

**WA2097 WebSphere  
Application Server 8.5  
Administration on Linux**



Web Age Solutions Inc.  
USA: 1-877-517-6540  
Canada: 1-866-206-4644  
Web: <http://www.webagesolutions.com>

The following terms are trademarks of other companies:

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

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

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

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

IBM, WebSphere, DB2 and Tivoli are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

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

For customizations of this book or other sales inquiries, please contact us at:

USA: 1-877-517-6540, email: [getinfousa@webagesolutions.com](mailto:getinfousa@webagesolutions.com)

Canada: 1-866-206-4644 toll free, email: [getinfo@webagesolutions.com](mailto:getinfo@webagesolutions.com)

Copyright © 2013 Web Age Solutions Inc.

This publication is protected by the copyright laws of Canada, United States and any other country where this book is sold. Unauthorized use of this material, including but not limited to, reproduction of the whole or part of the content, re-sale or transmission through fax, photocopy or e-mail is prohibited. To obtain authorization for any such activities, please write to:

Web Age Solutions Inc.  
439 University Ave  
Suite 820  
Toronto  
Ontario, M5G 1Y8

## Table of Contents

Chapter 1 - Course Introduction.....	15
1.1 Course Description.....	15
1.2 Course Prerequisites.....	15
1.3 Course Objectives.....	16
1.4 Course Objectives.....	16
1.5 Introductions.....	16
1.6 Agenda – Day 1.....	17
1.7 Agenda – Day 2.....	17
1.8 Agenda – Day 3.....	18
1.9 Agenda – Day 4.....	18
1.10 Agenda – Day 5.....	18
Chapter 2 - Java EE and WebSphere Overview.....	19
2.1 Goals of Enterprise Applications.....	19
2.2 What is Java?.....	20
2.3 What is Java EE?.....	21
2.4 The Java EE Specifications.....	22
2.5 Role of Application Server.....	22
2.6 What is a Servlet?.....	24
2.7 Servlet Execution.....	24
2.8 What is a JSP?.....	25
2.9 JSP Code Sample.....	26
2.10 What is an EJB?.....	26
2.11 EJB Types.....	27
2.12 EJB Examples.....	28
2.13 Model-View-Controller Architecture.....	29
2.14 MVC – An Example.....	30
2.15 WebSphere Application Server Platform.....	31
2.16 "Base" Edition.....	31
2.17 "Base" Edition.....	32
2.18 Other WAS Editions.....	32
2.19 Other WAS Editions.....	33
2.20 IBM HTTP Server.....	33
2.21 Other Products.....	34
2.22 Java EE and WebSphere Overview Questions.....	34
2.23 Java EE and WebSphere Overview Answers.....	35
2.24 Reference.....	35
Chapter 3 - WebSphere Installation with Installation Manager.....	37
3.1 General System Requirements.....	37
3.2 Pre-Install Tasks.....	37
3.3 Pre-Install Tasks.....	38
3.4 Installation Manager.....	39
3.5 Product Repositories.....	39
3.6 Silently Installing Installation Manager.....	40
3.7 Recording Response Files With IM.....	41

3.8 Installation Manager Console Mode.....	41
3.9 Installing Products Silently With Response Files.....	42
3.10 WebSphere Profiles.....	42
3.11 Benefits of Profiles.....	43
3.12 Creating Profiles.....	43
3.13 Profile Management Tool.....	44
3.14 Profile Management Tool.....	45
3.15 manageprofiles command.....	45
3.16 manageprofiles command.....	46
3.17 Product Files Directories.....	47
3.18 Profile Directories.....	48
3.19 Cell/Node/Server.....	48
3.20 Config Directory.....	50
3.21 Important Configuration Files.....	50
3.22 Starting and Stopping the Server.....	51
3.23 Application Server Installation Questions.....	52
3.24 Application Server Installation Answers.....	52
3.25 Reference.....	53
3.26 Lab Summary.....	53
3.27 Common Lab Errors.....	54
Chapter 4 - Administration Console and Application Deployment.....	55
4.1 Single Server Admin Structure.....	55
4.2 Administration Clients.....	55
4.3 Logging In.....	56
4.4 Logging In.....	57
4.5 Console Overview.....	58
4.6 Navigation Tree.....	59
4.7 Workspace Area.....	60
4.8 Configuration and Runtime.....	60
4.9 Help Area.....	61
4.10 Save Configuration.....	61
4.11 Save Configuration.....	62
4.12 Save Conflicts.....	63
4.13 Configuration Repository Checkpoints.....	64
4.14 Common Configuration.....	65
4.15 Scope.....	65
4.16 Application Installation.....	66
4.17 Installation Steps.....	67
4.18 Installation Steps.....	67
4.19 Finishing Up.....	68
4.20 Start Application.....	69
4.21 Installing Applications From a Monitored Directory.....	69
4.22 Applications on File System.....	71
4.23 Admin Console & Application Deployment Questions.....	71
4.24 Admin Console & Application Deployment Answers.....	71
4.25 Reference.....	72

4.26 Lab Summary.....	72
4.27 Common Lab Errors.....	72
Chapter 5 - Web Server and Web Server Plug-in.....	73
5.1 Using a Web Server.....	73
5.2 Web Server Plug-in Architecture.....	74
5.3 Web Server Integration.....	75
5.4 IBM HTTP Server Installation.....	75
5.5 IBM HTTP Server Configuration.....	76
5.6 Plug-in Configuration.....	77
5.7 Plug-in Configuration.....	78
5.8 Plug-in Configuration Scenarios.....	79
5.9 IBM HTTP Server Services.....	80
5.10 Web Servers in WebSphere.....	80
5.11 Defining Web Servers.....	81
5.12 Mapping Applications to Web Servers.....	82
5.13 Plug-in Administration.....	82
5.14 Plug-in Administration.....	83
5.15 Plug-in Tuning.....	85
5.16 Virtual Host.....	86
5.17 Virtual Hosts in WebSphere.....	86
5.18 Creating a Virtual Host in WebSphere.....	87
5.19 Relation to Enterprise Applications.....	88
5.20 Plug-in Syntax.....	88
5.21 Plug-in Syntax.....	89
5.22 Web Server & Web Server Plug-in Questions.....	90
5.23 Web Server & Web Server Plug-in Answers.....	90
5.24 Lab Summary.....	90
5.25 Common Lab Errors.....	91
Chapter 6 - Troubleshooting.....	93
6.1 Problem Determination.....	93
6.2 Error Reporting.....	93
6.3 Browser to Web Server Connection Problem.....	94
6.4 HTTP Request Problems.....	95
6.5 HTTP Request Solutions.....	95
6.6 Tools for Resolving WebSphere Problems.....	97
6.7 Admin Console Messages.....	97
6.8 Configuration Problems.....	97
6.9 Runtime Messages.....	98
6.10 Message Details.....	99
6.11 Basic and HPEL Logging Mode.....	100
6.12 Important Log Files.....	100
6.13 Important Log Files.....	101
6.14 Configuring JVM Log Files.....	102
6.15 Viewing JVM Logs.....	102
6.16 Log Format.....	103
6.17 Event Types in JVM Log.....	104

6.18 Event Types in JVM Type.....	105
6.19 Setting the Log Levels.....	106
6.20 Level Specification String.....	107
6.21 Tracing.....	107
6.22 Configuring Tracing.....	108
6.23 Reading Java Exceptions.....	109
6.24 Common Exceptions.....	110
6.25 Server Does Not Start.....	110
6.26 dumpNameSpace Tool.....	110
6.27 Remote Debugging.....	111
6.28 Thread Dump.....	112
6.29 IBM Tools.....	114
6.30 Troubleshooting Questions.....	115
6.31 Troubleshooting Answers.....	115
6.32 Reference.....	115
6.33 Lab Summary.....	116
6.34 Common Lab Errors.....	116
Chapter 7 - WebSphere 8 HPEL Logging.....	117
7.1 New HPEL Logging Feature.....	117
7.2 Benefits of HPEL.....	117
7.3 HPEL Log Repository Architecture.....	118
7.4 HPEL Logging Configuration.....	119
7.5 HPEL Trace Configuration.....	119
7.6 Configuration of Log Detail Level With HPEL.....	120
7.7 Log Repository Location.....	120
7.8 Viewing HPEL Log Records in Admin Console.....	121
7.9 Viewing HPEL Log Records by Command Line.....	122
7.10 HPEL Text Log.....	123
7.11 Summary.....	123
7.12 Lab Summary.....	123
7.13 Common Lab Errors.....	124
Chapter 8 - IBM Support Assistant.....	125
8.1 Source of Errors.....	125
8.2 Support Tools.....	125
8.3 IBM Support Assistant.....	126
8.4 ISA Workbench.....	127
8.5 Customizing the Workbench.....	127
8.6 ISA Preferences.....	128
8.7 Search for Information.....	129
8.8 Case Management.....	130
8.9 Collecting Information.....	130
8.10 Guided Troubleshooting.....	131
8.11 Troubleshooting Tools.....	132
8.12 Manage Service Requests.....	132
8.13 ISA Lite.....	133
8.14 Summary.....	134

Chapter 9 - JDBC and Data Sources.....	137
9.1 What is JDBC?.....	137
9.2 JDBC Driver.....	137
9.3 Typical JDBC Use.....	138
9.4 Problems to Address.....	138
9.5 JDBC Data Sources.....	139
9.6 Data Source Architecture.....	139
9.7 Typical Use of Data Source.....	140
9.8 Configuring JDBC in WebSphere.....	141
9.9 Creating JDBC Provider.....	141
9.10 Creating JDBC Provider.....	142
9.11 Using WebSphere Variables.....	142
9.12 Creating Data Source.....	143
9.13 Connection Pools.....	144
9.14 Advanced DB2 Settings.....	144
9.15 Data Source Resource Reference.....	145
9.16 JPA and Data Source.....	146
9.17 JPA Administration.....	146
9.18 JPA Resource Reference.....	147
9.19 Logging JPA SQL Statements.....	148
9.20 Data Source Configuration Errors.....	149
9.21 JDBC & Data Sources Questions.....	149
9.22 JDBC & Data Sources Answers.....	149
9.23 Reference.....	150
9.24 Lab Summary.....	150
9.25 Common Lab Errors.....	150
Chapter 10 - JMS Overview and WebSphere MQ Configuration.....	151
10.1 Messaging Introduction.....	151
10.2 Messaging Example.....	151
10.3 Messaging Components.....	152
10.4 Messaging Types.....	153
10.5 JMS (Java Messaging Service).....	153
10.6 Message Driven Beans (MDB).....	154
10.7 MDB as Message Consumer.....	154
10.8 JMS Resources.....	155
10.9 Messaging in WebSphere.....	156
10.10 Default Messaging.....	156
10.11 WebSphere MQ.....	157
10.12 Which To Use?.....	157
10.13 WebSphere MQ JMS Configuration.....	158
10.14 WebSphere MQ JMS Provider.....	159
10.15 Connection Factory.....	159
10.16 Destination.....	160
10.17 Activation Specification.....	161
10.18 Binding MDB to Activation Specification.....	162
10.19 Relationship Between Messaging Components.....	163

