Merative Cúram for Developers (Customization) 8.X

CUR091

Course Description

This course introduces learners to the techniques that are used to customize Cúram out-of-the-box (OOTB) solutions for Cúram implementation projects.

The first part of the course describes the approach that Cúram uses for customizing OOTB applications compliantly. The main part of the course describes how to customize client and server artifacts, including source code and non-source code artifacts. The final part of the course outlines how SOAP and RESTful web services are used to integrate with external applications and how custom features can be implemented using events, deferred processes, and batch processes. In addition, the course outlines the approaches for localizing, supporting, and upgrading Cúram.

During the course, learners will perform impact analysis on OOTB applications and implement compliant customizations.

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 Cúram implementation projects. The course is also useful for anyone who needs a technical understanding of the Cúram Platform, including testers and support engineers.

Topics: The course covers the following topics:

- Interpreting the Cúram Reference Model and processing class architecture.
- Impact analysis using product documentation.
- Customizing client and server artifacts compliantly.
- Integrating with external applications by using SOAP and RESTful web services.
- Implementing events, deferred processes, and batch processes.
- Product releases, support, and upgrades.

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

- Describe the approaches for customizing Cúram artifacts.
- Perform impact analysis to determine the changes required for customizations.
- Customize out-of-the-box artifacts compliantly:
  - o Client artifacts.
  - o Non-source code server artifacts.
  - o Modeled application classes.
  - o Non-modeled application classes.
- Describe additional features for customizing Cúram:
  - o SOAP and RESTful web services for real-time integration.
  - o Events, deferred processes, and batch jobs.
  - o Localization.
- Briefly outline the installation, upgrade, and support processes.

Prerequisites: Before taking this course, students should have completed the following course:

- *Merative Cúram for Developers (ADE) 8.X*

Duration: 40 Hours (5 days ILT)

Skill Level: Intermediate

Version: This course was initially developed for Cúram 8.1.

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. Additional time is allowed for daily reviews and exercise reviews. Learners taking this course in a self-paced environment should allow more time to complete exercises.

<u>Unit 1 - Getting Started with Cúram Customization</u>

Lesson 1 – Course Introduction
Duration: 15 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 – Customization Overview
Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:
- Outline the typical stages of the Cúram implementation lifecycle.
- Describe the approaches for modifying Cúram to meet customer requirements.
- Outline the techniques for customizing source code and non-source code artifacts.

Exercise 1 – Reference Customization Resources
Duration: 20 minutes

Exercise 2 – Define a Custom Component
Duration: 40 minutes

Lesson 3 – Analyzing the Reference Application
Duration: 20 minutes

Learning objectives: After completing this lesson, learners should be able to:
- Interpret Cúram Reference Models using entity-relationship diagrams (ERDs)
- Describe how the Cúram Analysis Document Generator (CADG) is used to analyze artifacts in the reference application.

Exercise 3 – Run the Reference Application
Duration: 20 minutes

Exercise 4 – Analyze the Reference Application
Duration: 60 minutes

<u>Unit 2 - Customizing Client Artifacts</u>

Lesson 1 – Customizing Pages, Navigation, and Properties
Duration: 35 minutes

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

- Describe how to customize:
  - UIM pages.
  - Application navigation files.
  - Property files, images, and style sheets.
- Describe the purpose of the key client configuration files and how to customize them.
- Outline the following features: session management, browser management, and auto-recovery.

Exercise 5 – Customize Pages and Navigation
Duration: 50 minutes

Lesson 2 – Custom Widgets
Duration: 45 minutes

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

- Outline how domain plug-ins are used.
- Briefly describe how to develop widgets by using the following tools:
  - Custom Widget Development Framework (CWDF).
  - UI Add-Ons Development Environment.
- List options and considerations when developing widgets.
- Briefly describe how to define pods.
- List resources for developing widgets and pods.

Exercise 6 – Custom Widgets
Duration: 50 minutes

<u>Unit 3 - Customizing Server Artifacts</u>

Lesson 1 – Customizing Non-Source Code Artifacts
Duration: 30 minutes

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

- Describe the approach for customizing the main server configuration files.
- Describe how to customize message files, code tables, and data manager files.
- Describe how to enable and disable configurable validations.

Exercise 7 – Customize Server Artifacts (Non-Source Code)
Duration: 90 minutes

Exercise 8 – Extract Configuration Data
Duration: 40 minutes

Lesson 2 – The Processing Class Architecture
Duration: 15 minutes

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

- Describe the processing class architecture and the purpose of each of its constituent layers.

Exercise 9 – Navigate the Processing Class Architecture (Modeled Classes)
Duration: 60 minutes

Lesson 3 – Customizing Modeled Classes
Duration: 40 minutes

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

- Identify Cúram APIs that you can reference from your code.
- Outline the techniques for customizing source code artifacts:
  o Structs.
  o Façade and process classes.
  o Entity classes.
  o Domain Definitions.
  o Relationships.

Exercise 10 – Customize Modeled Classes
Duration: 170 minutes

Unit 4 - Customizing Non-Modeled (Handcrafted) Classes

Lesson 1 – Introduction to Handcrafted Interfaces
Duration: 30 minutes

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

- Outline how service layer classes developed using PI differ from those developed using the classic development approach.
- Explain the purpose of the entity and entity DAO interfaces.
- List components that use handcrafted service layer interfaces.

Exercise 11 – Navigate the Processing Class Architecture (Non-Modeled Classes)
Duration: 60 minutes

Lesson 2 – Customizing using Strategy
Duration: 35 minutes

Learning objectives: After completing this lesson, learners should be able to:
- Explain how Cúram uses Guice for dependency injection.
- Interpret the Cúram documentation to select the appropriate approach for customizing handcrafted interfaces.
- List the techniques for customizing handcrafted interfaces.
- Outline how handcrafted implementations can be replaced using the Strategy pattern.

Exercise 12 – Customize Handcrafted Service Layer Classes Using Strategy
Duration: 70 minutes

Lesson 3 – Customizing using Events
Duration: 30 minutes

Learning objectives: After completing this lesson, learners should be able to:
- Provide an overview of event handling in components developed using PI.
- Explain how business and persistence events are used for customizing.
- Describe how to add attributes to PI entities.
- Describe how to customize service layer implementations by using the Event/Listener pattern.

Exercise 13 – Customize Handcrafted Service Layer Classes Using Events
Duration: 70 minutes

Lesson 4 – Referencing Handcrafted APIs
Duration: 15 minutes

Learning objectives: After completing this lesson, learners should be able to:
- Identify where handcrafted service layer APIs can be called from.
- Compare the approach for accessing data using the classic approach and accessing data using the PI approach.

Exercise 14 – Reference Handcrafted Service Layer Classes
Duration: 65 minutes

Unit 5 - Additional Tools for Customization

Lesson 1 – Integrating Using SOAP and RESTful Web Services
Duration: 35 minutes

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

- Describe the support that Cúram provides for SOAP web services.
- Outline the steps for creating inbound and outbound SOAP web services in Cúram.
- Outline the steps for developing REST APIs in Cúram.

Exercise 15 – Add a Simple REST API (Optional)
Duration: 90 minutes

Lesson 2 – Additional Processing Types
Duration: 35 minutes

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

- List the processing types supported by Cúram.
- Outline the operation of deferred processes.
- List the steps for developing deferred processes.
- Explain how to use EVX events to implement the Observer pattern.
- Describe the support for batch processing in Curam.

Exercise 16 – Use Additional Processing Types (Optional)
Duration: 60 minutes

Lesson 3 – Localization
Duration: 30 minutes

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

- Outline the support Cúram provides for configuring different languages and regions.
- List the key configuration files that are used to regionalize your installation.
- Describe how multilingual support operates.

Exercise 17 – Update Localizeable Fields
Duration: 30 minutes

Lesson 4 – Support Tools and Resources
Duration: 30 minutes

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

- Explain the numbering format for identifying Cúram releases.
- List the main steps for installing the development environment.
- List documentation resources for understanding Cúram releases.
- Briefly describe the upgrade process and tools.
- Describe how to identify deprecated artifacts.

Exercise 18 – Gather Troubleshooting Data
Duration: 45 minutes

<u>Unit 6 - Course Conclusion</u>

Lesson 1– Case Study
Duration: 10 minutes

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

- Describe the purpose and approach for completing the case study.

Exercise 19 – Case Study: Customize Service Supplier
Duration: 300 minutes

Lesson 2 – 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 Cúram training.

<u>Appendix</u>

After reviewing the appendix, learners should be able to:

- Use examples to describe how to implement the following features:
  - Custom data conversion and sorting.
  - SOAP and REST APIs
  - Deferred processing
  - EVX events
  - Batch streaming and chunking.
- Describe how to set optimistic locking and table-level auditing in Cúram.