected in the configuration.
- Advice: Use the DB2 Performance wizard to tune the indicated database.

Viewing performance advice

- In Tivoli Performance Viewer, click the Advisor link
- From the list of messages, click a link to see more detail
 - Messages can be sorted by severity



The screenshot shows the main content area of the Tivoli Performance Viewer. It displays a table of performance advice messages. The columns are 'Select', 'Severity', 'Message', and 'Status'. There are five rows of data, each with a checkbox in the 'Select' column and a severity level (Config or Warning) in the 'Severity' column. The 'Message' column contains links to detailed advice. The 'Status' column shows all entries as 'Unread'. At the bottom of the table, it says 'Page: 1 of 3' and 'Total 13'.

Select	Severity	Message	Status
<input type="checkbox"/>	Config	TUNE5003W: The JVM maximum heap size...	Unread
<input type="checkbox"/>	Config	TUNE5012W: The size of the minimum ...	Unread
<input type="checkbox"/>	Config	TUNE5042W: Enable servlet caching f...	Unread
<input type="checkbox"/>	Warning	TUNE0318I: There is no data availab...	Unread
<input type="checkbox"/>	Warning	TUNE0318I: There is no data availab...	Unread

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-40. Viewing performance advice

To view advice messages in Tivoli Performance Viewer, click the Advisor link.

From the list of messages, click a link to see more detail.



Performance advice detail

General Properties

Message
TUNE5042W: Enable servlet caching for better performance.

Severity
Config

Description
Servlet caching is not enabled.

User Action
To enable servlet caching in the administrative console, click Servers > Application servers > server_name > Web container settings > Web container and select Enable servlet caching under the Configuration tab. Click Apply or OK. You must restart your Application Server.

Detail
Currently, servlet caching is disabled.

[Back](#)

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-41. Performance advice detail

In this example, the message suggests enabling servlet caching for better performance. Servlet caching is a web container setting that is disabled by default. The User Action section in the advice details provides instructions for enabling servlet caching.

Performance advisor suggested practices

- Use only during stable production load tests
 - Application must remain stable during production tests
 - Any exceptions and deadlock issues must be resolved before running
 - The test load must be consistent
 - Varied load might lead to contradictory advice
- Enable after production load tests reach peak load levels
 - Exclude ramp-up and ramp-down times from monitoring
 - Increasing or decreasing loads might lead to contradictory advice
 - Certain types of advice are only generated when processor is being stressed (processor use > 50%)
- Important: tune your application before you tune WebSphere

[Performance monitoring](#)

© Copyright IBM Corporation 2016

Figure 10-42. Performance advisor suggested practices

When using the performance advisors, processor utilization must rise above 50% before advice is generated. Typically when running your production level load, you push the processor usage to 80–100% before turning on one of the performance advisors.

Consider the following when using a performance advisor for tuning:

If the load changes on the system under test, contradictory advice is generated. This behavior is because the collected PMI data shows a different type of environment, causing the advice to shift. To avoid this situation, always run the advisors while simulating the load WebSphere experiences during deployment (peak load).

If the pool size minimum and maximum values are the same, the performance advisor rules are much more likely to give contradictory advice when load fluctuates.

The amount of processor usage determines the amount of system activity. The advisors do not consider disk activity, network activity, memory usage, or other factors to get a more realistic view of system load.

Recommendations are generated only when processor load reaches 50% and higher.

Performance advisors from different application servers might give contradictory advice on the same node resources. This behavior is because the application servers take into account *only* how they are individually employing the resource. In this situation, if the advice from the different

advisors varies greatly, consider the generated advice and decide what changes to make. However, if all advisors are giving the same recommendations, then you must seriously consider the suggested changes.

If the performance advisor suggests setting a pool size to X, you must set the minimum value to X/2 and the maximum value to X.

If the performance advisor suggests setting the *prepared statement cache* value to a certain setting, check the amount of memory that is available before setting. The advisors do not consider the amount of actual physical memory available on the system.