10.20 JMS Messaging Configuration Questions.....	163
10.21 JMS Messaging Configuration Answers.....	164
Chapter 11 - JMS Using Default Messaging Provider.....	165
11.1 Service Integration Bus.....	165
11.2 Service Integration Bus.....	165
11.3 Service Integration Bus.....	166
11.4 Service Integration Bus.....	166
11.5 Bus Members.....	167
11.6 Messaging Engine.....	167
11.7 Message Store.....	167
11.8 Destination.....	168
11.9 Default Messaging JMS Configuration.....	168
11.10 Creating a Service Integration Bus.....	169
11.11 Adding a Bus Member.....	169
11.12 Adding a Bus Member.....	170
11.13 Adding a Bus Member.....	170
11.14 Bus Member Message Store.....	171
11.15 Add a Destination.....	172
11.16 Default Messaging JMS Provider.....	172
11.17 JMS Connection Factory.....	173
11.18 JMS Destination.....	174
11.19 JMS Activation Specification.....	175
11.20 MDB Listener Binding.....	175
11.21 Relationship Between Messaging Components.....	176
11.22 JMS Messaging Configuration Questions.....	176
11.23 JMS Messaging Configuration Answers.....	177
11.24 Resources.....	177
11.25 Lab Summary.....	177
11.26 Common Lab Errors.....	178
Chapter 12 - WebSphere Security.....	179
12.1 Security Overview.....	179
12.2 Architecture Components.....	180
12.3 Security Components.....	180
12.4 Digital Certificates.....	181
12.5 SSL (Secure Sockets Layer).....	182
12.6 SSL in WebSphere.....	183
12.7 WebSphere SSL Configuration.....	184
12.8 Java Security.....	185
12.9 JAAS.....	185
12.10 Java EE Security.....	186
12.11 Authentication and Authorization.....	186
12.12 Challenge Mechanism.....	187
12.13 Configure Challenge Mechanism.....	187
12.14 User Registry.....	188
12.15 Federated Repositories.....	189
12.16 Security Configuration.....	190



12.17 User Registry Configuration.....	191
12.18 Authentication Mechanism Configuration.....	191
12.19 LTPA and Single Sign On Mechanism.....	193
12.20 Configuring LTPA.....	194
12.21 Single Sign-on (SSO).....	195
12.22 Admin Console Roles.....	195
12.23 Fine Grained Administration Security.....	197
12.24 Security Domains.....	198
12.25 Stopping Secure Servers.....	199
12.26 WebSphere Security Questions.....	200
12.27 WebSphere Security Answers.....	200
12.28 Lab Summary.....	200
12.29 Common Lab Errors.....	200
Chapter 13 - Java EE Security.....	201
13.1 Authorization.....	201
13.2 Java EE Roles.....	201
13.3 Sample Role Mapping.....	202
13.4 Mapping Roles to Users/Groups.....	202
13.5 Map Users/Groups.....	203
13.6 Special Roles.....	203
13.7 Using Security Domains.....	204
13.8 Defining Security Roles.....	205
13.9 Resource Protection in Web Module.....	205
13.10 Resource Protection in EJB Module.....	206
13.11 Unprotected Methods.....	207
13.12 "Unchecked" Method Permissions.....	208
13.13 Run-As Delegation.....	208
13.14 Java EE Security Questions.....	209
13.15 Java EE Security Answers.....	210
13.16 Lab Summary.....	210
13.17 Common Lab Errors.....	210
Chapter 14 - Web Service Administration.....	213
14.1 What is a Web Service?.....	213
14.2 SOAP Overview.....	214
14.3 WSDL Overview .....	214
14.4 WSDL Syntax Overview.....	215
14.5 Service Endpoint Definition.....	215
14.6 Web Service Administration Tasks.....	216
14.7 Configure Endpoint URL in Provider Side.....	217
14.8 Publishing WSDL File.....	218
14.9 Configure Endpoint URL in Consumer Side.....	219
14.10 Working with Policy Sets.....	220
14.11 Stopping a Service Listener.....	222
14.12 REST Web Services.....	222
14.13 Chapter Questions.....	223
14.14 Chapter Answers.....	224

14.15 Lab Summary.....	224
14.16 Common Lab Errors.....	224
Chapter 15 - Performance Management.....	227
15.1 What Affects Performance.....	227
15.2 How to Measure Performance?.....	228
15.3 WebSphere Queuing Network.....	229
15.4 Performance Tuning.....	230
15.5 System Settings for Tuning.....	231
15.6 Top Tuning Parameters.....	231
15.7 Top Tuning Parameters.....	232
15.8 Performance Monitoring Service.....	232
15.9 Enabling Performance Monitoring.....	233
15.10 Performance Statistics.....	234
15.11 Performance Data Available.....	234
15.12 Performance Advisors.....	235
15.13 Performance and Diagnostic Advisor.....	237
15.14 Tivoli Performance Viewer.....	238
15.15 Launching Tivoli Performance Viewer.....	239
15.16 Launching Tivoli Performance Viewer.....	240
15.17 Tivoli Performance Viewer Advisor.....	240
15.18 Summary Reports.....	241
15.19 Performance Data Options.....	242
15.20 Viewing Performance Data.....	242
15.21 Viewing Performance Data.....	243
15.22 Logging Performance Data.....	244
15.23 Performance Servlet.....	245
15.24 Performance Questions.....	245
15.25 Performance Answers.....	245
15.26 Lab Summary.....	246
15.27 Common Lab Errors.....	246
Chapter 16 - Java Memory Management.....	247
16.1 Java Heap Space.....	247
16.2 Heap Size Limits.....	248
16.3 Default Size Limits.....	250
16.4 Garbage Collection Basics.....	250
16.5 Allocation Failure (AF).....	251
16.6 OutOfMemoryError.....	252
16.7 Memory Leak.....	253
16.8 Free Space Threshold.....	254
16.9 Garbage Collection Mechanism.....	254
16.10 GC Performance Improvements.....	257
16.11 Generational Garbage Collection.....	257
16.12 Thread Local Heap (TLH).....	258
16.13 Memory Tuning Goals.....	258
16.14 Steady State Heap Size.....	260
16.15 Analyzing Steady State Behavior.....	261

16.16 Effect of Initial Heap Size on GC.....	263
16.17 Optimizing GC Duration.....	265
16.18 Summary.....	265
Chapter 17 - Memory Management in WebSphere.....	267
17.1 Configuring the Heap.....	267
17.2 Garbage Collection Policy.....	268
17.3 Monitoring Memory Using PMI.....	268
17.4 Heap Dumps.....	269
17.5 Analysis Using IBM Support Assistant.....	270
17.6 Leak Advisor.....	271
17.7 IBM Java SDK v7.....	272
17.8 Summary.....	272
17.9 Lab Summary.....	273
17.10 Common Lab Errors.....	273
Chapter 18 - Command Line Administration.....	275
18.1 Command Line Tools.....	275
18.2 Common Binary Commands.....	275
18.3 wsadmin.....	276
18.4 wsadmin Options.....	276
18.5 Benefits of wsadmin.....	277
18.6 Properties.....	278
18.7 Security.....	278
18.8 Scripting Languages.....	278
18.9 JMX in WebSphere.....	279
18.10 Default Scripting Objects.....	280
18.11 Getting Help.....	281
18.12 AdminConfig.....	281
18.13 Object ID.....	282
18.14 Saving Configuration.....	283
18.15 AdminControl.....	284
18.16 Object Names.....	285
18.17 AdminApp.....	285
18.18 AdminTask.....	287
18.19 AdminTask exportWasprofile.....	287
18.20 Common Scripting Patterns.....	288
18.21 Common Scripting Patterns.....	289
18.22 Common Scripting Patterns.....	289
18.23 Profiles.....	290
18.24 Things to Remember.....	291
18.25 Command Line Administration Questions.....	291
18.26 Command Line Administration Answers.....	292
18.27 Reference.....	292
18.28 Lab Summary.....	293
18.29 Common Lab Errors.....	293
Chapter 19 - Jython Scripting Tools.....	295
19.1 Managing Configurations with Properties Files.....	295

19.2 Exporting Properties File.....	296
19.3 Filtering Exported Properties.....	296
19.4 Options When Exporting Properties.....	297
19.5 Property File Templates.....	298
19.6 Validate a Properties File.....	299
19.7 Importing a Properties File.....	299
19.8 Scripting in RAD.....	300
19.9 Script Configuration.....	300
19.10 Jython Command Assistance in Admin Console.....	301
19.11 Command Assistance Notifications.....	302
19.12 Jython Script Library.....	303
19.13 AdminServerManagement.....	304
19.14 AdminApplication.....	305
19.15 AdminJDBC.....	307
19.16 Scripting Tool Questions.....	308
19.17 Scripting Tool Answers.....	308
19.18 Lab Summary.....	309
19.19 Common Lab Errors.....	309
Chapter 20 - Network Deployment Architecture.....	311
20.1 Workload Management (WLM) Basics.....	311
20.2 Role of Network Deployment Software.....	311
20.3 Similarities With Base Edition.....	312
20.4 Differences With Base Edition.....	312
20.5 Coexistence With Base Edition.....	313
20.6 Network Deployment Cell.....	314
20.7 Deployment Manager and Node Agents.....	314
20.8 Configuration File Synchronization.....	315
20.9 Configuration File Synchronization.....	316
20.10 Synchronization Problems.....	316
20.11 Firewall Issues.....	317
20.12 WebSphere Profiles - Recap.....	318
20.13 Network Deployment Profiles.....	318
20.14 Federating Nodes.....	319
20.15 Federating Nodes.....	319
20.16 Federating an Application Server Profile.....	321
20.17 Federating a Custom Profile.....	321
20.18 Prerequisites Before Adding Nodes.....	322
20.19 Steps Taken When Adding a Node.....	323
20.20 Recovering an Existing Node.....	323
20.21 Removing Nodes.....	325
20.22 Admin Console Changes.....	326
20.23 Controlling Deployment Manager and Node Agents.....	326
20.24 Automatically Starting Servers.....	327
20.25 Network Deployment Architecture Questions.....	328
20.26 Network Deployment Architecture Answers.....	328
20.27 Lab Summary.....	329

