



Key Data

Course #: 8419A

Number of Days: 4

Format: Instructor-Led

Certification Exams:

- Microsoft Axapta Programming Exam
- Microsoft CRM Professional Exam

Certification Track: None

This course syllabus should be used to determine whether the course is appropriate for the students, based on their current skills and technical training needs.

Course content, prices, and availability are subject to change without notice.

Course Syllabus

Microsoft® Axapta® Development III

Elements of this syllabus are subject to change.

Introduction

This 4 day course introduces the student to advanced development in Microsoft® Axapta® using X++ and development tools. This course begins with learning the development environment and tools including application architecture, debugging, performance monitoring and application upgrade. The student will then learn more about Fetch of data from the database, Classes, Forms and Reports. The exercises included in this course will focus on typical additions to and modifications of the standard application. This course is meant to be a more advanced course covering practical development in Microsoft Axapta.

Audience

This course is intended for individuals who will be developing within Microsoft Axapta using X++. This audience typically includes technical consultants who will be working with Microsoft Axapta to develop customizations and modifications to meet clients' needs. This course will be most beneficial for someone who is familiar with the concepts of object oriented programming, SQL and basic programming in X++. Additionally, consultants who are responsible for training or supporting the customer will benefit from this course.

At Course Completion

After completing this course, students will be able to:

- Use the integrated development tools in Microsoft Axapta.
- Examine, analyze and understand the structures of the standard application.
- Monitor the application with respect to execution and performance.
- Optimizing the database access
- Design and create a solution with respect to best practices.
- Create advanced forms with filtering, sorting, aggregation and joins.
- Use of advanced form controls as Tables, Tree, ActiveX and Window.
- Create advanced reports containing consolidated data from multiple sources.

Prerequisites

It is required that students have general knowledge on the following subjects:

- Object Oriented Analysis, Design and Programming
- Database Design and SQL

Please refer to the section *Suggested reading* to obtain references covering the subjects above.

It is required that students have completed:

- Development II: Introduction to X++

It is assumed that the student has been working with development in Microsoft Axapta and has gained some experience on the topics covered by Development I & II.

For a referral to a Microsoft Certified Partner for Learning Solutions in your area, see the Microsoft Training and Certification Web site at <http://www.microsoft.com/traincert>. Call your local Microsoft Certified Partner Learning Solution for more information and to register for classes.

Student Materials

The student kit includes a comprehensive workbook and other necessary materials for this class.

The following is provided in the student kit:

- Development III: Advanced X++ Course Manual

Chapter 1: Development Environment and Tools

This chapter gives a comprehensive foundation for the use of the development environment and integrated tools.

Sections
<p>Development environment</p> <ul style="list-style-type: none">• Files (Client, Application, Server)• Transfer of modifications to test environment• AOS setup (Prefer development in 3-tier thin environment) <p>Application architecture</p> <ul style="list-style-type: none">• X-references• Application hierarchy browser• MorphXPlorer <p>Performance/Monitoring</p> <ul style="list-style-type: none">• Debugger• System Monitoring• Method Trace• Development Warnings• SQL Trace• Query Time Limit• Deadlocks• Database Trace• Code Profiler <p>Application upgrade</p> <ul style="list-style-type: none">• Upgrade project• Compare tool
Exercises
<ul style="list-style-type: none">• Use the tools to analyze the structures of the standard application.• Find the bottleneck in a specific function in the standard application• Make an upgrade of a project containing a few modified application objects.

After completing this chapter, students will:

- be able to make a transfer of modifications from the development environment to test or live environment
- be able to browse, examine, analyze and understand the structures of the standard application

- be able to use the monitoring tools and interpret the related reporting
- understand the upgrade of modifications from one version or service pack to another version or service pack

Chapter 2: Fetch of Data

This chapter focuses on fetching of data from the database. The goal is to make optimal communication with the database with respect to performance. In addition, the student will have knowledge in why and how the use of temporary tables.

Sections
<ul style="list-style-type: none">• Query (Structure, Properties)• Cost based optimizer• while select (Syntax, Qualifiers, Field lists, Join, Placeholders/Literals)• Caching• Locking• Temporary tables
Exercises
<ul style="list-style-type: none">• Build a query in the AOT and with X++• Create a fetch using a query and a while select• Improve performance by implementing display-method caching• Reduction of locking• Monitor the impact of the use of field list, aggregation and caching• Use of multiple instances of a temporary table.

