

Microsoft 2541

Core Data Access with Microsoft Visual Studio 2005

Length: 3 Days

Introduction:

Elements of this syllabus are subject to change.

This three-day instructor-led workshop provides students with the knowledge and skills to develop data-access applications by using the Microsoft .NET Framework and Microsoft Visual Studio 2005. The workshop describes how to access and update data in a database by using ADO.NET. The workshop also describes how to create, read, and write XML data.

Audience

This workshop is intended for corporate and Independent Software Vendor (ISV) application developers who have a desire to learn more about specific technology areas in distributed application development.

At Workshop Completion:

After completing this workshop, students will be able to:

- Connect to databases and read data.
- Query and update databases by using commands.
- Perform transactional operations.
- Perform disconnected operations programmatically.
- Perform disconnected operations by using Visual Studio 2005 wizards.
- Perform XML operations on disconnected data.
- Read and write XML data.
- Process XML data by using the Document Object Model DOM.

Prerequisites:

Before attending this workshop, students must:

- Be able to manage a solution environment using the Visual Studio 2005 integrated development environment IDE and tools
- Understand the .NET Framework 2.0 and the Common Language Runtime
- Be able to program an application using a .NET Framework 2.0 compliant language
- Know how to make assemblies available to other applications
- Have a good understanding of basic relational database concepts
- Be able to use basic SQL commands
- Have a good understanding of XML including XML declaration, elements, attributes, and namespaces
- Be able to explain XML schema concepts

Workshop Outline:

Unit 1: Connecting to Databases and Reading Data

This unit introduces the fundamental skills required to connect to a database and read data from the database. It describes how to use ADO.NET data providers to connect to various different kinds of databases, and how to execute a query that returns a scalar value from the database. It also describes how to use connection pooling to achieve scalability, and how to handle connection events and exceptions.

Lessons

- What Is ADO.NET?
- The Process for Connecting to a Database and Reading Data
- What Is Connection Pooling?

Lab 1: Connecting to Databases and Reading Data

- Exercise 1. Connecting to a Database

- Exercise 2. Saving a Connection String Securely in an Application Configuration File
- Exercise 3. Retrieving Data from a Database
- Exercise 4. Handling Connection Events and Exceptions
- Exercise 5. Configuring and Using Connection Pooling (if time permits)

After completing this unit, students will be able to:

- Describe the key features of ADO.NET.
- Explain the process for connecting to a database and reading data from a database.
- Describe the benefits of connection pooling.
- Connect to a database.
- Save a connection string securely in an application configuration file.
- Retrieve data from a database.
- Handle connection events and exceptions.

Unit 2: Querying and Updating Databases by Using Commands

This unit describes how to create and run commands that return a scalar value, return a result set, update data in the database, or update the schema of the database. It also describes how to create and run parameterized commands.

Lessons

- ADO.NET Commands
- The Process for Passing Parameters into Commands

Lab 2: Querying and Updating Databases by Using Commands

- Exercise 1. Creating and Running Query Commands
- Exercise 2. Creating and Running Parameterized Commands
- Exercise 3. Creating and Running Update Commands
- Exercise 4. Using Globalized Formats for Storing and Accessing Data (if time permits)

After completing this unit, students will be able to:

- List the different types of ADO.NET commands.
- Explain the process for creating and executing parameterized commands.
- Create and run query commands.
- Create and run parameterized commands.
- Create and run update commands.

Unit 3: Performing Transactional Operations

This unit describes how to manage transactions in a .NET application at the middle tier. It describes how to write ADO.NET code to start, commit, and rollback local transactions. It also describes how to specify an appropriate isolation level for a transaction, and how to enlist in distributed transactions.

Lessons

- What Is a Transaction?
- The Process for Managing Local Transactions
- The Process for Managing Distributed Transactions
- Isolation Levels

Lab 3: Performing Transactional Operations

- Exercise 1. Performing Data Updates within a Transaction
- Exercise 2. Choosing an Appropriate Isolation Level for a Transaction
- Exercise 3. Enlisting in a Distributed Transaction
- Exercise 4. Managing Transactions in the Data Tier (if time permits)

After completing this unit, students will be able to:

- Describe the concept of transactional operations.
- Explain the process for managing a local transaction.
- Explain the process for managing a distributed transaction.
- Describe the effect of setting the isolation level for a transaction.
- Perform data updates within a transaction.
- Choose an appropriate isolation level for a transaction.
- Enlist in a distributed transaction.

Unit 4: Performing Disconnected Operations Programmatically

This unit describes how to create and use DataSets programmatically. It describes how to create DataSet, DataTable, and DataColumn objects, how to populate a DataSet manually, and how to load and save data by using a DataAdapter. It also describes how to create in-memory views on data by using a DataView.