20.28 Common Lab Errors.....	329
20.29 Lab Topology.....	330
Chapter 21 - Web Server Management and Cluster Topology.....	331
21.1 Web Server Management.....	331
21.2 Administering IBM HTTP Server.....	332
21.3 IBM HTTP Server Architecture.....	333
21.4 Non-IHS Web Server.....	333
21.5 Managing Multiple Web Servers.....	334
21.6 Cluster.....	335
21.7 Vertical Scaling.....	336
21.8 Horizontal Scaling.....	337
21.9 "Mixed" Scaling.....	338
21.10 Heterogeneous Scaling.....	338
21.11 Creating a Cluster.....	339
21.12 First Cluster Member Options.....	340
21.13 Cluster Member Options.....	342
21.14 Cluster Member Options.....	343
21.15 Managing Clusters.....	343
21.16 Mapping Applications to Clusters.....	344
21.17 Cluster Topology Screen.....	345
21.18 Topology.....	345
21.19 Factors Affecting Topology.....	346
21.20 Factors Affecting Topology.....	346
21.21 Coexistence Scenarios.....	347
21.22 Common Topologies.....	347
21.23 All in One.....	348
21.24 Vertical Scaling.....	349
21.25 Server Separation.....	350
21.26 "Mixed" Scaling.....	351
21.27 Web Server Horizontal Scaling.....	352
21.28 Web Server Management & Cluster Topology Questions.....	352
21.29 Web Server Management & Cluster Topology Answers.....	353
21.30 Reference.....	353
21.31 Lab Summary.....	353
21.32 Common Lab Errors.....	353
21.33 Lab Topology.....	354
Chapter 22 - Workload Management and Application State.....	355
22.1 Types of WLM.....	355
22.2 Web Server WLM.....	355
22.3 Web Module WLM Architecture.....	356
22.4 Web Module WLM Architecture.....	357
22.5 Weighted Routing.....	357
22.6 State Management.....	358
22.7 Server Affinity.....	359
22.8 Weighted Routing with Server Affinity.....	359
22.9 EJB WLM.....	360

22.10 Factors Overriding EJB WLM.....	361
22.11 Problems With Server Affinity.....	361
22.12 Server Affinity With Failover.....	362
22.13 Data Replication Service.....	362
22.14 Number of Replicas.....	363
22.15 Enabling Server Affinity Failover.....	363
22.16 Enabling Server Affinity Failover.....	364
22.17 Session Database.....	365
22.18 Session Tuning.....	366
22.19 Distributed Session Tuning.....	367
22.20 Workload Management & Application State Questions.....	367
22.21 Workload Management & Application State Answers.....	368
22.22 Lab Summary.....	368
22.23 Common Lab Errors.....	368
Chapter 23 - Overview of WebSphere Intelligent Management Features.....	369
23.1 Challenges to Address.....	369
23.2 Evolution of WAS ND 8.5.....	369
23.3 WebSphere 8.5 “Intelligent Management”.....	370
23.4 “On Demand Router”.....	370
23.5 Intelligent Routing.....	371
23.6 Service Policies.....	371
23.7 Adjusting for Capacity.....	372
23.8 Dynamic Clustering.....	373
23.9 Application Edition Management.....	373
23.10 Edition Control Center.....	374
23.11 Edition Management Use Cases.....	374
23.12 Concurrent Activation.....	375
23.13 Application Validation.....	375
23.14 Application Rollout.....	376
23.15 Health Management.....	376
23.16 Issues Solved by Health Management.....	377
23.17 Summary.....	377
23.18 Reference.....	378
Chapter 24 - Course Summary.....	379
24.1 WebSphere Certification.....	379
24.2 Reference.....	380

# Chapter 1 - Course Introduction

---

## ***Objectives***

Key objectives of this chapter

- Course Description
- Course Prerequisites
- Course Objectives
- Introductions
- Agenda

## **1.1 Course Description**

- This course teaches the basics of the deployment and administration of applications to a WebSphere Application Server 8.5 environment
- This course describes the following editions of WebSphere
  - ◇ Base edition - single-server environment
  - ◇ Network Deployment - clustered environment
- This course contains extensive hands-on labs

## **1.2 Course Prerequisites**

- To get the most out of this course students are expected to meet the following prerequisites:
  - ◇ Understanding of basic Internet concepts
  - ◇ Experience using a Web browser
  - ◇ Basic Linux operational skills
- Although not required, any experience with the following can enhance student experience:
  - ◇ Web servers and web-based applications
  - ◇ Java programming and technology

### **1.3 Course Objectives**

- After completing this course, students will be able to:
  - ◇ Install WebSphere Application Server
  - ◇ Install and configure Java EE applications
  - ◇ Configure a web server to forward requests to WebSphere
  - ◇ Use various tools to solve several types of problems
  - ◇ Configure various application resources
  - ◇ Configure WebSphere Security

### **1.4 Course Objectives**

- After completing this course, students will be able to:
  - ◇ Administer the environment using the web-based Admin Console or command line tools
  - ◇ Monitor WebSphere performance and understand what affects WebSphere performance
  - ◇ Use various techniques to create a more robust environment and avoid common problems
  - ◇ Create and configure a cluster of Application Servers with Network Deployment
  - ◇ Configure appropriate topologies of web servers and Application Servers

### **1.5 Introductions**

- Instructor
- Students
  - ◇ Please introduce yourself
  - ◇ Java and Java EE knowledge
  - ◇ Prior WebSphere knowledge



- ◇ Personal goals for the class

## **1.6 Agenda – Day 1**

- Course Intro
- WebSphere Overview
- Installation
  - ◇ Lab: WebSphere Installation
- Console Introduction & Application Deployment
  - ◇ Lab: Install an Application
- Web Server & Web Server Plugin
  - ◇ Lab: IBM HTTP Server & Plug-in
- Troubleshooting Tools
  - ◇ Lab: Solving Problems

## **1.7 Agenda – Day 2**

- HPEL Logging
  - ◇ Lab: HPEL logging
- IBM Support Assistant
- JDBC & Data Sources
  - ◇ Lab: Install Database Application
- JMS Intro & WebSphere MQ
- JMS Messaging with Default Messaging
  - ◇ Lab: Enable JMS Messaging (WebSphere MQ or Default Messaging)
- Security Configuration
  - ◇ Lab: Configure WebSphere Security

## **1.8 Agenda – Day 3**

- Java EE Security
  - ◇ Lab: Configure Java EE Security
- Web Services Administration
  - ◇ Lab: Web Services
- Performance Monitor
  - ◇ Lab: Monitor Performance
- Java Memory Management
  - ◇ Lab: WebSphere Memory

## **1.9 Agenda – Day 4**

- Command Line Admin
  - ◇ Lab: Use wsadmin Scripting
- Jython Scripting Tools and Assistance
  - ◇ Lab: Additional Jython Scripting
- Network Deployment Architecture
  - ◇ Lab: Deployment Manager Profile
  - ◇ Lab: Configure Cell Architecture

## **1.10 Agenda – Day 5**

- Web Server and Cluster Topology
  - ◇ Lab: Configure Cluster and Remote Web Server
- Workload Management and Application State
  - ◇ Lab: Install Clustered Application
- Intelligent Management
- Course Summary

## Chapter 2 - Java EE and WebSphere Overview

---

### **Objectives**

At the end of this chapter, participants will be able to understand the:

- Basic Concepts of Java EE
- Role of a Java EE Application Server
- Products of WebSphere Platform
- Features of WebSphere Application Server

### **2.1 Goals of Enterprise Applications**

- Provide secure access to data and functionality from many types of client devices
- Integrate existing systems
- Provide an infrastructure that can easily scale to meet increased demand
- Reduce cost of development and maintenance to provide increased ROI
  - ◇ Allow new features to be added without major redesign of an application

### **Enterprise Applications**

What are enterprise applications? These are software systems that run the day to day operations of a business. Currently, most businesses use "best of breed" solutions for their problems. That means, separate software applications are used to solve specific problems. For example, an application runs the online e-commerce site. A separate application is run in the warehouse to manage inventory. Accounting and sales automation is done by yet another software.

These enterprise applications have to meet a set of common demands. They must provide secure access to their data and functionality. The GUI front end for these applications may be remotely located in different machines than the application itself. This calls for client server computing. They must perform well within expectation. They must be able to scale up to meet increasing demand for activity (as the business grows). Ideally, the scaling should be as painless as possible. This requires specific scaling support from the applications.

The fact that a business uses many enterprise applications means, the same data may be duplicated in many of them. Synchronizing the data across the business becomes a problem. Enterprise applications, as a result, should be able to integrate well with other applications.

Java EE is a set of standards that attempt make it easier for applications to meet these needs.

Throughout this class we will learn various techniques provided by Java EE and WebSphere Application Server that will help you develop and deploy first class enterprise applications.

Most businesses buy off the shelf applications and then customize them to fit their specific need. The software architecture should be such that new features can be added by surgically making changes to the application. This is in the realm of good software design and mostly out of the scope of this class.

## 2.2 What is Java?

- Java is an Object-Oriented programming language
- Business logic code is written in classes.
- Classes are compiled in a machine neutral binary format called byte code.
- At execution time, a virtual machine converts the byte code into machine specific executable code which is then executed by the OS and the CPU.
  - ◇ This allows the same byte code to be executed in many different OS.
- Java comes in various editions
  - ◇ Java Micro Edition (Java ME)
  - ◇ Java Standard Edition (Java SE).
  - ◇ Java Enterprise Edition (Java EE) is the focus of this class.

### What is Java?

One of the allures of Java to the programmers is that it makes it easy to develop applications and run it in many different operation systems (OS) and hardware. Every hardware platform and OS uses a different format to store executable code. The format depends on how the OS and the hardware (CPU) are designed. To solve this problem, Java depends on an intermediary executable format called byte code. This format does not depend on any specific OS. A virtual machine is required to execute this byte code. This is called a Java Virtual Machine (JVM). At execution time, the JVM converts the byte code into machine specific executable code which can then be executed by the OS and the CPU.

Java also provides an extensive set of pre-written utilities to help developers quickly implement commonly required solutions. For example, date arithmetic, such as the number of days or weeks between two dates, can be quite complex to implement from scratch. Java provides class libraries for date arithmetic so that the developers don't have to code that. This makes Java an extremely productive development environment.

Not all applications need all the utilities provided by Java. Making the full range of capabilities of Java to these applications will be an overkill and waste valuable resources. This is especially important for machines such as cell phones that do not have enough horse power and storage capacity. To address

this problem, Java is made available in three separate editions. Java ME is meant for mobile and PDA devices. Java SE is meant for desktop GUI applications and web browser applets. To develop client server enterprise applications, the characteristics of which we have already talked about, you will need Java EE.

## 2.3 What is Java EE?

- Java Enterprise Edition extends Java SE with additional support for client-server applications.
- Allows developers to focus on business logic rather than infrastructure
- Defines several major application components:
  - ◇ Servlet
  - ◇ JSP
  - ◇ EJB
  - ◇ JPA Entity
- Also defines JavaServer Faces (JSF) as a standard framework for Java web applications
- An “Application Server” is a class of software that provides the framework to execute Java applications written according to the Java EE standards.
  - ◇ Provides flexibility in choice of vendors as many offer Java EE compliant software

### What is Java EE?

Meeting the demands of an enterprise application, such as security, integration and scalability can be an arduous task for the programmers. These topics go into the very advanced areas of software development. Very few programmers are actually capable of implementing solutions for these problems from the ground up. Even if they could, it will take them years to get it right.