After completing this chapter students will be able to:

- program optimal database access
- understand the concept of literals and placeholders
- understand the caching mechanisms in Microsoft Axapta
- use temporary tables in classes, forms and reports

Chapter 3: Classes

This chapter will teach the student to be able to create and modify classes and methods, according to Best Practice using object oriented concepts.

Sections
Object Oriented implementation in Microsoft Axapta Most important differences from the standard application and C#, VB.NET and Java examples
<ul style="list-style-type: none">• Classdeclaration• Variables• Data members• Methods (qualifiers, parameters and return value)• Overriding/Overloading• Access modifiers• Types of Microsoft Axapta Classes (Examples in parenthesis)

Sections
<p>Application classes</p> <ul style="list-style-type: none">• Basic functionality (Dialog)• Wrapping of general routines (NumberSequence)• Form support classes (LedgerJournalEngine)• Table "polymorphism" (SalesTableType)• API to the modules (LedgerVoucher, CustVoucher)• Batch processing (RunBase) <p>Kernel classes</p> <ul style="list-style-type: none">• Tools (Binary, DialogBox, RecordSortedList)• Application objects (FormDataSource, QueryBuildRange,...)• Integration (COM, ODBC, IO)• System classes (xApplication, xCompany, xInfo)• Runbase framework• Args-objects
Exercises
<ul style="list-style-type: none">• Design an abstract class• Design and implementation of an interface.• Implementation of accessor methods• Implementation of foundation classes• Use of kernel classes to access metadata• Implementation of exception handling• Design and implement a new class hierarchy using polymorphism.• Create a new class using the Runbase-framework.• Modify the class LedgerExchAdj to include a query and to initialize the query from the current selected account in the form.

After completing this chapter, students will be able to:

- understand the implementation of the object oriented concept in Microsoft Axapta
- create a new class according to Best Practice
- understand and implement a solution using polymorphism
- use the RunBase-framework to create new batch processes
- transfer information using the args-object

Chapter 4: Forms

This chapter will teach the student how to create and modify advanced forms.

Sections
<ul style="list-style-type: none">• Architecture• AOT (general methods, datasource, design)• Kernel classes and object model• Properties• Best practice (Initializing via args, Design, Validation / Modification)• Placement of code (Priority: Table, Support class, Datasource, Design)• Top ten methods to use• Handles to the objects in forms• Interaction between form and RunBase classes• Queries. Synchronizing (Active/Passive), Joins, Modify by code (Filtering, Sorting, Aggregation)• Special controls in short (ActiveX, List view, Table, Tree, Window)• Special forms (Lookup, Wizard)• Activate a form from X++ using ClassFactory
Exercises
<ul style="list-style-type: none">• Modify the form with customer transactions to toggle between active and passive synchronizing.• Create a form with ledger transactions to include dynamic modification of filtering.• Modify the employee form to include a web browser control.• Create a lookup form to select an item which includes a picture of the item.

After completing this chapter, students will be able to:

- understand the structure of a form and the underlying kernel classes
- make decisions of where to place the code
- use the top-ten methods in forms
- dynamically modify the fetch of data

Chapter 5: Reports

This chapter will teach the student how to fetch data into reports based on queries or X++ code.

Sections
<ul style="list-style-type: none">• Architecture• AOT (general methods, data sources, design)• Kernel classes and object model• Top ten methods to use• Handles to the objects in reports• The RunBaseReportStd framework• Fetch data by Query and X++• Templates / Autodesign / Design• Use of temporary tables for consolidation• Calculation on (sub)totals• Report Wizard

Exercises

- Create a report using aggregate functions.
- Create a report using a temporary table to consolidate data from multiple sources.
- Create a report which dynamically modifies of some controls.
- Create a report with calculations on totals.

After completing this chapter, students will be able to:

- understand the structure of a report and the underlying kernel classes
- make decisions of where to place the code
- implement a dialog using the RunBaseReportStd framework
- understand the principles of fetching data in a reports
- make dynamically changes to the design
- use temporary tables for consolidation

Appendix A: Solutions

This Appendix contains all the answers to the chapter exercises, Test Your Knowledge, and Test Your Skills sections.

Suggested reading

It is required that students as a part of the prerequisites have a knowledge level on the following subjects as described in this literature:

Object Oriented Analysis and Design

- Object-Oriented Modeling and Design, James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen; Prentice Hall International Editions; ISBN: 0-13-630054-5

Database design and SQL

The Essence of Databases, F.D Rolland, Prentice Hall; ISBN: 0-13-727827-6