Lessons

- What Is the ADO.NET Disconnected Model?
- The Process for Loading and Saving Data in a DataSet

- What Are DataViews?

Lab 4: Performing Disconnected Operations Programmatically

- Exercise 1. Creating a DataSet Programmatically
- Exercise 2. Populating and Saving a DataSet
- Exercise 3. Adding, Modifying, and Deleting Data in a DataSet
- Exercise 4. Merging DataSets
- Exercise 5. Creating and Using DataViews (if time permits)

After completing this unit, students will be able to:

- Describe the ADO.NET disconnected data model.
- Explain the process for loading and saving data in a DataSet.
- Describe how to use DataViews to provide a view on data.
- Create a DataSet programmatically.
- Populate and save a DataSet.
- Add, modify, and delete data in a DataSet.
- Merge DataSets.

Unit 5: Performing Disconnected Operations by Using Visual Studio 2005 Wizards

This unit describes how to create typed DataSets, DataAdapters, and TableAdapters by using the TableAdapter Configuration Wizard and the Data Source Configuration Wizard in Visual Studio 2005. It also describes how to write type-safe code to access data in a typed DataSet.

Lessons

- Comparing Untyped DataSets with Typed DataSets
- What Are Table Adapters?
- Demonstration: Creating a Typed DataSet by Using Visual Studio 2005 Wizards

Lab 5: Performing Disconnected Operations by Using Visual Studio 2005 Wizards

- Exercise 1. Creating a Typed DataSet by Using the DataSet Designer
- Exercise 2. Loading, Displaying, and Saving Data in a Typed DataSet
- Exercise 3. Adding Code to a Typed DataSet
- Exercise 4. Creating a Typed DataSet by Using the Data Source Configuration Wizard
- Exercise 5. Adding Queries to a Table Adapter (if time permits)

After completing this unit, students will be able to:

- Explain the differences between untyped DataSets and typed DataSets.
- Explain how table adapters work in conjunction with data tables in a typed DataSet.
- Describe the process for creating a typed DataSet by using Visual Studio 2005 Wizards.
- Create a typed DataSet by using the DataSet Designer.
- Load, display, and save data in a typed DataSet.
- Add code to a typed DataSet.
- Create a typed DataSet by using the Data Source Configuration Wizard.

Unit 6: Performing XML Operations on Disconnected Data

This unit describes how to read and write DataSets in XML format. It describes how to read and write data only, schema only, or a combination of the two. It also describes how to read and write a DataSet as a DiffGram, so that modifications to the data can be retained when the DataSet is serialized to XML format.

Lessons

- XML Representations of DataSets
- What Are DiffGrams?

Lab 6: Performing XML Operations on Disconnected Data

- Exercise 1. Saving a DataSet as XML Data
- Exercise 2. Loading a DataSet from XML
- Exercise 3. Saving and Loading DataSet Schema Information

After completing this unit, students will be able to:

- Describe scenarios in which you can represent a DataSet as XML data.
- Explain how a DiffGram retains information about modifications to a DataSet.
- Save a DataSet as XML data.
- Load a DataSet from XML.
- Save and load DataSet schema information.

Unit 7: Reading and Writing XML Data

This unit describes how to use the XmlReader and XmlWriter classes to serially read and write XML data. It describes how to read elements, attributes, and text content in an XML document, and perform validation against an XML

schema. It also describes how to create an XML document and write elements, attributes, namespace declarations, and text content.

Lessons

- The Process for Serially Reading XML Data
- The Process for Serially Writing XML Data

Lab 7: Reading and Writing XML Data

- Exercise 1. Writing XML Data by Using XmlWriter
- Exercise 2. Reading XML Data by Using XmlReader

After completing this unit, students will be able to:

- Describe the process for serially reading XML data.
- Describe the process for serially writing XML data.
- Write XML data by using XmlWriter.
- Read XML data by using XmlReader.

Unit 8: Processing XML Data by Using DOM

This unit describes how to process XML data in memory by using the Document Object Model (DOM). It describes how to load an XML document into a DOM tree, and validate the XML document against an XML schema. It also describes how to read existing XML content, modify XML content, and save the DOM tree to an XML document.

Lessons

- What Is DOM?
- What Are DOM Trees?
- Types of XML Nodes in a DOM Tree

Lab 8: Processing XML Data by Using DOM

- Exercise 1. Reading XML Data by Using DOM
- Exercise 2. Writing XML Data by Using DOM

After completing this unit, students will be able to:

- Explain what DOM is and describe how to process XML data in memory by using DOM.
- Explain what a DOM tree is and describe how to represent XML documents as a DOM tree.
- List the types of nodes in a DOM tree.
- Read XML data by using DOM.
- Modify XML data by using DOM.