Java EE provides a monumental helping hand to the developers. It allows the developers to focus on the business logic and hand and take their minds off the infrastructural demands such as security, transaction management and scalability. Java EE provides easy to use APIs and means to implement these needs. An application server is an execution runtime that executes applications written according to the Java EE standards.

## 2.4 The Java EE Specifications

- Define the programming model for component-based Enterprise Applications
  - ◇ The Java EE API extends the standard Java API
- Define a standard Application Server environment to execute these components
- Define a standard way for packaging components of an Enterprise Application
  - ◇ This includes "Deployment Descriptors" with deployment information about Java EE applications
- Provide for a flexible architecture

### The Java EE Specifications

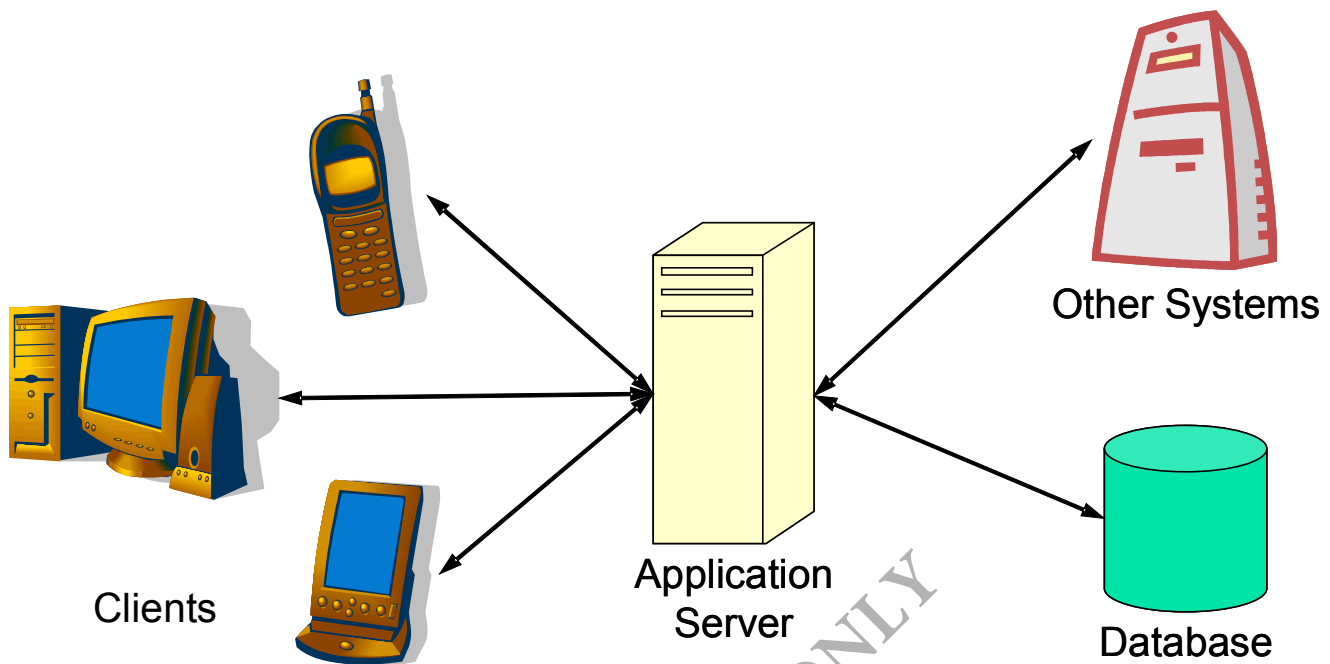
We have already discussed how application code is written according to the Java EE standards. In addition to that, Java EE also provides standards for packaging an application. This is somewhat similar to the way non-Java applications are packaged in .EXE files in Windows. A packaged Java EE application is deployed in an application server. The application server acts as a virtual operating system for these applications.

The current version of Java EE is 6. Prior to Java EE 5, the technology was named J2EE. The last version of J2EE was 1.4. WebSphere Application Server 8.0 supports Java EE 6, and is backwards compatible with J2EE 1.2 through Java EE 5.

Details about Java EE components, the structure of a Java EE application, and the architecture of an Application Server will be discussed later in this and other lectures.

## 2.5 Role of Application Server

- "Multi-Tiered" Application
  - ◇ Client, Application Server, Data



## Role of Application Server

The diagram above shows a standard n-tier architecture where the GUI front end of an application is separated from the business logic which is separated from the data storage.

The GUI front end, also called the client, can be a browser, cell phone, PDA (such as iPhone) or any other desktop application. A Java EE enterprise application, installed in the application server machine, uses the JSP technology to generate dynamic web pages which are then rendered by the browsers. The business logic of the application is written using EJB and JavaBean technologies. The application server may access back end systems like databases, PeopleSoft, Oracle Financial, SAP etc. Java EE provides mechanisms to access the back end systems in a secure and reliable manner.

When the enterprise application interacts with the client front end, the application server implements all the network communication features needed by a client/server style programming. Developers who develop enterprise applications are completely shielded from this task. Writing network communication code is complicated and error prone, so the help from the application server in this area is most welcome. The application server also manages the user authentication and access authorization tasks.

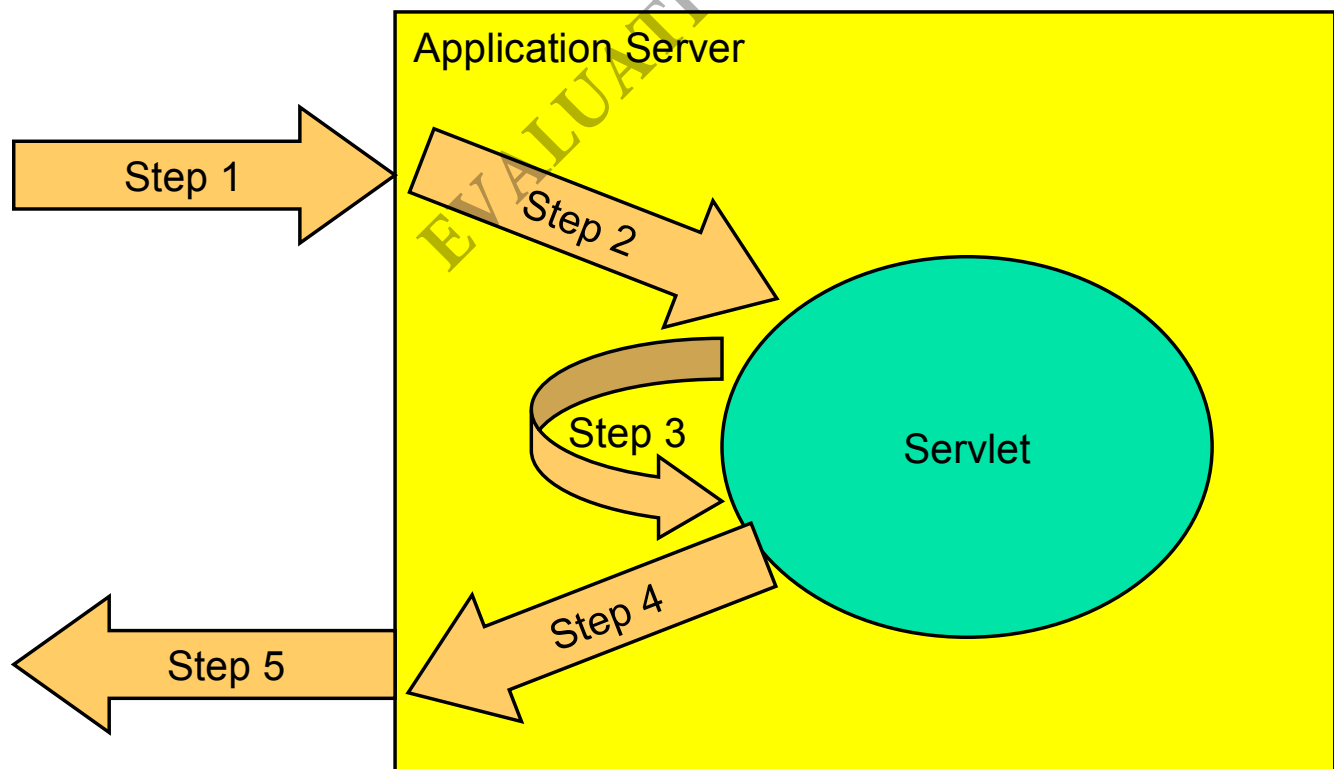
When the enterprise application interacts with a back end resource, such as a database, the application server manages the transaction. Transaction management is a hugely important issue. As shown in the diagram above, the application needs to use the services of multiple back end systems. Let us say that it saves data in a database as well as SAP. What happens if SAP rolls back the transaction. Unless proper precaution is taken the database will commit the data independent of SAP. This will cause the business to go into an invalid state. Java EE solves the problem through the use of distributed or global transaction. Once this is enabled, if any of the back end resources roll back the global transaction, all remaining resources also rollback in tandem. The application server acts as the transaction coordinator

in this case.

## 2.6 What is a Servlet?

- Java code designed to handle specific client requests
- Coded by developers with a few special operations which will be called by the server
  - ◇ These will be called when a client request for the Servlet is received
  - ◇ Define what code will execute when this happens
- Good for: making decisions, handling errors, delegating responsibility to other classes
- Not good for: displaying output to a client using HTML, directly manipulating data

## 2.7 Servlet Execution





## **Servlet Execution**

1. Client submits request to Application Server
2. Application Server uses URL to find Servlet designed to handle the request
3. Servlet executes code written by application developers to process request
4. Servlet delivers response to be forwarded back to client
5. Application Server sends the response back to the client using the protocol that was used to send the request

## **2.8 What is a JSP?**

- JavaServer Page
- An HTML file with special “JSP tags” to display dynamic content
- Dynamically processed by the server when a request is received
  - ◇ JSP tags only execute on the server
  - ◇ Client just sees standard HTML output
- Good for: displaying output to a client using HTML, using HTML editing tools for design
- Not good for: handling complex client requests, directly manipulating data

### **What is a JSP?**

JSP files can use other markup besides HTML. HTML for Web Browsers is most common.

## 2.9 JSP Code Sample

displayBalance.jsp executes on server

```
<HTML><BODY>
<jsp:useBean id="customer" type="com.myhost.Customer" scope="request" />
<jsp:useBean id="account" type="com.myhost.Account" scope="request" />
Hello, <%=customer.getFirstName()%>!  
You have <%=account.getBalance()%> dollars
</BODY></HTML>
```

dynamically  
generates

HTML displayed on client

```
<HTML><BODY>
Hello, Stuart!  
You have 1000 dollars
</BODY></HTML>
```

### JSP Code Sample

The code from the JSP file above is the contents of the file as it exists on the server. The special tags access information from other components on the server and display values in the output. Just like a Servlet, the JSP only executes when a request is received from a client. When the output is received by the client, they only see standard HTML which includes the values that were dynamically generated. They may not even know that a JSP was used to generate the output.