Unit summary

- Describe performance monitoring and tuning methods
- Use the Tivoli Performance Viewer to monitor application server resources
- Use the performance servlet to generate performance data
- Configure the Request Metrics tool to generate performance data about the end-to-end request flow
- Use Performance Advisors to generate suggested tuning actions

Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-43. Unit summary

Review questions

1. What are the two performance data collection technologies in WebSphere?
2. Which WebSphere performance tool do you use to monitor overall system health?
3. True or False: The Performance Monitoring Infrastructure is enabled by default.
4. True or False: The Tivoli Performance Viewer Advisor tool generates tuning advice and automatically applies it to the environment.



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-44. Review questions

Write your answers here:

- 1.
- 2.
- 3.
- 4.

Review answers

1. What are the two performance data collection technologies in WebSphere?
The answer is: The Performance Monitoring Infrastructure (PMI) and request metrics provide the data collection mechanisms in WebSphere.
2. Which WebSphere performance tool do you use to monitor overall system health?
The answer is: The Tivoli Performance Viewer monitors overall system health.
3. True or False: The Performance Monitoring Infrastructure is enabled by default.
The answer is True.
4. True or False: The Tivoli Performance Viewer Advisor tool generates tuning advice and automatically applies it to the environment.
The answer is False. The performance advisor tools do not automatically tune the environment. You must tune manually and test the effect of the changes.



Performance monitoring

© Copyright IBM Corporation 2016

Figure 10-45. Review answers

Exercise: Using the performance monitoring tools

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 10-46. Exercise: Using the performance monitoring tools

Exercise objectives

After completing this exercise, you should be able to:

- Enable various levels of Performance Monitoring Infrastructure (PMI) statistics for an application server
- Monitor an application server by using Tivoli Performance Viewer
- Configure user settings for Tivoli Performance Viewer
- Examine summary reports and performance modules in Tivoli Performance Viewer
- View performance messages from the Tivoli Performance Viewer Advisor
- Enable and configure the Request Metrics tool
- View Request Metrics messages in the standard logs of an application server

Unit 11. Course summary

Estimated time

00:05

Overview

This unit summarizes the course and provides information for future study.

How you will check your progress

It is not necessary to check your progress on the summary.

References

WebSphere Application Server V9 documentation in IBM Knowledge Center, Monitoring, and Tuning Performance:

http://www.ibm.com/support/knowledgecenter/en/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 11-1. Unit objectives

Course objectives

- Relate WebSphere Application Server to the WebSphere family of products
- Describe the features and standards in WebSphere Application Server V9
- Describe the use of WebSphere Application Server in cloud, hybrid cloud, and on-premises environments
- Describe the architectural concepts that are related to WebSphere Application Server
- Assemble and install server-side Java enterprise applications
- Use WebSphere administrative tools to configure and manage enterprise applications
- Use wsadmin scripting
- Configure WebSphere Application Server security
- View performance information about server and application components
- Troubleshoot problems by using problem determination tools and log files

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 11-2. Course objectives

To learn more on the subject

- IBM Training website:
 - www.ibm.com/training
- IBM Redbooks:
 - www.redbooks.ibm.com
- IBM Knowledge Center for IBM WebSphere Application Server traditional V9:
 - https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/as_ditamaps_was900>Welcome_base.html
- IBM support:
 - <http://www.ibm.com/support/en-us/?lnk=hmmsu>
- Product information:
 - <http://www.ibm.com/software/products/en/appserv-was>
 - https://www.ibm.com/common/ssi>ShowDoc.wss?docURL=/common/ssi/rep_ca/4/877/ENUSZP16-0344/index.html&lang=en&request_locale=en

Course summary

© Copyright IBM Corporation 2016

Figure 11-3. To learn more on the subject

Enhance your learning with IBM resources

Keep your IBM Cloud skills up-to-date

- IBM offers resources for:
 - Product information
 - Training and certification
 - Documentation
 - Support
 - Technical information



- To learn more, see the IBM Cloud Education Resource Guide:
 - www.ibm.biz/CloudEduResources

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 11-4. Enhance your learning with IBM resources

Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 11-5. Unit summary

IBM Training

IBM

Course completion

You have completed this course:

WebSphere Application Server V9 Administration



Any questions?

