



## ADO.NET for Web Applications using VB.NET (BDT4221) 3 days

### Course Description

This three-day intensive course teaches the essential elements of ADO.NET for Web applications 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 3.5, Visual Studio 2008 and SQL Server 2005/2008.

### Learning Objectives

- Understand the architecture and main classes of ADO.NET
- Gain fluency in programming ADO.NET using Visual Basic
- 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 newer features in ADO.NET 3.5 with Visual Studio 2008 and SQL Server 2005/2008.
- Implement a realistic case study that ties together all the concepts of ADO.NET in a practical demonstration

**Course Duration:** 3 days

**Prerequisites:** A basic knowledge of SQL and of programming the .NET Framework using Visual Basic. The student should also understand the fundamentals of XML. To get full benefit from the examples in the course the student should be able to write simple ASP.NET Web Forms applications. A working knowledge of SQL Server is also desirable.



## ADO.NET for Web Applications using VB.NET (BDT4221) 3 days

### Course Outline

1. 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
  
2. ADO.NET Connections
  - A. .NET Data Providers
  - B. IDbConnection
  - C. Connection Classes
  - D. Connection Strings
  - E. Connection Pooling
  - F. Connection Events
  - G. Error Handling
  
3. 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
  
4. 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



## ADO.NET for Web Applications using VB.NET (BDT4221) 3 days

5. 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
  
6. 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
  
7. 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
  
8. Concurrency and Transactions
  - A. DataSets and Concurrency
  - B. Optimistic Concurrency
  - C. Pessimistic Concurrency
  - D. Handling Concurrency Violations
  - E. ADO.NET Transactions
  - F. Database Transactions



ADO.NET for Web Applications using VB.NET (BDT4221)  
3 days

9. Newer Features in ADO.NET and ASP.NET
  - A. Using Server Explorer in Visual Studio 2008
  - B. New ASP.NET Data Access Controls
  - C. Asynchronous Database Operations
  - D. Multiple Active Result Sets
  - E. Bulk Copy in ADO.NET
  
10. LINQ to ADO.NET
  - 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

io/a0