## 2.10 What is an EJB?

- Enterprise JavaBean
- Java components that provide reusable business logic
- Able to use additional services provided by the server, like distributed transactions
- Not directly accessible from a web browser
  - ◇ An EJB client will be a servlet or another Java class
  - ◇ Used as part of processing many different types of requests

- Good at: directly manipulating data, providing transactional qualities, modeling business logic

## 2.11 EJB Types

- Two major types of EJB
  - ◇ Session Bean – Encapsulates business logic
    - Stateful – Retains information about a particular client interaction
    - Stateless – No information retained about client interaction
  - ◇ Message Driven Bean – Responds to asynchronous JMS messages
    - Client interacts by sending a message
- Java Persistence
  - ◇ JPA Entity – Represents persistent data in relational databases
  - ◇ A JPA Entity is not an EJB

### EJB Types

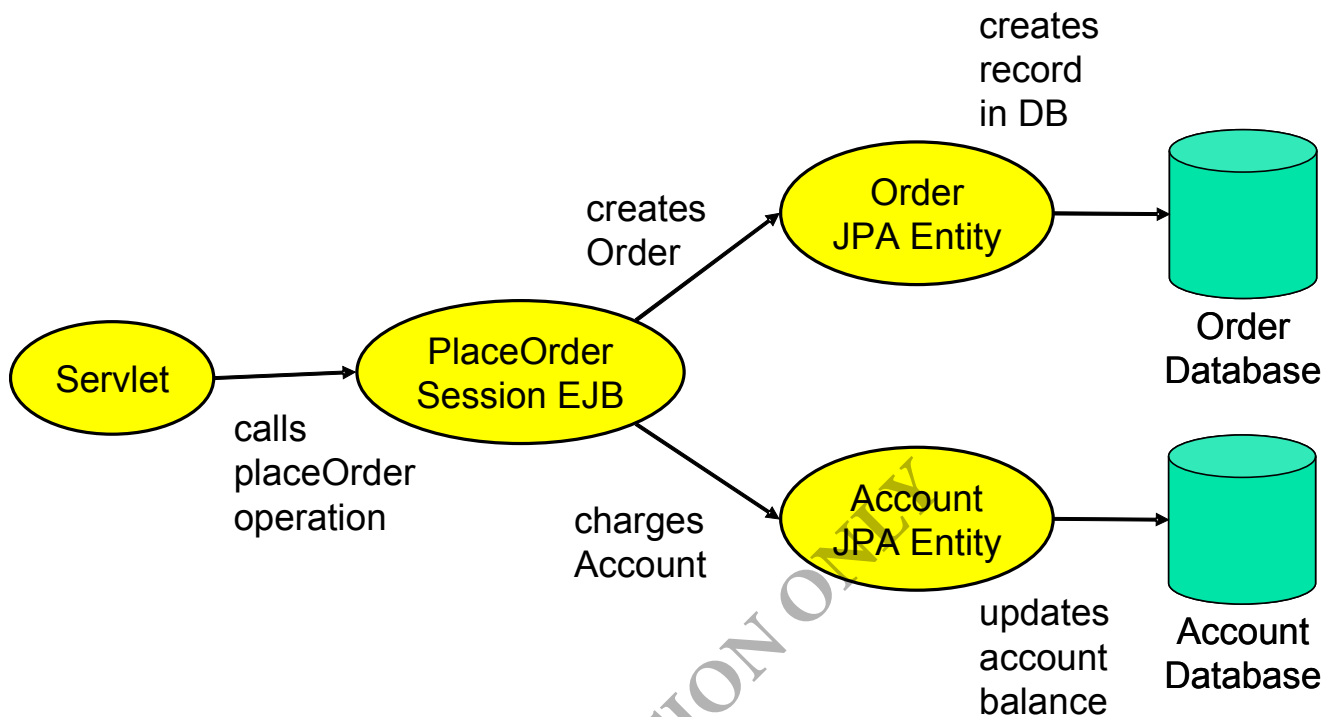
JPA stands for Java Persistence Architecture. It allows developers to load, save and delete data from a relational database without writing any SQL code. Writing database access code can be time consuming and mindlessly tedious. JPA definitely improves programmers' productivity in that regard. JPA entities represent the so called "nouns" in the system, such as, Customer, Order, Address etc. Nouns represent data that need to be saved in a database.

With previous versions of EJB technology the only standard “persistence” model were “Entity EJBs”. These were mapped to databases differently depending on the application server vendor and were not available outside of EJB technology. JPA makes persistence mapping to databases standard between vendors and makes it available even when not using EJBs. However, JPA is typically combined with EJBs to get the transactional semantics of the EJB environment.

A session EJB represents the verbs or actions in the system. Functionality such as placing of an order or canceling an order is implemented by the session EJBs.

A client sends a request to a session EJB by invoking one of its methods. The client waits for the method to complete its work. This is called synchronous style of programming. For various reasons, as we shall discover later, this style of client/server interaction is not always ideal. This is where message driven bean (MDB) comes in. A client can send it a message but does not wait for the message to be processed. The MDB processes the message "offline".

## 2.12 EJB Examples



### EJB Examples

The diagram above shows a classic case of how EJBs are used in a Java EE application. In this case, the application is web based. As a result, the request for action, to place an order, is made by a user from a browser and is received first by a servlet. The servlet then invokes the `placeOrder` method of the `PlaceOrder` session EJB. The data for the new order is then saved in two separate databases with the help of two JPA entities – `Order` and `Account`.

The example also illustrates distributed transaction. The transaction begins with the call to the ‘`placeOrder`’ operation and ends after all the steps have been executed. The JPA entities "inherit" this transaction that was started at the beginning of execution of the `placeOrder` method. This means that if there is a problem with creating the `Order` in the database the update to the `Account` database, which represents the charge, will not be saved. Instead the `Account` database will be restored to the state it was in before the operation began.

Since JPA entities are used to modify and query data they should not be used to encapsulate complex business logic. This business logic, which includes creating an `Order` and charging the appropriate `Account`, is instead provided in a Session EJB.

## 2.13 Model-View-Controller Architecture

- Splits up responsibilities in an application
- Allows different components of an application to do what they do best
- Well integrated with Java EE components
  - ◇ Model – EJB or JavaBean – data and business rules
  - ◇ View – JSP – displays content of model
  - ◇ Controller – Servlet – Translates user interactions with the view into actions in the model
- Web applications that use JSF instead of Servlet/JSP use JSF pages for the view and JSF “Managed Beans” for the controller logic

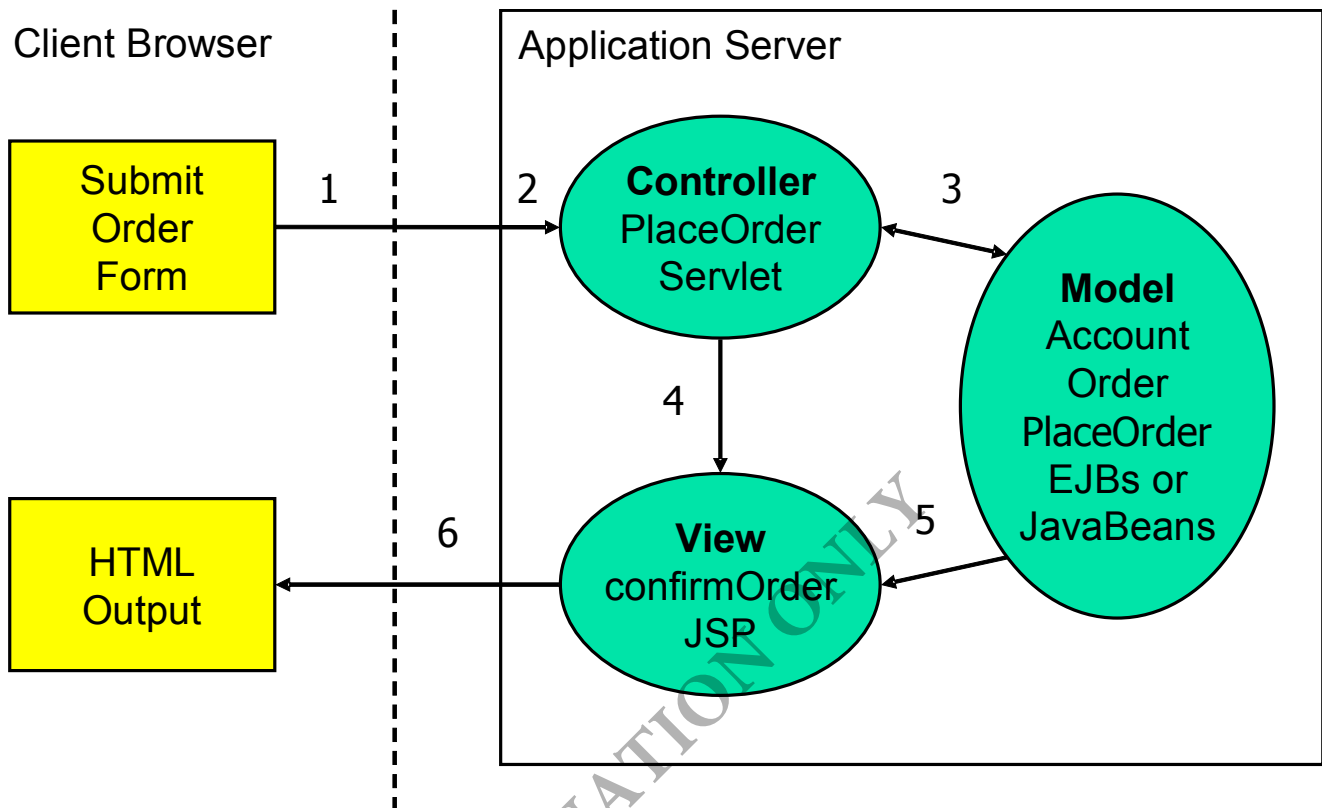
### Model-View-Controller Architecture

A JavaBean is a standard Java class which encapsulates properties and methods. Although not unique to the Java EE specification they are used heavily with Java EE applications.

A JSF Managed Bean is just a regular JavaBean which is managed by the JSF framework and whose properties or methods are linked to elements on a JSF page.

In this architecture, the model is the most flexible part. Lots of different things can be part of the model including other Java Classes which are not EJBs. Basically, anything that is not designed for client interaction can be considered to be part of the model.

## 2.14 MVC – An Example



### MVC – An Example

1. The user fills in the 'Submit Order' form and presses the 'submit' button.
2. The 'PlaceOrder' Servlet receives the request and checks that all the required information is present.
3. The Servlet asks other application components, like the PlaceOrder EJB in the previous example, to perform the work required to place the order.
4. If the order is created successfully, the Servlet will forward the request to the 'confirmOrder' JSP.
5. The 'confirmOrder' JSP will retrieve information from the model to display back to the user.
6. Once all the dynamic information is retrieved the JSP sends the response back to the user.

There can obviously be more complexity to this picture than is being shown. This is intended to demonstrate the major divisions of responsibilities. One thing to note is that all of the decisions and processing of the request has occurred before the JSP. Rather than the Servlet responding directly to the client, it delegates this responsibility to the JSP. The JSP is there just to display the results.

