

**COURSE ABSTRACT**

# Merative SPM for Developers (ADE)

## 7.X

9D73G

**Course Description**

This training provides learners with a technical understanding of the Merative Social Program Management (SPM) Application Development Environment (ADE).

A practical grasp of the SPM ADE is essential for teams wishing to build SPM-compliant solutions that provide high-quality citizen services. This course provides a solid grounding in the SPM model-driven development approach and ADE. It presents an architectural overview of the SPM application and introduces ADE features and tools for modeling, coding, building, and troubleshooting applications. During the course, learners will design and implement a simple end-to-end application using many ADE client and server features.

This training provides learners with the essential knowledge and hands-on skills required to use the SPM ADE to implement SPM solutions.

**General Information**

**Delivery Method:** Classroom (ILT) and self-paced with lab.

**Audience:** This course is intended primarily for developers and technical architects who will work on SPM implementation projects. The course is also useful for anyone who needs a technical understanding of the Merative SPM Platform, including testers and support engineers.

**Topics:** The course covers the following topics:

- Runtime and architectural review of Merative SPM
- Modeling SPM class types and relationships using Rational Software Architect Designer<sup>®</sup> (RSAD)

- Developing web client applications using User Interface Metadata (UIM) and client navigation features
- Implementing server features, including exception handling and validation
- Basic troubleshooting using trace and debugging tools

**Learning Objectives:** After completing the course, learners should be able to:

- Use features and tools in the SPM ADE for the following tasks:
  - Modeling classes and relationships
  - Implementing server-side features
  - Creating client pages and navigation
- Implement simple, end-to-end solutions using the SPM model-driven development approach and development tools
- Access and interpret developer guidelines contained in the SPM product documentation

**Prerequisites:** Learners should have a working knowledge of object-oriented concepts, Java, XML, SQL, and n-tier enterprise applications. The following course is recommended to gain a broad technical view of SPM:

- *Merative Social Program Management (SPM) 7.X Technical Overview (9D76G)*

**Duration:** 40 Hours (5 days ILT)

**Skill Level:** Intermediate

**Version:** This course was last updated for SPM V7. The course covers core topics that do not change much from one version of SPM to the next.

## Notes

The course duration gives learners an estimate of how much time they need to allocate to the course. The course duration does not specify the actual time required to complete the course, which varies by learner. The course agenda shows the schedule for a classroom (ILT) delivery. Learners taking this course in a self-paced environment should allow more time for exercises.

## Course Agenda

### Unit 1 - SPM Platform and ADE Overview

#### Lesson 1 – Course Introduction

Duration: 10 minutes

Learning objectives: After completing this lesson, learners should be able to:

- List the course objectives
- Outline the course format and resources
- List sources of help

## Lesson 2 – SPM Technical Review

Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Provide an overview of the components of SPM:
  - SPM Platform
  - SPM Application Modules
- Outline the runtime architecture used in SPM for the client and server applications
- Describe the product documentation that is provided with the SPM application

### Exercise 1 – Access Product Documentation

Duration: 30 minutes

## Lesson 3 – Introduction to the Application Development Environment (ADE)

Duration: 75 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Outline the model-driven development approach
- Describe the elements and supporting tools in the development environment
- List the client and server features
- List the outputs from code generators
- Describe the project directory structure
- List common build commands

### Exercise 2 – Access the ADE

Duration: 30 minutes

## Unit 2 – Implementing CRUDL Features

### Lesson 1 – Server Fundamentals Part One

Duration: 40 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Describe SPM class types
- Describe how UML stereotypes are used in SPM development
- Outline the SPM-generated class hierarchy
- Outline how to model and write server-side code

### Exercise 3 – Read a Person (Server Side)

**Duration: 60 minutes**

## Lesson 2 – Client Fundamentals Part One

**Duration: 50 minutes**

Learning objectives: After completing this lesson, learners should be able to:

- Describe the benefits of using User Interface Metadata (UIM)
- Describe the UIM elements that control the look of a page
- Define and use the following components:
  - Localization and externalizing strings
  - Common UIM elements
- Describe the client generation process
- Describe how to use the generated bean documentation

## Exercise 4 – Read a Person (Client Side)

**Duration: 60 minutes**

## Lesson 3 – Adding Create Pages

**Duration: 25 minutes**

Learning objectives: After completing this lesson, learners should be able to:

- Define a create page to insert data to the database
- Describe the purpose of the ACTION\_SET and ACTION\_CONTROL elements
- Link between pages
- List the typical steps for the SPM development process

## Exercise 5 – Add a Create Page

**Duration: 70 minutes**

## Lesson 4 – Adding Search Pages

**Duration: 35 minutes**

Learning objectives: After completing this lesson, learners should be able to:

- Describe the server-side tasks for returning multiple records
- Describe the client-side tasks for viewing multiple entries in a table
- Outline how to handle large return sets

## Exercise 6 – Add a Search Page

**Duration: 40 minutes**

## Lesson 5 – Adding Modify Pages

**Duration: 20 minutes**

Learning objectives: After completing this lesson, learners should be able to:

- Describe the modify stereotype operations
- Outline the support for record locking and optimistic locking
- Describe the 'pattern' for UIM modify pages:

- Combining SERVER\_INTERFACES
- Multiple connections in a FIELD

### Exercise 7 – Add a Modify Page

Duration: 65 minutes

### Exercise 8 – Case Study: Design

Duration: 35 minutes

## Unit 3 - Enhancing the Application

### Lesson 1 – Code Tables

Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:

- State the purpose of code tables
- Explain how to define code tables
- Explain how to associate code tables with your model
- Describe the outputs from the ctgen target
- Describe how to access code tables from your code

### Exercise 9 – Implement Code Tables

Duration: 40 minutes

### Lesson 2 – Data Manager

Duration: 15 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Outline how the Data Manager imports and exports data
- List the inputs to the Data Manager
- Describe the Data Manager configuration file
- Describe the structure of DMX files

### Exercise 10 – Import and Export Data

Duration: 30 minutes

### Lesson 3 – Application Navigation

Duration: 60 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Recognize the areas of the UI and the elements that can display in the application frame
- Relate the UI configuration files to the navigation features
- Outline the features for dynamically enabling or disabling menu and navigation options

## Exercise 11 – Configure Application Navigation

Duration: 110 minutes

### Unit 4 – Implementing Features for Robustness

#### Lesson 1 – Exception Handling

Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Describe how applications use exceptions to log errors
- Describe how to add localizeable messages by using the message catalog
- Explain how to add exception arguments

#### Exercise 12 – Add Exception Handling

Duration: 40 minutes

#### Lesson 2 – Validation

Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Explain the different types of validation in SPM
- Implement modeled validations
- Explain the purpose of the Informational Manager for server validation
- Describe how post-processing informational messages are returned
- Implement server validation by using the Informational Manager

#### Exercise 13 – Implement Modeled and Server-Side Validations

Duration: 80 minutes

#### Lesson 3 – Testing & Troubleshooting

Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Outline the support provided for JUnit testing
- Use the SPM test framework to write JUnit tests
- Describe SPM features for troubleshooting
- List basic troubleshooting tips

#### Exercise 14 – Perform Testing and Tracing

Duration: 40 minutes

### Unit 5 – Additional Server Features

## Lesson 1 – Modeling Operations

Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Describe and use the following modeling techniques:
- Code Packages
- Stereotyped operations
- Handcrafted SQL
- Review the processing types supported in SPM

## Exercise 15 – Implement the Non-Standard Stereotype

Duration: 60 minutes

## Lesson 2 – Modeling Associations

Duration: 20 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Describe and use the following modeling techniques:
  - Indices
  - Foreign Keys
  - Aggregations
  - Assignments

## Exercise 16 – Use Assignment and Aggregation

Duration: 40 minutes

## Lesson 3 – Additional Modeling Features

Duration: 20 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Describe table-level auditing
- Use exit points
- Generate unique IDs

## Exercise 17 – Add Packages, Exit Points, and Unique IDs

Duration: 60 minutes

## Unit 6 – Additional Client Features

### Lesson 1 – Additional UIM Features

Duration: 45 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Define the following features for UIM pages:
  - Multiple submit buttons
  - Select Lists
  - Reusable UIM
  - Define online help
  - Invoke the UIM Generator to create pages
  - Special menu types, including Wizards
  - Expandable lists
  - Editable lists
  - Images

### Exercise 18 – Add a Select List and Online Help

Duration: 45 minutes

### Lesson 2 – Adding Page Logic

Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:

- Outline how to add logic to your UIM pages using JSP scriptlets
- Describe how to invoke JavaScript functions based on user events
- Explain how to create JavaScript validations
- Describe how to implement static and dynamic conditional clusters using the CONDITION element

### Exercise 19 – Add a Conditional Cluster

Duration: 45 minutes

### Lesson 3 – Widgets

Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:

- List out-of-the-box domain specific controls or widgets
- Outline the use of the Custom Widget Development Framework (CWDF) for creating custom widgets
- Define what is meant by a pod

## Unit 7 – Course Conclusion

### Exercise 20 – Case Study: Implementation

Duration: 250 minutes

### Lesson 1 – Course Summary



**Duration: 5 minutes**

Learning  
objectives:

After completing this lesson, learners should be able to:

- Review the objectives of the course
- List the next options for SPM training