

MB6-894

Development, Extensions, and Deployment for Microsoft Dynamics 365 Finance

Exam number: MB6-894

Exam title: Development, Extensions, and Deployment for Microsoft Dynamics 365 Finance

Publish date:

Language(s) this exam will be available in:

Audience: (Sales and Presales Roles, IT professionals, Developers, Information workers, etc.):

Technology:

Credit type (example: MCSA):

Exam provider (VUE, Certiport, or both):

The content of this exam was updated on February 4, 2020. Please continue scrolling to the red line section below to view the changes.

Exam Design

Audience Profile

Candidates for this exam are technical consultants, programmers, or IT personnel who implement the basic technical and development tasks required to customize Microsoft Dynamics 365 Finance and Microsoft Dynamics 365 Commerce.

Candidates should have a basic knowledge of Microsoft Dynamics 365 Finance and Microsoft Dynamics 365 Commerce. Additionally, experience with Visual Studio, and object-oriented programming is required.

Candidates for this development exam should have a moderate understanding of programming concepts and processes, as well as an understanding of Finance and Commerce modifications, enhancements, and new development projects that apply to the different business process scenarios.

Skills measured

Understand the Architecture and Development Environment (20-25%)

Describe the Microsoft Dynamics 365 Finance architecture

May include but is not limited to:

- Describe the development environment
- Describe the components in the application stack
- Identify cloud architecture components
- Explain the server architecture
- Describe the layer architecture

Describe the Microsoft Dynamics 365 Commerce architecture

May include but is not limited to:

- Identify components of the retail solutions and their purpose
- Describe differences and uses for the Retail Server and the Retail Store Scale Unit (RSSU)
- Describe hardware components
- Describe and create components for the Modern Point of Sale (MPOS) and the Cloud Point of Sale (Cloud POS)

Manage development with Microsoft Visual Studio and Lifecycle Services (LCS)

May include but is not limited to:

- Identify the windows and basic navigation of Visual Studio
- Describe differences between and uses for projects, models, solutions, and packages
- Describe and perform the build process
- Describe the features of LCS
- Use LCS to deploy hotfixes, manage assets, and complete the build process

Develop new elements by using Application Explorer (20-25%)

Create and manage labels and resources

May include but is not limited to:

- Create new label files
- Create and use labels
- Describe elements and uses for labels
- Identify best practices for labels
- Identify uses for resources

Create and manage data types

May include but is not limited to:

- Describe uses for base enumerations
- Create new base enumerations

- Identify best practices for base enumerations
- Describe primitive and extended data types
- Create new extended data types
- Identify key properties for extended data types
- Implement best practices for extended data types

Create and manage tables

May include but is not limited to:

- Identify the components of a table
- Describe various types of table relationships
- Describe various types of table indexes
- Implement best practices for tables, relationships, and indexes

Read and Write Basic X++ (20-25%)

Describe the X++ environment

May include but is not limited to:

- Identify the characteristics of X++
- Use the code editor to write X++
- Describe the features of IntelliSense
- Identify basic syntax for X++
- Describe the use of common key words in X++
- Describe the features of the debugger
- Use the comparison tool
- Use the best practice checker

Work with X++ control statements

May include but is not limited to:

- Work with variables, operators, conditional statements, and loops
- Use built-in functions
- Use communication tools

Work with classes

May include but is not limited to:

- Create new classes
- Describe scoping events and parameters
- Describe inheritance
- Identify key best practices when writing X++

Manipulate data with X++

May include but is not limited to:

- Identify techniques for data retrieval
- Explain uses for transaction integrity checking
- Insert, update, and delete records using X++
- Identify best practices for manipulating data with X++

Manage exceptions in X++

May include but is not limited to:

- Identify exception types
- Use try/catch statements
- Describe the features of optimistic concurrency control
- Describe best practices for exception handling with X++

Manage the User Interface and Security for Developers (20-25%)

Manage the user interface

May include but is not limited to:

- Identify components of various form patterns
- Identify common form patterns and sub patterns
- Identify key properties for form elements
- Apply and validate form patterns
- Describe uses for tiles
- Describe how to join data in forms
- Implement best practices for form development

Manage the user navigation

May include but is not limited to:

- Identify types of menu items
- Create menu items and menus
- Identify key properties for menu items
- Describe uses for menu items
- Implement best practices for menus and menu items

Manage security in the development environment

May include but is not limited to:

- Describe the security architecture
- Describe the components of role-based security
- Create new privileges, duties, and roles in the application explorer
- Describe the extensible data security framework

Develop and extend retail components (15-20%)

Modify retail components

May include but is not limited to:

- Describe the components of the Commerce Runtime (CRT) and how to extend the CRT
- Describe the components of the Commerce Data Exchange (CDX) and how to extend the CDX
- Identify components of the retail architecture that can be extended
- Describe how to extend various components of the retail architecture

Deploy retail modifications

May include but is not limited to:

- Describe how to package and deploy retail modifications into a source environment
- Review and apply retail hot fixes
- Create and deploy packages for retail
- Resolve conflicts and rebuild packages

Monitor retail components

May include but is not limited to:

- Identify, use and extend diagnostics and monitoring tools for retail components
- Identify, setup, and monitor data distribution and offline databases
- Understand hot, warm, and cold paths for diagnostics and monitoring
- Use event logs, and LCS to monitor and diagnose issues with retail components

See below changes as of February 4, 2020...

