

Course 10265A:

Developing Data Access Solutions with Microsoft Visual Studio 2010

Exam 70-516:

Duration: 40 Hours (5 Days)

About this Exam

This exam is designed to test the candidate's knowledge and skills on accessing data sources using ADO.NET and the .NET Framework.

Questions that contain code will be presented in either VB or C#. Candidates can select one of these languages when they start the exam.

Audience Profile

Candidates for this exam use Microsoft Visual Studio 2010 and ADO.NET 4 to develop the data access layer of an application. Candidates should have a minimum of two to three years of experience developing application components that interact with a variety of data sources. In addition, candidates should have a thorough understanding of relational database concepts and should have a minimum of one to two years of experience using a relational database management system.

Candidates should also have a good grasp of the following:

- ADO.NET 4 coding techniques and framework components
- ADO.NET Data Services LINQ
- LINQ to SQL
- Entity Framework technologies
- Structured Query Language (SQL)
- stored procedures
- Database Structures/Schemas (Objects) XML

Credit Toward Certification When you pass Exam 70-516: TS: Accessing Data with Microsoft .NET Framework 4, you complete the requirements for the following certification(s):

- MCTS: .NET Framework 4, Data Access

Exam 70-516: TS: Accessing Data with Microsoft .NET Framework 4: counts as credit toward the following certification(s):

- MCPD: Windows Developer 4
- MCPD: Web Developer 4

Note This preparation guide is subject to change at any time without prior notice and at the sole discretion of Microsoft. Microsoft exams might include adaptive testing technology and simulation items. Microsoft does not identify the format in which exams are presented. Please use this preparation guide to prepare for the exam, regardless of its format.

Prerequisites

Before attending this course, students must have:

- An understanding of the problem-solving techniques that apply to software development, including the following principles of software development:
 - Modern software development models
 - Typical phases of a software development lifecycle
 - Concepts of event-driven programming
 - Concepts of object-oriented programming
 - Creating use-case diagrams
 - Designing and building a user interface
 - Developing a structured application
- A basic understanding of the following scripting techniques and some hands-on experience writing scripts:
 - Web scripting techniques
 - Macro scripting techniques
 - Windows scripting techniques
- A general understanding of the purpose, function, and features of following .NET Framework topics:
 - Common Language Runtime
 - .NET Framework class library
 - Common Type System
 - Component interoperation
 - Cross-language interoperability
 - Assemblies in the Common Language Runtime
 - Application domains
 - Runtime hosts supported by the .NET Framework
- Experience using Visual Studio 2008 in the following task areas:
 - Declaring and initializing typed variables using the Camel case naming convention
 - Using arithmetic, relational, and logical operators in code statements
 - Using branching statements to control code execution
 - Using looping statements to iterate through collections or repeat steps until a specified condition is met
 - Creating classes and methods to establish the basic structure of an application
 - Using methods and events to implement the programming logic of an application
 - Identifying syntax and logic errors
 - Accessing and managing data from a data source
- Experience in object oriented design and development as follows:
 - Creating and accessing classes and class properties
 - Creating and accessing methods and overloaded methods
 - Implementing inheritance, base classes, and abstract classes
 - Declaring, raising, and handling events
 - Responding to and throwing exceptions
 - Implementing interfaces and polymorphism
 - Implementing shared and static members
 - Implementing generics
 - Creating components and class libraries
- Experience in N-Tier application design and development as follows:
 - Managing a software development process
 - Controlling input at the user interface level in Windows client and Web applications
 - Debugging, tracing, and profiling .NET applications
 - Monitoring and logging .NET applications
 - Implementing basic testing best practices
 - Performing basic data access tasks with LINQ
 - Basics of LINQ to XML
 - Basics of LINQ to Entities
 - Basics of LINQ to SQL

- Implementing basic security best practices in .NET Applications
 - Basics of Code Access Security
 - Basics of Role-Based Security
 - Basics of Cryptography Services
- Implementing basic service calls
 - Basics of creating and consuming XML Web Services
 - Basics of creating and consuming WCF Services
- Using .NET Configuration Files
- Deploying .NET Framework Applications using Click Once and the MS Installer
- Data access experience in Windows client application development as follows:
 - Connect to a data source
 - Implement data binding
 - Implement data validation at the UI layer
- Data access experience in Web application development as follows:
 - Connect to a data source
 - Implement dynamic data
 - Implement data validation at the UI layer

Course Outline

Module 1: Architecture and Data Access Technologies

This module describes the commonly used data access technologies and scenarios in which you are likely to use them.