## 2.15 WebSphere Application Server Platform

- Software which provides the infrastructure for dynamic e-business
- A set of related products to build, deploy and integrate Java EE applications
- Comes in various editions
  - ◇ Community Edition
  - ◇ Express Edition
  - ◇ Base Edition
  - ◇ Network Deployment Edition
  - ◇ z/OS Edition

### WebSphere Platform

We will not discuss these editions going forward.

**WebSphere Application Server Community Edition 3.0.x:** Free Java EE server that can be used for both personal and commercial use. It provides support for Java EE 6 and Java SE 6 standards. It's code base is based on the Apache Geronimo open source application server. It is a light weight application server and does not provide the administration interface provided by other editions and lacks a number of production level features provided by other editions.

**WebSphere Application Server for z/OS:** This application server contains all the functionality of the Network Deployment edition and built to run on z/OS operating system.

Discussion of the above editions is outside the scope of this course. Additional information on these editions can be found at

<http://www.ibm.com/software/webservers/appserv/was/>

## 2.16 "Base" Edition

- A Java EE 6 compliant Application Server
- Official name of the product is "WebSphere Application Server"
- Supports Java EE 6 which includes:
  - ◇ EJB 3.1
  - ◇ JPA 2.0

- ◇ Servlet 3.0
- ◇ JSP 2.2
- ◇ JSF 2.0
- Also supports J2EE 1.2 through Java EE 5 for backwards compatibility
- Supports Java SE 6 (and optionally Java SE 7)
- Integrates with several different Web Servers

## 2.17 "Base" Edition

- Support for relational databases and Web Services
- Web-based and command line administration
- Additional tools for deployment, troubleshooting and performance included
- Single-server environment
- No clustering or failover support
- Comes with an assembly tool (Rational Application Developer Assembly and Deploy) to support application assembly and configuration

### "Base" Edition

Rational Application Developer provides a wizard based development for Java EE based applications consisting of EJBs, Servlets, JSPs, JSF and/or Web Services. It also provides support for debugging and testing applications.

## 2.18 Other WAS Editions

- Express Edition
  - ◇ Same "core" features as Base edition
  - ◇ Licensing limited to smaller server hardware
  - ◇ Full Java EE 6 compliance
- Network Deployment
  - ◇ Same "core" features as Base edition



- ◇ Multi-server environment
- ◇ Adds clustering and failover support
- ◇ New “Intelligent Management” features in WAS ND 8.5
- ◇ Web services gateway functions
- ◇ Ships with Edge Components for network load balancing

## **Other WAS Editions**

For full comparison of the licensing and features of the various editions refer to IBM WebSphere Application Server product information.

### **2.19 Other WAS Editions**

- WAS Community Edition (WAS CE)
  - ◇ Full support for Java EE 6
  - ◇ User-friendly web-based administration
  - ◇ Eclipse WTP plug-in available
  - ◇ Free to download and use, even in production
  - ◇ Fee-based support available at different levels
- WebSphere Hypervisor Edition
  - ◇ Optimized for server virtualization environments

## **Other WAS Editions**

WAS CE is generally used in smaller environments where cost is a large factor.

WebSphere Base and Network Deployment editions represent the bulk of the WebSphere environments for most WebSphere clients.

### **2.20 IBM HTTP Server**

- Packaged with WebSphere Application Server
  - ◇ Available if you do not have another web server to integrate with

### WebSphere Application Server

- Based on the Apache Web Server with additional features
  - ◇ IBM support for SSL secure connections
  - ◇ Integration with WebSphere Base or Network Deployment Administration Console
  - ◇ Security integration through LDAP

## 2.21 Other Products

- IBM Business Process Manager (BPM)
  - ◇ Provides a runtime environment for business process choreography
- WebSphere Portal Server
  - ◇ Delivers a single point of interaction with applications, content, processes and people
- WebSphere Commerce Suite
  - ◇ Provides complete solutions for e-commerce sites
- Rational Application Developer
  - ◇ Build and test Java EE Enterprise Applications
- WebSphere MQ
  - ◇ Exchanges information across different platforms allowing for integration of existing solutions

## 2.22 Java EE and WebSphere Overview Questions

1. What is the relationship between an Application Server and an Enterprise Application?
2. Which Java EE component closely resembles an HTML page?
3. What role do Servlets play in the MVC architecture?
4. Which Java EE component contains reusable business logic or data?
5. What are the 3 main editions of WebSphere Application Server (WAS)?

## 2.23 Java EE and WebSphere Overview Answers

1. An Application Server is used to run an Enterprise Application
2. JSP
3. Controller
4. EJB
5. Base, Express, and Network Deployment

## 2.24 Reference

- WebSphere Software Platform
  - ◇ [www.ibm.com/software/websphere/](http://www.ibm.com/software/websphere/)
- WebSphere DeveloperWorks Zone
  - ◇ [www.ibm.com/developerworks/websphere](http://www.ibm.com/developerworks/websphere)
- WebSphere Information Center
  - ◇ <http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp>
- WebSphere Redbooks
  - ◇ <http://www.redbooks.ibm.com/portals/websphere>
- Java EE Documentation
  - ◇ <http://www.oracle.com/technetwork/java/javaee/overview/>

EVALUATION ONLY

## Chapter 3 - WebSphere Installation with Installation Manager

---

### ***Objectives***

At the end of this chapter, participants will be able to understand:

- Pre-Installation Tasks
- WebSphere Profiles
- Installation Manager
- Silent Install
- Creating Profiles
- Directory Structure of WebSphere Application Server
- Configuration Repository

### **3.1 General System Requirements**

- WAS can be installed on Windows, Linux, AIX, HP-UX, Sun Solaris, IBM i, and z/OS
- There are 32-bit and 64-bit editions of the software available
- Although WAS does not require a database it can support many different databases
- Specific Web Browser versions are required for the web-based administration and InfoCenter access
- Check for the latest software and hardware prerequisites at:

<http://www.ibm.com/software/webservers/appserv/was/requirements/>

### **General system Requirements**

Before installing WebSphere Application Server, always read the system requirements document and make sure that your OS, database, web server and hardware meets the minimum requirements.

### **3.2 Pre-Install Tasks**

- Confirm hardware and software prerequisites
- Check WebSphere support site for notices

- Configure TCP/IP networking\*
- Check ports to prevent conflicts
- Read platform-specific installation information from InfoCenter
- Create/verify platform-specific user IDs and groups for installing and running WAS

## Pre-Install Tasks

When you install WebSphere it will be tied to a hostname or IP address during the installation. After supplying this information the value must remain fixed for the computer hosting the Application Server. The value specified during installation will be used by remote machines to connect to the application server. It is best to use a fixed hostname and use this during the installation. This will allow you to continue to use a DHCP server in your environment and reassign IP addresses regularly.

Some sample ports used by WebSphere. A complete list in the InfoCenter can be found by searching for "port number settings":

9080 - Web container's HTTP port.  
9443 - Web container's HTTPS port.  
9060 - Admin console's HTTP port.  
9043 - Admin console's HTTPS port.  
2809 - Bootstrap port.  
8880 - SOAP connector port.

If using IBM HTTP Server, the following ports are used:

IBM HTTP Server Port: 80  
IBM HTTPS Server Administration Port: 8008

## 3.3 Pre-Install Tasks

- WebSphere can be installed by a user having administrator (Windows) or root (UNIX) level access or a non-administrator or non-root userid.
- Non-root users can also perform silent installations.
- However, there are certain restrictions when installing as a non-root user, such as
  - ◇ Non-root install for the web server plug-in is supported only if the application server is installed as a non-root user.
  - ◇ Entries for the application server are not created in the Windows Services Panel.

## Pre-Install Tasks

Various notes about non-root installation and other topics can be found by searching the InfoCenter with the term “non-root”.

In addition to the application server, non-root users can install all WebSphere components such as the HTTP server, web server plug-in and Rational Application Developer (RAD). These components are discussed later in the course.

## 3.4 Installation Manager

- WebSphere 8.0 completely changed the method of installation
- The “Installation Manager” tool, or IM, is now used for all installation tasks
- This tool is common across multiple IBM products and has it's own InfoCenter

<http://pic.dhe.ibm.com/infocenter/install/v1r5/index.jsp>



- This tool is installed separately before installing WebSphere software
- The Installation Manager supports graphical (GUI) and command line (or “silent”) installation

## Installation Manager

The InfoCenter above is for the Installation Manager version used with WebSphere 8.5.

## 3.5 Product Repositories

- Installation Manager works with installable product software from “repositories”
- Product software like WebSphere 8.5 should be extracted to the same directory which will have a 'repository.config' file

 Copyright	5/5/2011 11:58 PM	Text Document	1
 repository	5/6/2011 6:25 AM	CONFIG File	1

- Product repositories will contain complete meta-data on product name, version, supported platforms, features, prerequisites, etc
  - ◇ Even the IM has it's own repository for installation

- Repositories can be configured within Installation Manager to define the list of software IM can install

#### **Add a repository**

Specify a repository and add to the repository preference list.



Repository:

/root/Software/WAS-ND-85/Main/repository.config

### **3.6 Silently Installing Installation Manager**

- “Silent” or command line installation of the Installation Manager is done with the 'imcl' command
  - ◇ You could also use the 'installc' or 'userinstc' commands but 'imcl' gives you more control of the options when installing silently
- Since Installation Manager is not yet installed you run this command from the 'tools' subdirectory of the Installation Manager product image
- Some of the options used are:
  - ◇ install com.ibm.cic.agent – This is the “feature id” of the IM itself
  - ◇ -acceptLicense – To indicate license acceptance
  - ◇ -accessRights <mode> - This can be 'admin', 'nonAdmin', or 'group'
  - ◇ -installationDirectory <directory>
  - ◇ -repositories <path to repository.config> - When installing IM this will be the repository of IM itself
- Example:

```
<IM_SOFTWARE>/tools/imcl install com.ibm.cic.agent  
-acceptLicense -repositories ../repository.config  
-accessRights nonAdmin -installationDirectory /opt/IBM/IM
```

#### **Silently Installing Installation Manager**

There are also 'installc' (which would install in administrator mode) and 'userinstc' (which installs in non-admin mode) which could be used to install IM silently.



The <mode> of the -accessRights option indicates the access mode Installation Manager is installed with. The allowed values imply the following:

- admin – This IM installation and managed packages can be access by administrators.
- nonadmin – This IM installation and managed packages will only be accessible to one user.
- group – All users in a specified group can share the Installation Manager instance and managed shared installed packages. Group mode is not available on Windows or IBM i.

