

Overview:

IBM Websphere is a now a well-established platform of choice for developing distributed, scalable J2EE enterprise applications. This course demonstrates how to build applications using the IBM Websphere Studio Application Developer, and to then test and deploy those applications on the IBM Websphere application server. This course will introduce you to J2EE technology, and how it can be applied when using the IBM Websphere platform

This course combines lectures with practical exercises that help the delegates to put what they have learned on the course into practice. The exercises specifically build on what has been recently taught and are built up as the course progresses.

This course is aimed at experienced Java developers, who need to learn the skills required to build J2EE applications and deploy them on the Websphere Application Server platform.

Prerequisites:

Delegates should be familiar with the Java Programming language as far as at least being comfortable with its basic use and with OO concepts.

Objectives

At the end of this course, students should be able to:

- Understand the J2EE architectural paradigm, and appreciate how it can be used
- Develop and deploy Java Server Pages, Servlets and Java Beans within a web container
- Be able to implement the Jakarta Struts framework
- Use JNDI to locate distributed objects, including enterprise Java beans, and application server specific components
- Develop and deploy Entity Enterprise Java Beans
- Develop and deploy Session Enterprise Java Beans
- Appreciate the alternative persistence strategies, identifying the pros and cons of each
- Deploy a complete J2EE application within a J2EE compliant application server
- Understand Java Messaging Services, and be able to build and deploy messaging applications
- Develop and deploy Message-Driven Enterprise Java Beans

1. Introduction to J2EE

- What is J2EE?
- What technologies does J2EE encapsulate?
- What is the architecture of J2EE applications?
- When should I be considering J2EE in my applications?

2. Introduction to Websphere

- Introduce the Websphere product stack
- Identify the core products
- Developer Tools
- Server Tools

3. Working with Websphere Studio Application Developer

- Workspaces
- Perspectives
- Views
- Editors

4. Introduction to Java Servlets

- What is a Java Servlet
- Java Servlet lifecycle
- Introduction to the Servlet API
- How to develop and deploy Java Servlets

5. More on Java Servlets

- Processing form data from Java Servlets
- Session tracking from Java Servlets

6. JavaServer Pages

- What is a JavaServer Page?
- Directives in JSP
- Scriptlets in JSP
- Scripting Elements in JSP
- Standard Actions in JSP

7. JavaBeans and JavaServer Pages

- Using JavaBeans with JSP
- JSP and Servlet interaction
- Web application architecture

8. Introducing the JSP Standard Tag Library

- The Core Actions
- Using the EL in Tags

9. Introducing Jakarta Struts

- The need for frameworks
- Struts overview
- Components of a Struts application

10. Implementing Jakarta Struts

- Building a Struts Application
- Working with Forms
- Validation with ActionForms
- Processing with Action Objects

11. JNDI

- What is JNDI?
- How to use JNDI Locating remote objects
- Registering remote objects

12. Enterprise Java Bean Overview

- The four types of EJB
- Issues associated with EJB
- Packaging EJBs for deployment

13. Entity Beans

- Entity bean architecture
- Home and Remote interfaces
- Finder Methods
- Primary Keys
- Entity Context
- Deploying entity beans

14. Session Beans

- Session bean home and remote interfaces
- Using JNDI to locate entity beans
- Using entity beans from session beans
- Session Context
- Deploying session beans
- Accessing session beans from within JSP
- Design issues surrounding EJB development

15. Web Applications and EJB

- Web applications and EJB interaction
- Design considerations
- Architectural example

16. Java Messaging Service

- Introduction to JMS
- Topics and Queues
- Implementing messaging using Java

17. Message Driven Beans

- The role of the MDB
- Building an MDB
- Deploying an MDB