

# 102/201 WebSphere MQ Introduction and Administration + Metastorm Integration Manager 8.5 Developer Workshop



## Course Syllabus

<b>Important Facts:</b>	5 days  60% Lecture  40% Hands-on Labs
<b>Target Audience:</b>	Future Developers and Architects using Metastorm Integration Manager for custom integration and development
<b>Class Objectives:</b>	<p>After completion of this five-day course, the student will be able to apply a basic working knowledge of WebSphere MQ capabilities including:</p> <ul style="list-style-type: none"><li>• Concepts</li><li>• Architecture</li><li>• Creating queue managers</li><li>• Administration of queues and channels</li><li>• Diagnosing and correcting problems</li><li>• Triggering</li><li>• Remote Administration</li><li>• Multi-Hopping</li><li>• Dead Letter Queue Handler</li><li>• WebSphere MQ Clients</li></ul> <p>In addition, the student will be able to apply an advanced working knowledge of Metastorm Integration Manager (MIM) tools and technologies for integrating applications and data including:</p> <ul style="list-style-type: none"><li>• Creating custom components</li><li>• Creating Process Flows</li><li>• Integration with Databases</li><li>• Integration with WebSphere MQ</li><li>• Adding exits to Transfer Requests</li><li>• Available sample exits</li><li>• Working with the XMScript programming language</li><li>• Working with XML data in XMScript Programs</li><li>• Accessing the Registry from programs</li><li>• Writing custom exits with the XMScript programming language</li><li>• Adding custom status to exits and components</li></ul>

**Prerequisites:**

This class is intended for students who will be responsible for working in a Metastorm Integration Manager environment to support integration with or without Managed File Transfer capabilities. This is an advanced level course and students in this class should possess some familiarity with networking, programming and computer fundamentals. This class assumes the student has no prior experience with WebSphere MQ. Students should be reasonably familiar with:

- Programming with a language such as C, C++, C#, Java, Perl, Ruby or JavaScript

**Topic Outline:****Day 1**

- Messaging and Queuing Concepts
- WebSphere MQ Introduction
- Basic Queue Manager Administration
  - Lab: Create a Queue Manager
- WebSphere MQ Objects
  - Lab: Create and Verify a Local Queue
- WebSphere MQ Sample Programs
  - Lab: Testing Local Queue
- Message Queue Interface (MQI)
- Application Design
- Triggering
  - Lab: Trigger Enablement
- Basic Problem Determination
  - Lab: Local Problem Determination

**Day 2**

- Distributed Queue Management
  - Lab: Setup and Test Channels
- Distributed Queuing Problem Determination
  - Lab: Distributed Problem Determination
- Remote Administration
  - Lab: Enabling Remote Administration
- Multi-Hopping
  - Lab: Multi-Hopping
- Queue Manager Clusters Overview
  - Lab: Clustering Demo
- Dead Letter Queue Handler
  - Lab: Setting Up a Dead Letter Queue Handler
- WebSphere MQ Clients
  - Lab : Configuring Clients

**Day 3**

- Introduction
- Components and Architecture
  - Lab: Component Management using the Process Monitor
- Managed File Transfer Overview

- o Lab: Basic Transfer with the Process Monitor
- Process Flow Overview
  - o Lab: The Transfer\_File Process Flow
- Workbench and Registry
  - o Lab: Creating a Registry Project with the Workbench
- Modeling Process Flows with ProVision
  - o Lab: Working with simple workflow models
- A Quick Tour of XMScript
  - o Lab: Working with Simple XMScript Programs
  - o Lab: Developing XMScript Programs with the Workbench

#### **Day 4**

- XML Support in XMScript
  - o Lab: Programming with XML in XMScript
- XMScript Workflow Servlet Engine (WSE)
  - o Lab: Working with the XMScript WSE
- Database Object Services (DBOS)
  - o Lab: Configuring and using DBOS
- Creating Activities in Workflow Models
  - o Lab: Working with Activities in workflow models
- Policies and XMDirectory()
  - o Lab: Working with the XMDirectory() Class in XMScripts

#### **Day 5**

- XMQ Wrapper
  - o Lab: Working with the XMQ Wrapper Class in XMScripts
- Status Processing
  - o Lab: Adding Status to XMScript-based Activities
- Control Flow in Workflow Models
  - o Lab: Adding Control Logic to Process Flows
- Transfer Request Exits
  - o Lab: Invoking Exits
- Creating Custom Exits
  - o Lab: Developing Custom Exits
- Creating Custom Status in Exits
  - o Lab: Adding Status to the Exit