Lessons

- Data Access Technologies
- Data Access Scenarios

Lab: Analyzing Data Access Scenarios

- Identifying Data Access Technologies

After completing this module, students will be able to:

- Describe the key data access technologies available to .NET Framework developers.
- Assign appropriate data access technologies to common data access scenarios.

Module 2: Building Entity Data Models

This module introduces the concepts of data modeling, and in particular, Entity Data Models (EDMs). It explains how you can use EDMs to decouple the conceptual data structure in your applications from the logical data structure in the data store.

Lessons

- Introduction to Entity Data Models
- Modifying the Entity Data Model
- Customizing the Entity Data Model

Lab: Using Entity Data Models

- Generating an EDM from the Adventure Works Database
- Adding Entities and Associations
- Using the Generate Database Wizard
- Mapping Entities to Multiple Tables
- Implementing an Inheritance Hierarchy
- Using Stored Procedures
- Creating a Complex Type

After completing this module, students will be able to:

- Describe and create an Entity Data Model.
- Modify an Entity Data Model by using the Entity Designer.
- Customize a model to meet their business requirements.

Module 3: Querying Entity Data

This module explains how to query an entity data model by using common methods such as LINQ to Entities, Entity SQL, and the classes in the Entity Client namespace.

Lessons

- Retrieving Data by Using LINQ to Entities
- Retrieving Data by Using Entity SQL
- Retrieving Data by Using Entity Client Provider
- Retrieving Data by Using Stored Procedures
- Unit Testing Your Data Access Code

Lab: Querying Entity Data

- Retrieving All Contact Entities
- Retrieving Contact Entities by Using a Filter
- Retrieving Rewards Claimed Entities
- Querying the Rewards Family of Entities
- Executing a Stored Procedure

After completing this module, students will be able to:

- Retrieve data by using LINQ to Entities.
- Retrieve data by using the Entity SQL language.
- Retrieve data by using the Entity Client Provider.
- Retrieve data by using stored procedures in the entity model.
- Create unit tests for their data access code.

Module 4: Creating, Updating, and Deleting Entity Data This module introduces you to the ways that the Entity Framework enables you to modify the data in your database. You apply changes to the entities managed by the Object Context class. The Object Context class is responsible for tracking all changes to entities and then persisting these changes to the database on request. **Lessons**

- Understanding Change Tracking in the Entity Framework
- Modifying Data in an Entity Data Model

Lab: Creating, Updating, and Deleting Entity Data

- Maintaining Contact and Reward Data
- Maintaining Rewards Claim Data

After completing this module, students will be able to:

- Describe how the Entity Framework implements change tracking.
- Describe how to modify data in the entity model, and persist the changes to the database.

Module 5: Handling Multi-User Scenarios by Using Object Services

This module introduces the concurrency model that the Entity Framework uses to address the issues faced by applications that must support multiple users who access the same data simultaneously. It also describes how the Entity Framework can make use of transactions to ensure data integrity.

Lessons

- Handling Concurrency in the Entity Framework
- Transactional Support in the Entity Framework

Lab: Handling Multi-User Scenarios by Using Object Services

- Handling Concurrency of Rewards Claimed Data
- Updating the Rewards Claimed and ArchivedRewardsClaimed Information by Using a Transaction

After completing this module, students will be able to:

- Describe the optimistic concurrency model that the Entity Framework uses.
- Manage transactions in applications that use the Entity Framework.

Module 6: Building Optimized Solutions by Using Object Services

This module explains best practices for designing and building a scalable, optimized data access layer by using Object Services. The module introduces several techniques that can be used to optimize the performance of queries that execute against the conceptual model.

