ADO.NET Using C#

Course Summary

Description

This three-day intensive course teaches the essential elements of ADO.NET such that at the end of the course the programmer is able to utilize its tremendous database manipulation powers to build effective database applications. The course includes a major case study demonstrating the use of ADO.NET in a realistic setting. It is current to .NET 4.6, Visual Studio 2017 and SQL Server 2016.

The course opens with an overview of ADO.NET and its relation to previous Microsoft data access technologies. It includes a discussion of ADO.NET architecture, main interfaces and classes, and programming with both the connected and disconnected models. The database for the case study is introduced.

The next two chapters cover in detail Connection and Command objects, which are essential in both connected and disconnected database access scenarios. The following chapter covers DataReaders, which provide a fast, forward-only reading capability. Programming with DataReaders bears a close resemblance to programming with the vintage recordset object.

Then the course focuses on the backbone of ADO.NET: DataSet and its related classes, such as DataAdapter, DataTable, DataRow, DataColumn, DataRelation, TableMappings and ColumnMappings. DataSet is able to handle multiple tables while remaining disconnected. It is eminently suited for building highly scalable applications for the Web. The close relationship between ADO.NET and XML is covered in detail. Transactions and concurrency are covered.

Additional features of ADO.NET are covered, including asynchronous operations, multiple active result sets and bulk copy. The last chapter covers Language Integrated Query (LINQ) and Microsoft's Entity Framework.

There are numerous example program implemented in a multiple-tier architecture, with separate tiers for data access and user interface. The course uses the LocalDB version of SQL Server 2016 Express, which is recommended by Microsoft for developers.

The essentials of ADO.NET do not depend on the user interface layer, which may be a Console program, a Windows program or a Web program. The examples in this course use Console and Windows Forms clients. Appendix B discusses some special considerations for Web clients. Sample programs are provided illustrating ADO.NET with ASP.NET Web Forms clients.

Objectives

After taking this course, students will be able to:

- Understand the architecture and main classes of ADO.NET
- Gain fluency in programming ADO.NET using C#
- Gain a thorough understanding of the use of disconnected DataSets for building highly scalable applications
- Acquire a working knowledge of the tight coupling of XML with ADO.NET
- Learn how to use additional features in ADO.NET, including asynchronous operations, multiple active result sets, and bulk copy
- Acquire a working knowledge of LINQ and the Entity Framework
- Implement a realistic case study that ties together many concepts of ADO.NET in a practical demonstration

ADO.NET Using C#

Course Summary (cont'd)

Topics

- Introduction to ADO.NET
- ADO.NET Connections
- ADO.NET Commands
- DataReaders and Connected Access
- Data Sets and Disconnected Access
- More About DataSets
- XML and ADO.NET
- Concurrency and Transactions
- Additional Features
- LINQ and Entity Framework

Audience

This course is designed for those wanting to learn the essential elements of ADO.NET.

Prerequisites

Before taking this course, students should have a basic knowledge of SQL and of programming the .NET Framework using C#. The student should also understand the fundamentals of XML. To get full benefit from the examples in the course, some knowledge of Windows Forms is desirable.

Duration

Three days

ADO.NET Using C#

Course Outline

I. Introduction to ADO.NET
- A. Microsoft Data Access Technologies
- B. From ADO to ADO.NET
- C. ADO.NET Architecture
- D. Namespaces and Classes
- E. Interfaces
- F. DataSets and Disconnected Access
- G. First ADO.NET Programs
- H. Acme Computer Case Study

II. ADO.NET Connections
- A. .NET Data Providers
- B. IDbConnection
- C. Connection Classes
- D. Connection Strings
- E. Connection Pooling
- F. Connection Events
- G. Error Handling

III. ADO.NET Commands
- A. IDbCommand
- B. Command Objects
- C. Creating Commands
- D. Executing Commands
- E. Parameterized Queries
- F. Command Types
- G. Using Stored Procedures
- H. Batch Queries

IV. DataReaders and Connected Access
- A. DataReaders
- B. IDataReader
- C. IDataRecord
- D. Type-Safe Accessors
- E. Null Columns
- F. ExecuteReader Options
- G. Multiple Result Sets
- H. Obtaining Schema Information

V. Data Sets and Disconnected Access
- A. DataSet
- B. DataAdapter
- C. Command Objects
- D. DataTable
- E. DataColumn
- F. DataRow
- G. Row States and Versions
- H. Accept or Reject Changes
- I. DataTable Events
- J. Updating a Database from a DataSet
- K. Command Builders

VI. More About DataSets
- A. Filtering DataTables
- B. Multiple Table DataSets
- C. Schema
- D. Constraints
- E. Relations
- F. Navigating a DataSet
- G. DataMapping
- H. Creating a DataSet Programmatically

VII. XML and ADO.NET
- A. Strong Coupling Between ADO.NET and XML
- B. Rendering XML from a DataSet
- C. Controlling XML Output
- D. Reading XML into a DataSet
- E. XML Schema and DataSets
- F. Typed DataSets
- G. Table Adapters
- H. Synchronizing DataSets and XML
- I. XML Serialization

VIII. Concurrency and Transactions
- A. DataSets and Concurrency
- B. Optimistic Concurrency
- C. Pessimistic Concurrency
- D. Handling Concurrency Violations
- E. ADO.NET Transactions
- F. Database Transactions

IX. Additional Features
- A. Asynchronous Database Operations
- B. Multiple Active Result Sets
- C. Bulk Copy

X. LINQ and Entity Framework
- A. Language Integrated Query (LINQ)
- B. Bridging Objects and Data
- C. Using Object Relational Designer
- D. Filtering, Ordering and Aggregation
- E. Inserts, Deletes and Updates
- F. LINQ to SQL
- G. LINQ to DataSet
- H. ADO.NET Entity Framework
- I. LINQ to Entities

XI. Appendix A. Acme Computer Case Study

XII. Appendix B. ADO.NET Web Programming

XIII. Appendix C. Learning Resources

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically