



## **Course 2542: Advanced Data Access with Microsoft Visual Studio 2005**

### **Introduction**

This two-day instructor-led workshop provides students with the knowledge and skills needed to use advanced data access features and techniques in the Microsoft .NET Framework and Microsoft Visual Studio 2005. The workshop describes how to access data and how to implement database functionality by using Microsoft ADO.NET 2.0 and Microsoft SQL Server 2005. The workshop also describes how to locate, edit, and transform XML by using XPath and Extensible Style sheet Language for Transformations (XSLT).

**Duration:** 2 Days

### **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 Course Completion**

After completing this course, students will be able to:

- Minimize and handle database operation conflicts.
- Handle large objects.
- Enhance database performance.
- Create managed code objects for SQL Server 2005.
- Query XML by using XPath.
- Transform XML by using XSLT style sheets.
- Minimize and handle database operation conflicts.
- Handle large objects.
- Enhance database performance.
- Create managed code objects for SQL Server 2005.
- Query XML by using XPath.
- Transform XML by using XSLT style sheets.
- Minimize and handle database operation conflicts.

## Prerequisites

Before attending this course, students must have:

- Have attended or studied Workshop 2541, Core Data Access with Microsoft Visual Studio 2005, or possess equivalent knowledge and skills.
- Know how to use delegates.
- Be able to use database stored procedures, triggers, and aggregates.
- Be able to explain XPath concepts and basic syntax.
- Be able to explain XSLT style sheet concepts.
- Have attended or studied Workshop 2541, Core Data Access with Microsoft Visual Studio 2005, or possess equivalent knowledge and skills.
- Know how to use delegates.
- Be able to use database stored procedures, triggers, and aggregates.

## Course Outline

### Unit 1: Minimizing and Handling Database Operation Conflicts

This unit describes how to minimize data access conflicts, and how to handle these conflicts when they occur. It describes how to implement optimistic concurrency in the ADO.NET disconnected model, and how to implement optimistic concurrency by using the various isolation levels available in SQL Server 2005.

#### Lessons

- Why Do Data Conflicts Arise?
- Isolation Levels Available in SQL Server 2005
- Guidelines for Using SQL Server 2005 Isolation Levels

### Lab 1: Minimizing and Handling Database Operation Conflicts

- Exercise 1. Reading Committed Data by Using Locks
- Exercise 2. Reading Committed Data by Using Statement-Level Snapshots
- Exercise 3. Reading Committed Data by Using Transaction-Level Snapshots
- Exercise 4. Handling Data Concurrency by Using ADO.NET

After completing this unit, students will be able to:

- Explain why data conflicts arise.
- Describe the isolation levels that are available in SQL Server 2005.
- Describe the guidelines for using SQL Server 2005 isolation levels.
- Read committed data by using locks.
- Read committed data by using statement-level snapshots.
- Read committed data by using transaction-level snapshots.

- Handle data concurrency by using ADO.NET.

## **Unit 2: Handling Large Objects**

This unit describes how to read and write large values efficiently to a SQL Server database. It describes how to read large binary values and large text values by using `SequentialAccess` for a `SqlDataReader`. It also describes how to write large binary values and large text values, and how to conserve resources when writing large values.

### **Lessons**

- What Are Binary Large Objects and Character Large Objects?
- The Process for Reading Large Objects from a Database
- The Process for Writing Large Objects to a Database

### **Lab 2: Handling Large Objects**

- Exercise 1. Reading Large Values from SQL Server
- Exercise 2. Writing Large Values to SQL Server
- Exercise 3. Conserving Resources When Writing Large Values to SQL Server

After completing this unit, students will be able to:

- Describe binary large objects and character large objects.
- Explain the process for reading large objects from a database.
- Explain the process for writing large objects to a database.
- Read large values from SQL Server.
- Write large values to SQL Server.
- Conserve resources when writing large values to SQL Server.

## **Unit 3: Enhancing Database Performance**

This unit describes how to enhance database performance by using new features available in ADO.NET 2.0. The unit describes how to perform asynchronous data operations, create multiple active result sets, perform batch updates, and perform bulk copies.