Lessons

- The Stages of Query Execution
- Change Tracking and Object Materialization
- Using Compiled Queries
- Using Design-Time Generated Entity Framework Views
- Monitoring Performance
- Performing Asynchronous Data Modifications

Lab : Building Optimized Solutions by Using Object Services

- Improving the Performance of Query Operations
- Improving the Performance of Update Operations

After completing this module, students will be able to:

- Explain how the Entity Framework executes queries.
- Understand the impact of tracking and object materialization on query performance.
- Describe how to use compiled queries.
- Describe how to use design-time generated views.
- Describe how to monitor query performance.
- Describe how to perform asynchronous data modifications.

Module 7: Customizing Entities and Building Custom Entity Classes

This module describes how to customize and extend entities with your own business logic.

Lessons

- Overriding Generated Classes
- Using Templates to Customize Entities
- Creating and Using Custom Entity Classes

Lab : Customizing Entities and Building Custom Entity Classes

- Using a Template to Add Custom Functionality to Entity Classes
- Creating Custom Entity Classes

After completing this module, students will be able to:

- Use partial classes and methods to add business logic to generated code.
- Create and use templates to customize code generation.
- Modify existing business classes to take advantage of entity functionality.

Module 8: Using POCO Classes with the Entity Framework

This module introduces the ways in which you can define custom entity classes in your Entity Framework application. By default, Microsoft Visual Studio generates a set of entity classes for you from the Entity Data Model (EDM). Instead of these generated classes, you may want to use an existing set of "plain old" CLR objects (POCO) business classes in your application. You can also extend the generated entity classes to add custom business functionality to your entity objects.

Lessons

- Requirements for POCO Classes
- POCO Classes and Lazy Loading
- POCO Classes and Change Tracking
- Extending Entity Types

Lab: Using POCO Classes with the Entity Framework

- Using POCO Classes
- Extending Your POCO Classes

After completing this module, students will be able to:

- List the requirements that their POCO classes must meet.
- Create POCO entities that support automatic lazy loading.
- Create POCO entities that support automatic change tracking.

- Describe the options for using interfaces and inheritance to create custom entity objects.

Module 9: Building an N-Tier Solution by Using the Entity Framework

This module explains how to address the architectural issues that can arise when building an N-Tier enterprise application by using the Entity Framework.

Lessons

- Designing an N-Tier Solution
- Defining Operations and Implementing Data Transport Structures
- Protecting Data and Operations

Lab : Building an N-Tier Solution by Using the Entity Framework

- Creating the Contacts and Orders Data Access Tier
- Protecting Data Access Operations

After completing this module, students will be able to:

- Describe the issues and strategies that are relevant to building n-tier applications.
- Understand the key components that they must create in order to implement an n-tier application.
- Describe how to protect operations and data in an n-tier application.

Module 10: Handling Updates in an N-Tier Solution by Using the Entity Framework

This module describes how you can handle data modifications in an n-tier solution. The module describes the different strategies for handling modifications that you should use for the different alternative formats for transporting data between tiers: data transfer objects (DTOs), self-tracking entities (STEs), and simple entities (SEs). The module also describes how to manage the exceptions that can occur during the data modification process.

Lessons

- Tracking Entities and Persisting Changes
- Managing Exceptions in an N-Tier Solution

Lab: Handling Updates in an N-Tier Solution by Using the Entity Framework

- Handling Updates in the Data Access Tier
- Detecting and Handling Order Conflicts

After completing this module, students will be able to:

- Describe strategies that they can adopt for tracking changes in the client application, and persisting those changes in the database.
- Describe how to trap and handle update and concurrency exceptions in the an n-tier solution.

Module 11: Building Occasionally Connected Solutions

This module describes how to access offline or occasionally connected data in client applications.

Lessons

- Offline Data Caching by Using XML
- Using the Sync Framework

Lab: Building Occasionally Connected Solutions

- Modifying the Orders Application to Use Offline XML Data
- Modifying the Orders Application to Synchronize Locally Cached Data

After completing this module, students will be able to:

- Cache data in local XML files by using LINQ to XML.
- Implement an occasionally connected application by using the Microsoft Sync Framework.

