

Title

OpenEdge 11 Developer Learning Path

Duration

Equivalent to 112 hours of learning.

Audience

OpenEdge Developers

Description

The OpenEdge 11 Developer Learning Path includes the following courses and assessment exams:

- Introduction to Progress OpenEdge
- Introduction to Progress Developer Studio for OpenEdge
 - Introduction to Progress Developer Studio for OpenEdge : Assessment Exam
- Introduction to Progress Application Server for OpenEdge for Developers
 - Introduction to Progress Application Server for OpenEdge for Developers : Assessment Exam
- Introduction to the Progress OpenEdge Reference Architecture
- Developing a Progress OpenEdge ABL Application
 - Developing a Progress OpenEdge ABL Application: Assessment Exam
- Using ProDataSets
 - Using ProDataSets: Assessment Exam
- Introduction to Object-oriented Programming
 - Introduction to Object-oriented Programming
- Building a Progress OpenEdge Multi-Tenant Application
 - Building a Progress OpenEdge Multi-Tenant Application: Assessment Exam
- Accessing Data in Progress OpenEdge Partitioned Tables
 - Accessing Data in Progress OpenEdge Partitioned Tables: Assessment Exam
- Developing Business Entities for Progress OpenEdge Data Object Services
- Creating Progress OpenEdge Data Object Services
- Introduction to Progress OpenEdge Integration
- Providing Progress OpenEdge Applications as REST Web Applications
- Accessing REST Services from a Progress OpenEdge ABL Application
- Using Corticon Business Rules in a Progress OpenEdge Application

See a visual representation of this learning path [here](#).

Version compatibility

This course is compatible with Progress OpenEdge 11.

What you will learn

This path includes the following tracks to help you grow your OpenEdge developer skills from basic to advanced:

- **OpenEdge ABL Application Development Core:** In this track, new developers are introduced to Progress OpenEdge and its core concepts and tools, including the Progress OpenEdge Developer Studio, the Progress Application Server for OpenEdge, and the Progress OpenEdge Reference Architecture. You can also learn about the Progress Advanced Business Language (ABL), Progress Software Corporation's proprietary language designed for developing business applications. It is a high-level language that supports procedural, dynamic, and object-oriented programming. You learn how to develop an OpenEdge application using Progress Developer Studio for OpenEdge. You also learn how to incrementally develop an application that utilizes OERA design principles and how to write code using recognized best practices.
- **Technology Jumpstart:** In this track, experienced developers continue their journey by learning to identify when and where to use datasets and then practice incorporating datasets into applications during hands-on exercises using Progress Developer Studio for OpenEdge. You learn how to define, populate, and perform data operations using datasets and then share data with other applications. You also learn how to use the object-oriented capabilities of Progress OpenEdge ABL. You are introduced to key features of Progress' object-oriented ABL, how to develop and test classes, and how to further develop an application by applying object-oriented programming features such as inheritance, interface classes, singletons, dynamic instances, and events.
- **Modernizing an OpenEdge application:** In this track, back-end ABL developers learn how to create Data Object Service and develop business entities.
- **Integration:** In this track, developers are introduced to Progress integration, as well as providing OpenEdge applications as a REST web application. You also learn how to access a REST service from a Progress ABL application.
- **Special topics:** In this track, developers can learn how to use Progress Corticon Business Rules in a Progress OpenEdge application. You can

also learn how to build an OpenEdge multi-tenant application and access data in partitioned tables.

Prerequisites

Before you begin the learning path, you should meet the following prerequisites:

- Basic knowledge of what an application is as well as the concepts of client and server parts of an application.
 - Experience with a high-level programming language.
 - Experience with managing services and programs in a Windows environment.
-