### **Lessons**

- ADO.NET Enhancements in the .NET Framework 2.0
- SQL Server Provider Statistics

### **Lab 3: Enhancing Database Performance**

- Exercise 1. Accessing Multiple Result Sets Concurrently
- Exercise 2. Performing Asynchronous Data Access Operations
- Exercise 3. Performing a Batch Update
- Exercise 4. Performing a Bulk Data Copy

After completing this unit, students will be able to:

- Describe the ADO.NET enhancements in Microsoft .NET Framework 2.0.
- Describe the support for run-time statistics in the .NET Framework Data Provider for SQL Server.
- Access multiple result sets concurrently.
- Perform asynchronous data access operations.
- Perform a batch update.
- Perform a bulk data copy.

#### **Unit 4: Creating Managed Code Objects for SQL Server 2005**

This unit describes how to create database objects for SQL Server 2005 in a .NET Framework programming language. It describes how to create stored procedures, triggers, user-defined functions, aggregates, and user-defined types in managed code. Additionally, it describes how to deploy an assembly that contains managed objects into SQL Server 2005, and how to declare database objects to reference the managed objects.

##### **Lessons**

- The Benefits of Creating Managed Code Objects
- Demonstration: The Process for Importing an Existing Assembly into SQL Server 2005
- Demonstration: The Process for Implementing Managed Code Objects in SQL Server 2005

#### **Lab 4: Creating Managed Code Objects for SQL Server 2005**

- Exercise 1. Creating Managed Stored Procedures and Triggers
- Exercise 2. Creating Managed User-Defined Functions
- Exercise 3. Creating a Managed Aggregate
- Exercise 4. Creating a Managed User-Defined Type
- Exercise 5. Importing Existing Assemblies into SQL Server 2005 (if time permits)

After completing this unit, students will be able to:

- Describe the benefits of creating managed code objects for SQL Server 2005.
- Explain the process for importing an existing assembly into SQL Server 2005.
- Explain the process for implementing managed code objects in SQL Server 2005.
- Create managed stored procedures and triggers.
- Create managed user-defined functions.
- Create a managed aggregate.
- Create a managed user-defined type.

## **Unit 5: Querying XML by Using XPath**

This unit describes how to use XPath in a .NET Framework application. It describes how to create an XPathNavigator object on an XML document, and how to locate content and evaluate expressions by using the XPathNavigator object. Additionally, it describes how to edit XML data by using the XPathNavigator object.

### **Lessons**

- The XPath Data Model
- The Process for Selecting and Editing XML Data by Using XPathNavigator
- The Process for Evaluating XPath Expressions by Using XPathNavigator

### **Lab 5: Querying XML by Using XPath**

- Exercise 1. Selecting XML Data by Using XPathNavigator
- Exercise 2. Evaluating XPath Expressions by Using XPath Navigator
- Exercise 3. Creating and Using Compiled XPath Expressions
- Exercise 4. Editing XML Data by Using XPathNavigator

After completing this unit, students will be able to:

- Describe the XPath data model.
- Explain the process for selecting and editing XML data by using XPathNavigator.
- Explain the process for evaluating XPath expressions by using XPathNavigator.
- Select XML data by using XPathNavigator.
- Evaluate XPath expressions by using XPathNavigator.
- Create and use compiled XPath expressions.
- Edit XML data by using XPathNavigator.

## **Unit 6: Transforming XML by Using XSLT Style Sheets**

This unit describes how to transform XML documents in a .NET Framework application. It introduces how to load an XSLT style sheet in an application, and how to execute the style sheet to transform an XML document. The unit also describes how to pass parameters into a style sheet, and how to create and use extension objects.

### **Lessons**

- What Is XSLT?
- The Process for Executing an XSLT Style Sheet
- What Are Extension Objects?

### **Lab 6: Transforming XML by Using XSLT Style Sheets**

- Exercise 1. Transforming an XML Document by Using an XSLT Style Sheet
- Exercise 2. Resolving External Resources During XSLT Processing
- Exercise 3. Passing Parameters into an XSLT Style Sheet

- Exercise 4: Creating and Using Extension Objects

After completing this unit, students will be able to:

- Describe the purpose of XSLT.
- Explain the process for executing an XSLT style sheet.
- Describe the purpose of extension objects.
- Transform an XML document by using an XSLT style sheet.
- Resolve external resources during XSLT processing.
- Pass parameters into an XSLT style sheet.
- Create and use extension objects.