Module 12: Querying Data by Using WCF Data Services Windows Communication Foundation (WCF) Data Services enable you to create highly flexible data services that can be used to provide access to data across the Internet or a corporate network. You can access these services by using REST-style URIs, and they can be easily consumed by a wide variety of applications. As WCF Data Services are built on top of standard Internet protocols such as HTTP and the Atom Publishing Protocol, they are an ideal choice for delivering data to AJAX applications and Rich Interactive Applications built using technologies such as Microsoft Silverlight. **Lessons**

- Introduction to WCF Data Services
- Creating a WCF Data Service
- Consuming a WCF Data Service
- Protecting Data and Operations in a WCF Data Service

Lab: Creating and Using WCF Data Services

- Exposing Order Data as a WCF Data Service
- Consuming a WCF Data Service
- Restricting Access to Data Exposed by a WCF Data Service
- Implementing a Business Operation in a WCF Data Service

After completing this module, students will be able to:

- Describe the purpose and features of WCF Data Services.
- Expose data by using a WCF Data Service.
- Implement a client application that can consume a WCF Data Service.
- Grant and restrict access to resources exposed by a WCF Data Service.

Module 13: Updating Data by Using WCF Data Services this module describes how to use WCF Data Services to create, update, and delete data. WCF Data Services use standard internet protocols such as HTTP and the Atom Publishing Protocol to enable update access to data across the Internet or a corporate network. **Lessons**

- Creating, Updating, and Deleting Data in a WCF Data Service
- Preventing Unauthorized Updates and Improving Performance
- Using WCF Data Services with Nonrelational Data

Lab: Updating Data by Using WCF Data Services

- Updating Entities by Using a WCF Data Service
- Creating and Deleting Entities by Using a WCF Data Service
- Restricting Create, Update, and Delete Requests

After completing this module, students will be able to:

- Create, update, and delete entities by using a WCF Data Service.
- Control access to data modification functionality, and improve performance by batching commands together.
- Use WCF Data Services to access and modify nonrelational data.

Module 14: Using ADO.NET

ADO.NET is a highly flexible framework for building applications that require access to data held in a data source. This module introduces ADO.NET and explains how you can use it to develop scalable, high-performance, data-driven applications.

Lessons

- Retrieving and Modifying Data by Using ADO.NET Commands
- Retrieving and Modifying Data by Using Datasets
- Managing Transactions and Concurrency in Multiuser Scenarios

Lab: Using ADO.NET

- Using ADO.NET to Retrieve Read-Only Information Quickly and Perform Simple Data Modifications
- Developing the Product List Web Application
- Enabling Data Modifications

After completing this module, students will be able to:

- Retrieve and update data by using ADO.NET commands and stored procedures.
- Retrieve and update data by using Dataset objects.
- Implement transactions and handle concurrency exceptions.

Module 15: Using LINQ to SQL

ADO.NET provides a mechanism that enables you to build applications that can query and maintain data that is held in a variety of sources in a database-agnostic manner. However, building applications by using ADO.NET requires that you are familiar with the Structured Query Language (SQL) language and features of the database management system that you are connecting to. Language-Integrated Query (LINQ) to SQL provides a higher-level abstraction for managing data that is held in a Microsoft SQL Server database, and is an ideal stepping stone for migrating ADO.NET applications toward the ADO.NET Entity Framework. This module introduces LINQ to SQL and explains how you can use it to abstract the low-level details of ADO.NET queries by developing against a logical data model.

Lessons

- Implementing a Logical Data Model by Using LINQ to SQL
- Managing Performance and Handling Concurrency

Lab: Using LINQ to SQL

- Using LINQ to SQL to Build a Data Access Layer
- Updating a Database by Using a Stored Procedure
- Building a Custom Entity Class

After completing this module, students will be able to:

- Design a logical data model by using LINQ to SQL.
- Manage performance by using LINQ to SQL, and handle concurrency.

