



Course Data Sheet

OO320 – Operations Orchestration 10.x Advanced Authoring

Course No.: OO320-105	Category/Sub Category: Operations Management/Cloud and Automation
For software version(s): 10.5 – 10.6 Software version used in the labs: 10.51 and/or 10.60	Course length: Three days (one extra day with CloudSlang)
Delivery formats: Instructor Led (ILT) and Virtual Instructor Led (VILT)	Training is available as a private session onsite.
To order visit: http://h20546.www2.hp.com/main/US/	

Course Description

This three-day course introduces students to advanced concepts of flow development and usage of the HPE Operations Orchestration (OO) software along with some integration. OO is part of HPE Cloud Automation solutions.

This course introduces best practices related to content authoring, structuring, and promotion. The execution order internal to OO is explained in detail. This course further focuses on advanced topics including using scriptlets, using Representational State Transfer (REST) to interact with OO Central, using Simple Object Access Protocol (SOAP) to interact with web services and using semaphores. A number of topics dive deeper into the use of OO to automate tasks, such as creating PDFs and interfacing with Subversion (SVN) from within OO flows.

A separate, full day topic discusses the use of CloudSlang, an open source, textual authoring tool that integrates with OO.

The course consists of focused, task-oriented lectures, text, and a series of detailed hands-on labs to teach the course material to the student. The hands-on labs use version 10.51 of the software as well as version 10.60.

Audience/Job Roles

This course is designed for Workflow developers, System Administrators, Automation Operators, DevOps integrators, and other personnel responsible for the implementation of OO.

Course Objectives

Upon successful completion of this course, you should be able to:

- Describe how to align flow authors to a common way to develop, document, package, and deliver OO 10.x content.
- Describe how OO executes workflows in terms of component execution order.
- Execute scriptlet methods in OO to manage flow data and flow execution

- Summarize the concepts related to semaphores and how to use them in OO
- Describe the use of the Source Control Management (SCM) tool and perform advanced tasks
- Extend OO through the use of external libraries
- Use the Web Services wizard and create flows using the newly generated operations
- Describe and use REST calls
- Write flows for Cloud Service Automation (CSA) lifecycle actions
- Describe integration content
- Create workflows using CloudSlang and execute them on OO Central

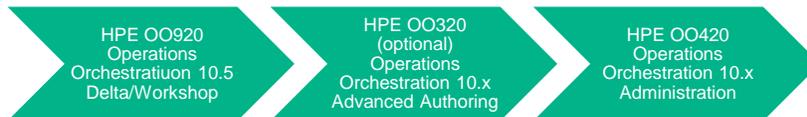
Prerequisites/Recommended Skills

To be successful in this course, you should have the following prerequisites or knowledge.

- *OO220 – Operations Orchestration 10.x Flow Development*
- Networking terms and concepts
- Web browsers and telnet or Secure Shell (SSH) connection methods
- Different operating system environments

Learning Path

Existing OO 9.x customers:



Existing OO10.x customers:



New OO customers:



Certification

This course supports the following certification exam:

- OO 10.x ASE

Course Topics

Modules	Objectives
Module 1: Course Overview	<ul style="list-style-type: none"> • Introductions • Class schedule and class logistics • Related courses • Documentation • Lab environment details
Module 2: Best Practices for Content Authoring, Structuring, and Promotion	<ul style="list-style-type: none"> • Align flow authors to a common way to develop, document, package, and deliver OO 10.x content
Module 3: Understanding Execution Order	<ul style="list-style-type: none"> • Describe how OO executes workflows in terms of component execution order • Describe how OO assigns execution priority to the various component types
Module 4: Scriptlets	<ul style="list-style-type: none"> • Describe scriptlets and their use in OO • Execute OO scriptlet methods to manage flow data and flow execution • Analyze sample flows that use scriptlets • Execute scriptlets in flows
Module 5: Using Semaphores	<ul style="list-style-type: none"> • Summarize the concept of semaphores and how to use semaphores in OO • Describe sample flows
Module 6: SCM Management	<ul style="list-style-type: none"> • Use the Source Control Management (SCM) tool and perform advanced tasks that you might encounter when projects and items are shared among multiple authors
Module 7: Generate PDFs	<ul style="list-style-type: none"> • Extend OO with PDF creation functionality and use it
Module 8: Web Services	<ul style="list-style-type: none"> • Explain how OO uses web services • Use the Web Services wizard to convert a Web Service Description Language (WSDL) file into OO operations • Explain how the imported web service content works • Create a flow that interoperates with an external web service provider • Use filters to extract data elements from a web service (SOAP) response • Use other operations to interoperate (HTTP RAW Post and Invoke Method 2) with web service providers
Module 9: REST Calls	<ul style="list-style-type: none"> • Explain how OO uses RESTful services • Use the REST wizard to convert a Web Application Description Language (WADL) file into OO operations • Explain how the imported content works • Describe useful operations to treat and manipulate the content using REST wizards
Module 10: Writing Flows for CSA	<ul style="list-style-type: none"> • Explain how OO and CSA communicate back and forth • List the required flow inputs • Use the <code>Get Artifact Properties</code> flow to request info from CSA • Use the <code>Update Service Component</code> operation to report back to CSA

Module 11: Integration Content (Server Automation and Service Manager)	<ul style="list-style-type: none">• Explain how OO integrates other HPE and ISV software, such as Server Automation (SA) and Service Manager (SM)• Describe the integration of content pack structures• List the OO content that interacts with SA and SM
Module 12: CloudSlang	<ul style="list-style-type: none">• Describe Git• Describe CloudSlang• Use CloudSlang standalone and within OO