### 3.7 Recording Response Files With IM

- Silently installing software with IM is still driven by response files
  - ◇ These are completely different than the response files used with previous WebSphere versions
- Although it would be possible to use a text editor to create a response file based on examples provided in WebSphere documentation this is error-prone
- A better option is to launch the graphical version of the Installation Manager in “record mode”
  - ◇ This will walk through the graphical wizard like you were installing products but instead of installing it will create a response file that could then be used with silent installation
- To do this you run the 'IBMIM' command from the 'eclipse' folder of the IM install with a few options
  - ◇ -record <response file to create>
  - ◇ -skipInstall <IM agent data location> - Should be a unique directory for recording
- Example

```
<IM_ROOT>/eclipse/IBMIM -skipInstall /opt/IBM/IM-Rec  
-record /root/install_response_file.xml
```

### 3.8 Installation Manager Console Mode

- When you can't use the GUI interface of Installation Manager you can run it in “console mode”

- ◊ To do this you run the 'imcl' command from the "tools" directory with the '-c' option

```
<IM_ROOT>/eclipse/tools/imcl -c
```

- Once started in console mode you can use various keys to indicate options for product repository locations and installing, updating, and uninstalling products
- Once you have indicated the options for a particular task, you can enter the 'G' key to generate a response file which can be run later or another key to perform the action right away

### 3.9 Installing Products Silently With Response Files

- Once you have a response file, you can use it to install a product silently according to the options in the response file
- You run the 'imcl' command line tool from the 'eclipse/tools' subdirectory of the IM install and point it to the response file along with a few options
  - ◊ install <path to response file>
  - ◊ -log <path to installation log>
  - ◊ -acceptLicense – To indicate license acceptance

- Example

```
<IM_ROOT>/eclipse/tools/imcl input  
/root/install_response_file.xml -log  
"WAS_silent_install.log" -acceptLicense
```

### 3.10 WebSphere Profiles

- In WebSphere 8.5, all run-time environment information is stored in a "profile"
- This information is separated from the actual product binary files that are shared by all WebSphere instances
- This "profile" includes:
  - ◊ Configuration files

- ◇ Log files
- ◇ Installed applications
- ◇ Property files

### **3.11 Benefits of Profiles**

- WebSphere product files only need to be installed once
  - ◇ Multiple profiles can be created from the same installation of product files
- Easy to configure multiple configurations on the same machine
  - ◇ This is common to test different versions of an application
- Better Usage of disk space
  - ◇ Less space required overall
  - ◇ Location of profile can be separate from product files
- Easier to apply fix packs and patches
  - ◇ These can update the product files and be applied to all profiles

#### **Benefits of Profiles**

The product files take a specific amount of space and do not grow because of activities like logging.

Separation of profiles can also improve backup efficiency. The product files do not change very often and do not need to be backed up as frequently. The profile directories would be backed up often.

If you want to have different service levels of product files on the same machine you must install the WebSphere product files more than once.

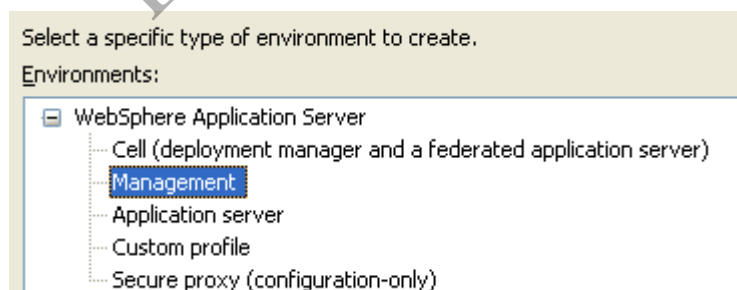
### **3.12 Creating Profiles**

- After installing the product files with Installation Manager you have to create a WebSphere profile to have a working configuration
  - ◇ This is now always done separately from the product file installation
- Profiles can be created in two ways

- ◇ Profile Management Tool
- ◇ manageprofiles command
- When created, profiles are given a name
- All files related to a profile are stored in a directory
  - ◇ Default is <WAS\_ROOT>/profiles/<profile-name> but can be changed

### 3.13 Profile Management Tool

- The Profile Management Tool (PMT) is a GUI based tool. It can be started in either of the following ways:
  - ◇ Execute pmt.bat/pmt.sh from the folder <WAS\_ROOT>/bin/ProfileManagement
  - ◇ Start PMT from First Steps console
  - ◇ Windows Start Menu
- The Profile Management Tool has different profile types available depending on the installed WebSphere edition
- All editions can create an "Application Server" profile
  - ◇ This is the "standalone" environment of a single server
- Network Deployment adds additional profile environments



### Profile Management Tool

In earlier versions Profile Creation Wizard was used. This was replaced by the Profile Management Tool in v6.1.

On executing the command the tool is started in a new window.

## 3.14 Profile Management Tool

- There are two ways to create a profile
  - ◇ "Typical" creation - Automatically given a profile name, location, unique ports, and run as Windows/Linux service
  - ◇ "Advanced" creation - You can supply profile name, node name, location, ports, whether to deploy the Admin Console and sample applications, and create a web server definition
  - ◇ Both options provide a choice to enable administrative security and provide the primary administrative identity

### Profile Management Tool

When using the "advanced" profile creation option there are two ways to specify the ports. You can use the "recommended" ports which are unique from other profiles or you can choose to override this and use "default" ports. You should only use default ports for multiple profiles if you know they will never need to run simultaneously.

## 3.15 manageprofiles command

- Profiles can also be created using the manageprofiles command
  - ◇ The Profile Management Tool runs this command internally
- This is a command line option that performs the same actions as discussed in the Profile Management Tool
  - ◇ You have additional commands available that are not provided by the graphical Profile Management Tool like deleting and backing up a profile
- It can be executed by running the manageprofiles command located in <WAS\_ROOT>/bin folder
- A log file is also created in <WAS\_ROOT>/logs/manageprofiles directory

### manageprofiles command

Some of the commands available for the manageprofiles tool:

- Creating a profile (manageprofiles -create)

- Deleting a profile (manageprofiles -delete)
- Listing profiles (manageprofiles -listProfiles)
- Get a profile name (manageprofiles -getName)
- Get a profile path (manageprofiles -getPath)
- Backup a profile (manageprofiles -backupProfile)
- Restore a profile (manageprofiles -restoreProfile)
- Get the default profile name (manageprofiles -getDefaultName)
- Set the default profile name (manageprofiles -setDefaultName)

### 3.16 manageprofiles command

- Some commands take parameters that represent arguments to complete the task
- For example, the create profile command can take the following arguments

```
app_server_root/bin/manageprofiles.sh -create  
-profileName profile_name  
-profilePath profile_root  
-templatePath template_path
```

- You can get general help for using the command

```
manageprofiles -help
```

- or help with a specific mode for the tool

```
manageprofiles -create -help
```

```
[root@linuxvm bin]# ./manageprofiles.sh -create -help
```

**Function:**

Creates a new profile

**Syntax:**

manageprofiles -create -<argument> <argument parameter> ...

**Arguments:**

The following command line arguments are required for this mode:

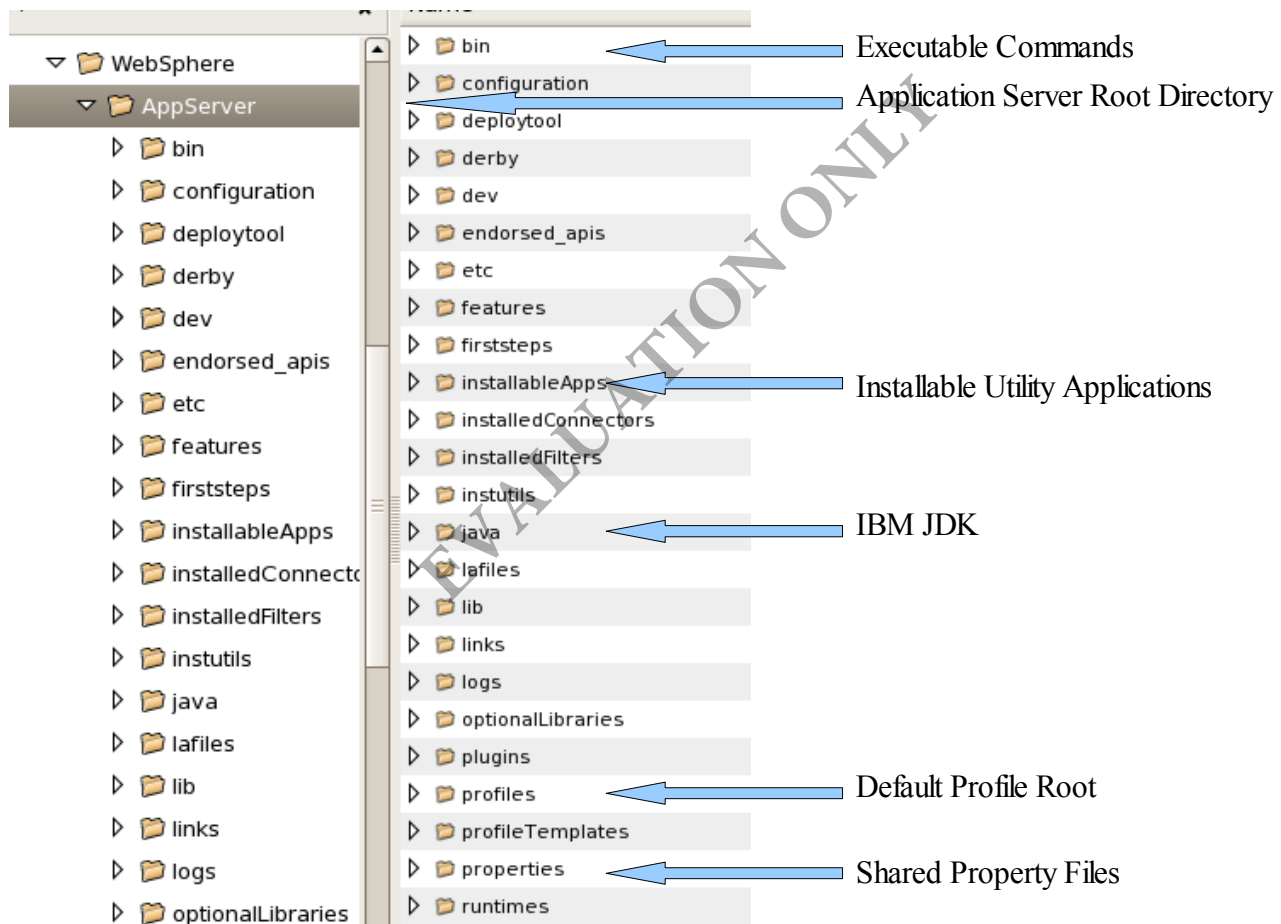
## manageprofiles command

Some other commands with parameters are shown below. For a list of all commands available refer the WebSphere Application Server 8.5 infocenter (<http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp>).