Exam Design

Audience Profile

Candidates for this exam are technical consultants, programmers, or IT personnel who implement the basic technical and development tasks required to customize Microsoft Dynamics 365 ~~for Finance and Operations~~ Microsoft Dynamics 365 for RetailCommerce.

Candidates should have a basic knowledge of Microsoft Dynamics 365 ~~for Finance and Operations~~ and Microsoft Dynamics 365 for RetailCommerce. Additionally, experience with Visual Studio, and object-oriented programming is required.

Candidates for this development exam should have a moderate understanding of programming concepts and processes, as well as an understanding of Finance and ~~Operations and RetailCommerce~~ modifications, enhancements, and new development projects that apply to the different business process scenarios.

Skills measured

Understand the Architecture and Development Environment (20-25%)

Describe the Microsoft Dynamics 365 ~~for Finance and Operations~~ architecture

May include but is not limited to:

- Describe the development environment
- Describe the components in the application stack

- Identify cloud architecture components
- Explain the server architecture
- Describe the layer architecture

Describe the Microsoft Dynamics 365 ~~for Retail~~Commerce architecture

May include but is not limited to:

- Identify components of the retail solutions and their purpose
- Describe differences and uses for the Retail Server and the Retail Store Scale Unit (RSSU)
- Describe hardware components
- Describe and create components for the Modern Point of Sale (MPOS) and the Cloud Point of Sale (Cloud POS)

Manage development with Microsoft Visual Studio and Lifecycle Services (LCS)

May include but is not limited to:

- Identify the windows and basic navigation of Visual Studio
- Describe differences between and uses for projects, models, solutions, and packages
- Describe and perform the build process
- Describe the features of LCS
- Use LCS to deploy hotfixes, manage assets, and complete the build process

Develop new elements by using Application Explorer (20-25%)

Create and manage labels and resources

May include but is not limited to:

- Create new label files
- Create and use labels
- Describe elements and uses for labels
- Identify best practices for labels
- Identify uses for resources

Create and manage data types

May include but is not limited to:

- Describe uses for base enumerations
- Create new base enumerations
- Identify best practices for base enumerations
- Describe primitive and extended data types
- Create new extended data types
- Identify key properties for extended data types
- Implement best practices for extended data types

Create and manage tables

May include but is not limited to:

- Identify the components of a table
- Describe various types of table relationships

- Describe various types of table indexes
- Implement best practices for tables, relationships, and indexes

Read and Write Basic X++ (20-25%)

Describe the X++ environment

May include but is not limited to:

- Identify the characteristics of X++
- Use the code editor to write X++
- Describe the features of IntelliSense
- Identify basic syntax for X++
- Describe the use of common key words in X++
- Describe the features of the debugger
- Use the comparison tool
- Use the best practice checker

Work with X++ control statements

May include but is not limited to:

- Work with variables, operators, conditional statements, and loops
- Use built-in functions
- Use communication tools

Work with classes

May include but is not limited to:

- Create new classes
- Describe scoping events and parameters
- Describe inheritance
- Identify key best practices when writing X++

Manipulate data with X++

May include but is not limited to:

- Identify techniques for data retrieval
- Explain uses for transaction integrity checking
- Insert, update, and delete records using X++
- Identify best practices for manipulating data with X++

Manage exceptions in X++

May include but is not limited to:

- Identify exception types
- Use try/catch statements
- Describe the features of optimistic concurrency control
- Describe best practices for exception handling with X++

Manage the User Interface and Security for Developers (20-25%)

Manage the user interface

May include but is not limited to:

- Identify components of various form patterns
- Identify common form patterns and sub patterns
- Identify key properties for form elements
- Apply and validate form patterns
- Describe uses for tiles
- Describe how to join data in forms
- Implement best practices for form development

Manage the user navigation

May include but is not limited to:

- Identify types of menu items
- Create menu items and menus
- Identify key properties for menu items
- Describe uses for menu items
- Implement best practices for menus and menu items

Manage security in the development environment

May include but is not limited to:

- Describe the security architecture
- Describe the components of role-based security
- Create new privileges, duties, and roles in the application explorer
- Describe the extensible data security framework

Develop and extend retail components (15-20%)

Modify retail components

May include but is not limited to:

- Describe the components of the Commerce Runtime (CRT) and how to extend the CRT
- Describe the components of the Commerce Data Exchange (CDX) and how to extend the CDX
- Identify components of the retail architecture that can be extended
- Describe how to extend various components of the retail architecture

Deploy retail modifications

May include but is not limited to:

- Describe how to package and deploy retail modifications into a source environment
- Review and apply retail hot fixes
- Create and deploy packages for retail
- Resolve conflicts and rebuild packages

Monitor retail components

May include but is not limited to:

- Identify, use and extend diagnostics and monitoring tools for retail components

- Identify, setup, and monitor data distribution and offline databases
- Understand hot, warm, and cold paths for diagnostics and monitoring
- Use event logs, and LCS to monitor and diagnose issues with retail components