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 11-6. Course completion

Appendix A. List of abbreviations

A

- ARM** Application Response Measurement
APAR authorized program analysis report
API application programming interface

B

- BSF** Bean Scripting Framework

C

- CaaS** capabilities as a service
CSlv2 Common Secure Interoperability Protocol Version 2
CORBA Common Object Request Broker Architecture

D

- DB** database
DD deployment descriptor
DMZ demilitarized zone
DTD document type definition

E

- EAR** enterprise archive
EE Enterprise Edition
EJB Enterprise JavaBean
EJS Enterprise Java Services

F

G

- GB** gigabyte
GC garbage collection
GCMV Garbage Collection and Memory Visualizer
GIF Graphics Interchange Format
GUI graphical user interface

H

- HA** high availability
HPEL High Performance Extensible Logging

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

HTTPS HTTP over SSL

I

IaaS infrastructure as a service

IBM International Business Machines Corporation

IDE integration development environment

IOP Internet Inter-ORB Protocol

I/O input/output

IP Internet Protocol

IPsec IP Security

ISC Integrated Solutions Console

ISP internet service provider

IT information technology

J

J2C Java EE Connector Architecture

JAAS Java Authentication and Authorization Service

JACC Java Authorization Contract for Containers

JAF JavaBeans Activation Framework

JAR Java archive

Java EE Java Platform, Enterprise Edition

JAXB Java Architecture for XML Binding

JAXP Java API for XML Processing

JAXR Java API for XML Registries

JAX-RPC Java API for XML Remote Procedure Calls

JAX-RS Java API for XML-based Remote Procedure Calls

JAX-WS Java API for XML Web Services

JCA Java EE Connector Architecture

JCE Java Cryptology Extension

JDBC Java Database Connectivity

JDK Java Development Kit

JIT just-in-time

JMS Java Message Service

JMX Java Management Extensions

JNDI Java Naming and Directory Interface

JPA Java Persistence API

JPG graphics file type or extension (lossy compressed 24-bit color image storage format developed by the Joint Photographic Experts Group)

JSF JavaServer Faces
JSP JavaServer Pages
JSR Java Specification Request
JSTL JavaScript Tag Library
JTA Java Transaction API
JVM Java virtual machine

K

KVM kernel-based virtual machine

L

LDAP Lightweight Directory Access Protocol
LTPA Lightweight Third Party Authentication

M

MAP message addressing property
MB megabyte
MBean Managed Bean (Managed Java object)
MQ Message Queue

N

ND network deployment (IBM Liberty WebSphere Application Server Network Deployment)
NT new technology

O

OOM out-of-memory
ORB Object Request Broker
OS operating system
OSGi Open Service Gateway initiative (originally)

P

PaaS platform as a service
PBW Plants By WebSphere
PDF Portable Document Format
PFBC properties file-based configuration
PHD portable heap dump
PID process identifier
PME programming model extensions
PMI Performance Monitoring Infrastructure
PMR problem management record
PMRM Privacy Management Reference Model and Methodology

PMT Program Management Tool

POJO plain old Java object

Q

R

RAR resource archive

RC return code

RM request metrics

RMI Remote Method Invocation

S

SAAJ SOAP with attachments API for Java

SAM Security Access Manager

SAR SIP application resource

SAS Secure Association Service

SDO Service Data Objects

SIB or SIBus service integration bus

SIP Session Initiation Protocol

SOA service-oriented architecture

SOAP SOAP is not an acronym; it is a word in itself (formerly an acronym for Simple Object Access Protocol)

