

Control-M 22: Fundamentals Automation API Developing


[Learning Path >](#)
Course Code: AUTO-CMAP-9220

Modality	Duration	Applicable Versions	Target Audience
Instructor-Led Training (ILT)	1 Day	Control-M 9.0.22.000	Automation API Developers

Course Overview

By accessing the capabilities of Control-M via its Automation Application Programming Interface (API) from the Developers' Self-Service Portal, developers can work with workflow orchestration, further enhancing the self-service nature of the company's DevOps processes.

This one-day instructor-led training covers basic concepts of Control-M Automation API, how to write job definitions in JSON format, working with environments, validating and executing job definitions file, and how to review the job definitions in Control-M environment.

In addition to this, it also talks about deploy and run services. Finally, students will also learn about advanced job definitions and the use of deploy descriptor.

Prerequisites

- Control-M 22: Overview (WBT)
- Control-M 22: Concepts for Operators and Schedulers (WBT)
- Control-M 21.x: Comparing Web Interface with Client Interface (WBT)

Recommended Trainings

- Control-M 21.x: Fundamentals Using (WBT)
- Control-M 22: Fundamentals Operating (ILT/ASP)
- Schedulers: Control-M 22: Fundamentals Scheduling (ILT/ASP) **OR**
- Administrators: Control-M 22: Fundamentals Administering (ILT/ASP)

Learning Objectives

- Describe the Control-M architecture and Control-M Roles
- Access the Control-M web interface
- Understand the purpose and usage of the Control-M Automation API
- Explain the Control-M Automation API Services
- Use the Environment, Session, Build, Run and Deploy Services
- Use JSON to create Control-M Jobs-as-Code
- Use a Deploy Descriptor to modify code to comply with different environment standards
- Understand how to build a CI/CD pipeline to automate code deployment

Course Modules

Module 1: Getting Started With Control-M Automation API

- Explain the purpose of, and differentiate between Control-M and Control-M SaaS
- List the interfaces that can be used to access Control-M
- Explain the purpose of Control-M Automation API
- Explain the purpose of the Control-M Automation API Services
- Access Control-M Automation API documentation
- Install the Automation API command line interface
- Generate an API Token
- Access and use the Control-M Swagger UI
- Use the Environment Service to connect an Automation API command line interface environment to the Control-M/EM
- Use the Session Service to login to Control-M and create a session token

Module 2: Developing Jobs Using Automation API

- Explain the process to build, deploy and run Jobs-as-Code
- Explain what Control-M Workbench is, and how to install it
- Explain key Control-M concepts:
 - Folders and jobs
 - Workspaces
 - New Day Process (NDP)
 - Viewpoints
- Define folders and jobs in the Planning domain of Control-M
- Define folders and jobs in JSON code
- Use the Build Service to validate JSON code

Module 3: Running Jobs

- Use the Run Service to run folders and jobs in a Control-M environment
- Use the Deploy Service to deploy folders and jobs to a Control-M environment

Module 4: Writing Jobs-as-Code

- Define Defaults in code, to simplify Jobs-as-Code creation
- Define dependencies between folders and jobs
- Define folder and job schedules
- Utilize the Control-M Python Client to define folders and jobs

Module 5: Using the Deploy Descriptor

- Understand and use Automation API Deploy Descriptors
- Assign, Replace and Add attributes when modifying code with a Deploy Descriptor
- Use JSON Path to identify attributes
- Filter which jobs/folders should be modified by using ApplyOn

Discount Options

Have multiple students? Contact us to discuss hosting a private class for your organization.

Contact us for additional information 