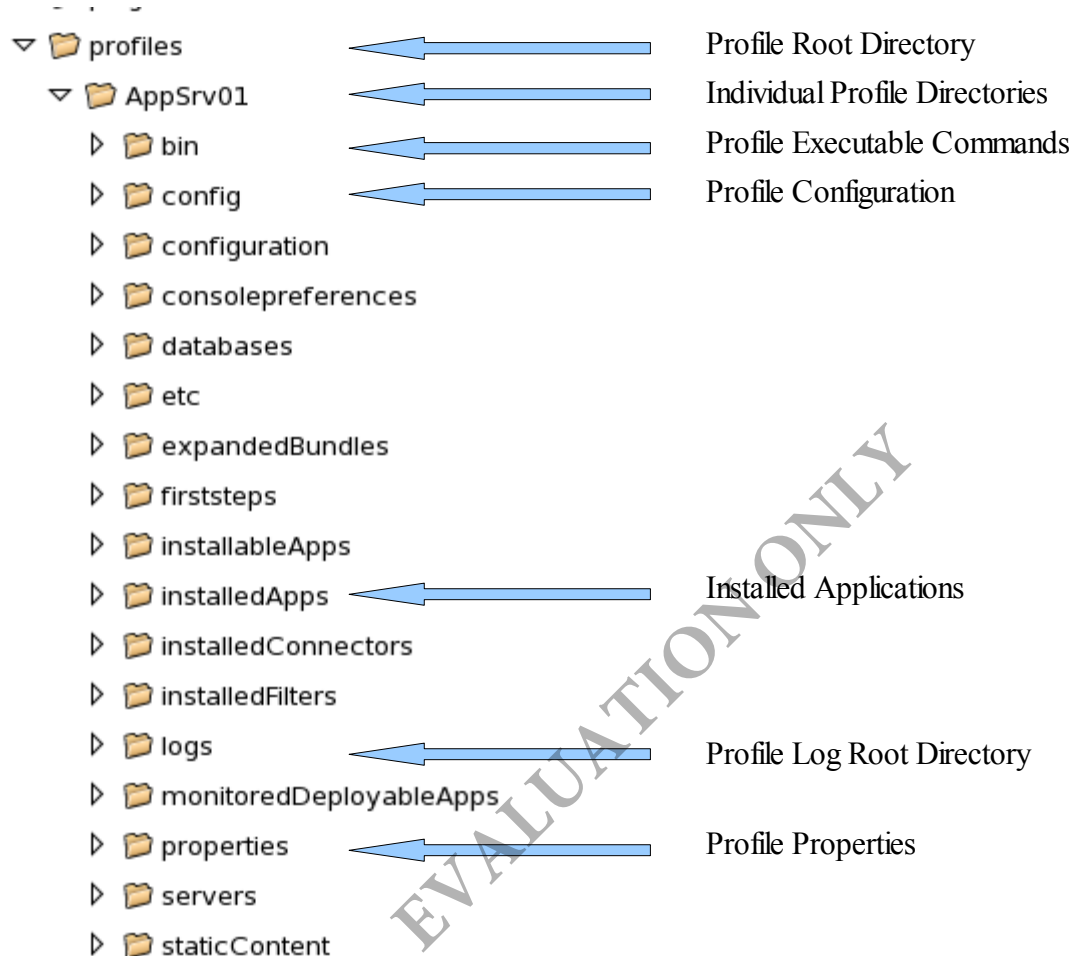
`manageprofiles.bat(sh) -backupProfile backupFile_name`: Creates a backup of the profile to the specified file. This file is stored as a zip file.

`manageprofiles.bat(sh) -deleteAll`: Used to delete all profiles.

## 3.17 Product Files Directories



### 3.18 Profile Directories



### 3.19 Cell/Node/Server

- In WebSphere, the terms Cell, Node, and Server have specific meanings
  - ◇ Cell – Everything inside of a configuration containing one or more Nodes
  - ◇ Node – A group of one or more Application Servers running on the same machine
  - ◇ Server – An Application Server process which handles application requests
- In a "standalone" server environment the Server definition gets the most



attention

- ◇ Cell and Node aren't as important because the cell configuration consists of a single Node with a single Application Server process

## **Cell/Node/Server**

The following profile types correspond to a "standalone" server environment:

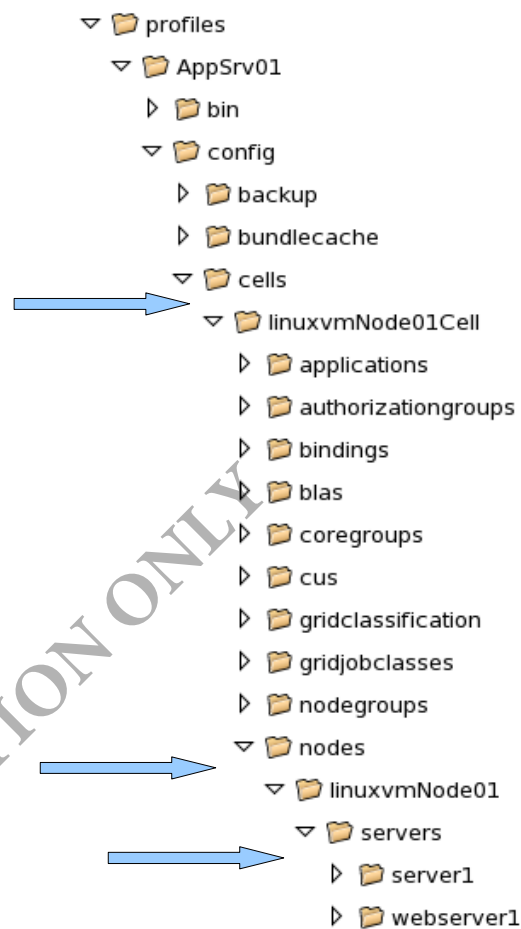
1. All Base edition profiles
2. Network Deployment edition "Application Server" profiles

A machine can have more than one "standalone" server configuration but these are managed separately. Configurations which have more than one Node and are part of the same configuration are only possible with Network Deployment which will be discussed later.

EVALUATION ONLY

## 3.20 Config Directory

- Contains configuration information in XML files
- Contains directories for the major definitions in WAS
  - ◇ Cell
  - ◇ Node
  - ◇ Server
- Modified using administrative tools, not by hand



### Config Directory

Even a "standalone" server has the configuration structure shown above. This is so that multi-server environments managed by Network Deployment can be formed simply by merging together the configuration of several Nodes. This process will be discussed in more detail later.

## 3.21 Important Configuration Files

- Cell level
  - ◇ security.xml - security settings
  - ◇ virtualhosts.xml - host name and port aliases
- Node level

- ◇ serverindex.xml - ports used by all servers on the node
- Server level
  - ◇ server.xml - all properties of a server
  - ◇ ws-security.xml - web services security settings
- All levels
  - ◇ resources.xml - resource configuration
  - ◇ variables.xml - WebSphere substitution variables

## 3.22 Starting and Stopping the Server

- Command Line batch file
  - ◇ In <PROFILE\_ROOT>/bin (the profile's executable directory)
  - ◇ 'startServer <server-name>' & 'stopServer <server-name>'
    - <server-name> is 'server1' in an Application Server profile
  - ◇ Better display of startup or shutdown errors
  - ◇ This is the only option on UNIX/Linux where the commands are 'startServer.sh' and 'stopServer.sh'
- Start from the Windows 'Start' Menu
  - ◇ Start => Programs => IBM WebSphere => IBM WebSphere Application Server V8.5 => Profiles => <profile-name> => Start the Server / Stop the Server
- Windows Services
  - ◇ Control Panel, then Administrative Tools => Services
  - ◇ Look for 'IBM WebSphere Application Server V8.5'

## Starting and Stopping the Server

If you are running WebSphere on a non-Windows environment the only option is to start and stop from the command line. The commands are also a little different as you have to include the extension for the script. For example, startServer.sh <server-name> and stopServer.sh <server-name>.

Running the startServer and stopServer commands from the profile's executable directory is the easiest

as it will always control the servers of that profile. The same commands are also available in the <WAS\_ROOT>/bin directory but may require the '-profileName' option to indicate the profile the command should be executed for. One profile will always be the "default" profile which can run commands from the <WAS\_ROOT>/bin directory without the '-profileName' option.

The Start menu icons and the command line batch files are interchangeable. This is because the Start menu icons actually call the command line batch files. The problem with the Start menu icons is that the command window disappears immediately and does not give you a chance to see if there were errors starting or stopping the server.

If the server is registered as a Windows service the command-line and Start menu options start the corresponding Windows service. These options are largely interchangeable but it is good to use the same method to stop the server as the method used to start it.

### **3.23 Application Server Installation Questions**

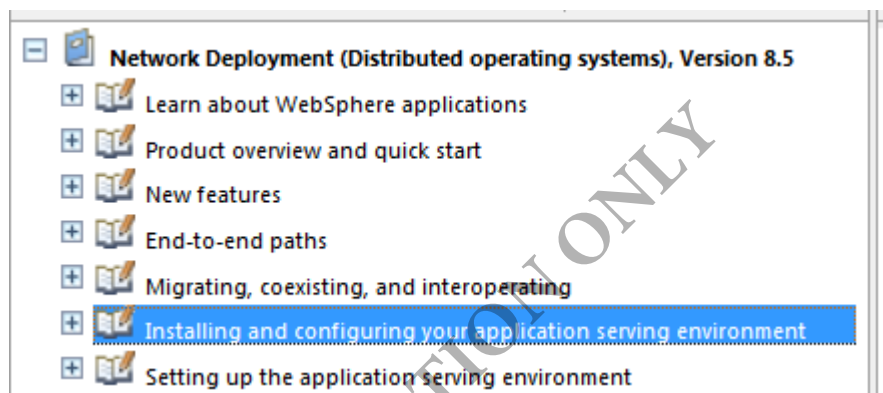
1. What IBM reference provides a good source of information on installing WAS?
2. What does WAS v8 create to store run-time environment information, including configuration and log files?
3. What directory contains all the executable commands in WAS?
4. What is the difference between a cell, a node, and a server?
5. T/F: WAS stores configuration information in a database.

### **3.24 Application Server Installation Answers**

1. InfoCenter
2. Profiles
3. bin directory
4. A cell contains all the configuration information and is a collection of nodes. A node is a collection of servers running on one machine. A server is an application server process which handles application requests.
5. False. It stores configuration in XML files in a profile's config directory.

### 3.25 Reference

- WebSphere System Requirements
  - ◇ <http://www.ibm.com/software/webservers/appserv/was/requirements/>
- WebSphere Support
  - ◇ <http://www.ibm.com/software/websphere/support/>
- Installation guide on InfoCenter:
  - ◇ <http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp>



### 3.26 Lab Summary

- WebSphere 8.5 Installation
  - ◇ Install Installation Manager
  - ◇ Record response file
  - ◇ Install WebSphere software
  - ◇ Create a WebSphere profile
  - ◇ Explore the major WebSphere directories
  - ◇ Verify installation
  - ◇ Start the Application Server
  - ◇ Test the Server
  - ◇ Stop the Server

### **3.27 Common Lab Errors**

- Be sure you are using the root user to login
- When installing, be sure to leave the Application Server samples deselected. This is done to improve server performance

EVALUATION ONLY