SQL Structured Query Language

SSL Secure Sockets Layer

SSO single sign-on

StAX Streaming API for XML

SVC supervisor call

SVG Scalable Vector Graphics

T

TCP Transmission Control Protocol

TCP/IP Transmission Control Protocol/Internet Protocol

U

UDDI Universal Description, Discovery, and Integration

UNIX Uniplexed Information and Computing System

UOW unit of work

URI Uniform Resource Identifier

URL Uniform Resource Locator

V

VMM virtual member manager

VPN virtual private network

W

WAR web archive

WLM workload management

WS web services

WS-AT Web Services Atomic Transaction

WS-BA Web Services Business Activity

WS-COOR Web Services Coordination

WS-I Web Services Interoperability

WSIF Web Services Invocation Framework

WSDL Web Services Description Language

X

XA extended architecture

XCT Cross Component Trace

XML Extensible Markup Language

XTP extreme transaction processing

Y

Z

z/OS zSeries operating system

Appendix A. Resource guide

Completing this IBM Training course is a great first step in building your IBM Middleware skills. Beyond this course, IBM offers several resources to keep your Middleware skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
 - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
 - For more information, see: <http://www.ibm.com/training>
- **IBM Training News**
 - Review or subscribe to updates from IBM and its training partners.
 - For more information, see: ibm.com/blogs/ibm-training
- **IBM Certification**
 - Demonstrate your mastery of IBM Middleware to your employer or clients through IBM Professional Certification. Middleware certifications are available for developers, administrators, and business analysts.
 - For more information, see: <http://www.ibm.com/certify>
- **Training paths**
 - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many IBM products and roles, including developers and administrators.
 - For more information, see:
<http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=a003096>

Social media links

Connect with IBM Middleware Education and IBM Training, and learn about the newest courses, certifications, and special offers by seeing any of the following social media websites.

- **Twitter**
 - Receive concise updates from Middleware Education a few times each week.
 - Follow Middleware Education at: twitter.com/IBMCLOUDedu

- **Facebook:**

- Follow IBM Training on Facebook to keep in sync with the latest news and career trends, and to post questions or comments.
- Find IBM Training at: facebook.com/ibmtraining

- **YouTube:**

- See the IBM Training YouTube channel to learn about IBM training programs and courses.
- Find IBM Training at: youtube.com/IBMTTraining

Support

- **Middleware Support portal**

- The Middleware Support website provides access to a portfolio of downloadable support tools, including troubleshooting utilities, product updates, drivers, and authorized program analysis reports (APARs). The Middleware Support website also provides links to online Middleware communities and forums for collaboratively solving issues. You can now customize the IBM Support website by adding or deleting portlets to show the most important information for the IBM products that you work with.
- For more information, see: <http://www.ibm.com/software/websphere/support>

- **IBM Support Assistant**

- The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
- IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
- For more information, see: <http://www.ibm.com/software/support/isa>

- **IBM Education Assistant**

- IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your IBM Training course.
- For more information, see: <http://www.ibm.com/software/info/education/assistant/>

Middleware documentation and tips

- **IBM Redbooks**

- The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe

installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many Middleware products. Often, Redbooks include sample code and other support materials available as downloads from the site.

- For more information, see: <http://www.ibm.com/redbooks>
- **IBM documentation and libraries**
 - IBM Knowledge Centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The IBM Knowledge Centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain Middleware products. The IBM Knowledge Center and library are located conveniently in the left navigation on product web pages.
- **developerWorks**
 - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the Middleware zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
 - For more information, see: <http://www.ibm.com/developerworks>

Services

- IBM Software Services for Middleware are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The Middleware Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for Middleware makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.
- For more information, see:
<http://www.ibm.com/services/us/en/it-services/systems/middleware-services/>



IBM Training



© Copyright International Business Machines Corporation 2016.