

This course teaches Java programmers how to use the JMS (Java Messaging Service) classes for developing applications in the IBM WebSphere MQ environment. Through lectures and extensive hands-on exercises, the students learn to design, develop and deploy industrial strength messaging applications. The course topics include JMS architecture, point-to-point messaging model, publish/subscribe model, working with queue and message objects, JMS administered objects and many other topics.

Audience

- This course is a must for Java developers who want to learn how to design, implement and deploy JMS application using the IBM WebSphere MQ transport

Prerequisites

- WT1043 Technical Introduction to WebSphere MQ, plus Java programming experience

Course Length

- 2 days

Learning Objectives

- Understand JMS architecture as implemented in WebSphere MQ
- Work with JMS administered objects
- Develop JMS based messaging applications
- Understand JMS security implementation
- Implement WebSphere MQ specific functions in JMS applications

Teaching Methods

- Lectures
- Hands-on workshops

Course Outline

WTF10

What is WMQ

- Integration = Reuse
- Integration Techniques: Procedure Call
- Integration Techniques: RPC
- Integration Techniques: File Sharing
- Integration Techniques: Database
- Integration Techniques: Message Queuing
- WMQ Concepts
- Message Descriptor (MQMD)
- Typical WebSphere MQ Environment
- Emerging WebSphere MQ Environment
- WebSphere MQ APIs

What is JMS

- JMS Concepts
- Java APIs Related to JMS
- More JMS Concepts
- JMS Administration
- JMS Object Model
- JMS Client Prototype
- JMS Reference

What is JNDI

- Interfacing naming and directory services
- JNDI is a Wrapper
- Directory Technologies
- What JNDI is not
- How are JMS and JNDI Related?
- JNDI Administration

JMS Administration

- JMSAdmin.config
- JMSAdmin utility

Examining a JMS Program

- JMS Client Prototype
- JMS Object Model
- JMS Reference
- Create JNDI Context Object
- Retrieve Connection Factory Object
- Retrieve Destination Object(s)

- Create Connection Object
- Create Session Object(s)
- Create Producer(s) / Consumer(s)
- Create Message Object(s)
- Start Message Delivery
- Cleanup
- Exceptions
- JMS Reference
- MQ Completion Codes
- MQ Reason Codes

JMS and Websphere MQ Resources

- Online Manuals
- Support Paks
- rfutil (utility to create, send, receive and examine messages)
- JMS Specification

Messages

- Message Descriptor (MQMD)
- MQMD Format
- Message Data Headers
- WMQ View
- MQHRF2 - Fixed
- JMS Logical View
- WMQ to JMS Mapping
- JMS Header Fields
- MQHRF2 - Folders
- JMS Header Fields
- Provider Specific Fields

JMS Message Selection

- Message Selector
- Specifying Property Values
- Message Selector Examples
- Message Selector Elements
- Message Selector Use

JMS Message Types

- JMS Message Type
- BytesMessage
- Create BytesMessage
- Unpack BytesMessage

- StreamMessage
- Create StreamMessage
- Unpack StreamMessage
- TextMessage
- Create TextMessage
- Unpack TextMessage
- MapMessage
- Create MapMessage
- Unpack MapMessage
- ObjectMessage
- Create ObjectMessage
- Unpack ObjectMessage

JMS Program Initiation

- JMS Daemon or Service
- Asynchronous Receive
- Listener Registration
- WMQ Trigger Mechanism
- Message Driven Bean (MDB)

Request-Reply Pattern

- Overview
- Request-Reply Properties
- Simple Request-Reply
- Typical Request-Reply
- Request-Reply Variation

Publish Subscribe Pattern

- Overview
- Publishers
- Subscribers
- Brokers
- Pub Sub Commands
- Parallels

Transaction Processing with JMS

- What is a Transaction?
- Transaction Types
- Local Transaction Scenario
- Local Transaction Coding
- Poison Messages
- Global Transaction Scenario