

Microsoft .NET is an advance in programming technology that greatly simplifies application development, both for traditional and proprietary applications. The .NET Framework eases the ability for companies to integrate their systems for Web Service deployment. Microsoft has provided tools for creating servers and applications that are flexible, while giving the end-user a faster and more complete interface. Part of .NET is a major new Object-Oriented programming language: C#.

This four-day course is designed to provide a sound introduction to the .NET Framework for programmers who already know the C# language and the fundamentals of Windows Forms. It is current to .NET 3.5 and Visual Studio 2008. The course focuses on core portions of the .NET Framework that are common across many application areas. Separate courses are available in specific areas, such as ADO.NET, XML Programming, Windows Forms, Windows Presentation Framework, Windows Communications Framework, ASP.NET and Web services. The course is practical, with many examples and a case study. The goal is to equip you to begin building significant applications using the .NET Framework.

Course Objectives:

- Gain a thorough understanding of the philosophy and architecture of .NET.
- Acquire a working knowledge of the .NET programming model and .NET Security.
- Learn how to implement database applications using ADO.NET and LINQ.

Audience: Experienced application developers or architects.

Prerequisites: A working knowledge of C#, including building simple GUIs with Windows Forms.

Number of Days: 4 days

- | | |
|---|--|
| <p>1. .NET Fundamentals</p> <ul style="list-style-type: none"> What Is Microsoft .NET? Open Standards and Interoperability Windows Development Problems Common Language Runtime Attribute-Based Programming Metadata Types NET Framework Class Library Interface-Based Programming Everything is an Object Common Type System ILDASM .NET Framework SDK Tools Language Interoperability Managed Code Assemblies Assembly Deployment | <ul style="list-style-type: none"> JIT Compilation ASP.NET and Web Services The Role of XML Performance <p>2. Class Libraries</p> <ul style="list-style-type: none"> Objects and Components Limitation of COM Components Components in .NET Class Libraries at the Command Line Monolithic versus Component Class Libraries Using Visual Studio References in Visual Studio References at Compile Time and Run Time Project Dependencies Project References |
|---|--|

- 3. **Assemblies, Deployment and Configuration**
 - Assemblies
 - Customer Management System
 - ILDASM
 - Assembly Manifest
 - Assembly Dependency Metadata
 - Assembly Metadata
 - Versioning an Assembly
 - AssemblyVersion Attribute
 - Strong Names
 - Digital Signatures
 - Verification with Digital Signatures
 - Hash Codes
 - Digitally Signing an Assembly
 - Digital Signing Flowchart
 - Signing the Customer Assembly
 - Signed Assembly Metadata
 - Private Assembly Deployment
 - Assembly Cache
 - Deploying a Shared Assembly
 - .NET Framework Configuration Tool
 - Versioning Shared Components
 - How the CLR Locates Assemblies
 - Resolving an Assembly Reference
 - Version Policy in a Configuration File
 - Configuration Tool and Config Files
 - Finding the Assembly
 - Application Settings
 - Application Settings Using Visual Studio
 - Application Configuration File
 - User Configuration File
- 4. **Metadata and Reflection**
 - Metadata
 - Reflection
 - System.Reflection.Assembly
 - System.Type
 - System.Reflection.MethodInfo
 - Dynamic Invocation
 - Late Binding
- 5. **I/O and Serialization**
 - Input and Output in .NET
 - Directories
 - Files and Streams
- 6. **.NET Programming Model**
 - “Read” Command
 - Code for “Write” Command
 - Serialization
 - Attributes
 - Garbage Collection
 - Finalize Method
 - C# Destructor Notation
 - Dispose
 - Finalize/Dispose Test Program
 - Garbage Collection Performance
 - Generations
 - Processes
 - Threads
 - .NET Threading Model
 - Race Conditions
 - Thread Synchronization
 - Monitor
 - Synchronization of Collections
 - Asynchronous Calls
 - Asynchronous Delegates
 - Using a Callback Method
 - .NET 2.0 BackgroundWorker
 - Application Isolation
 - Application Domain
 - Application Domains and Assemblies
 - AppDomain
 - CreateDomain
 - App Domain Events
 - Distributed Programming in .NET
 - Windows Communication Foundation
 - .NET Remoting Architecture
 - Remote Objects and Mobile Objects
 - Object Activation and Lifetime
 - Singleton and SingleCall
- 7. **.NET Security**
 - Fundamental Problem of Security
 - Authorization
 - Authentication
 - The Internet and .NET Security
 - Code Access Security

- Role-Based Security
- .NET Security Concepts
- Configuring Security in .NET
- Caspol.exe
- Policy Levels
- Code Groups
- Standard Code Groups
- Other Assemblies
- Programming Code Access Security
- Evidence-Based Security
- Host and Assembly Evidence
- Enumerating Evidence
- Imperative CAS Using Evidence
- EvidenceCASClient Program
- DoSomethingForClient Method
- Running over the Intranet
- Configuring LocalIntranet_Zone
- Permissions
- IPermission Interface
- IPermission Demand Method
- IPermission Inheritance Hierarchy
- Stack Walking
- Assert
- Demand
- Deny
- PermitOnly and Revert Methods
- Other CAS Methods
- ZoneIdentityPermission Class
- Declarative CAS
- Role-Based Security in .NET
- Identity Objects
- Principal Objects
- Windows Principal Information
- Custom Identity and Principal
- BasicIdentity.cs
- BasicSecurity.cs
- Users.cs
- Roles.cs
- RoleDemo.cs
- PrincipalPermission

8. Interoperating with COM and Win32

- Interoperating Between Managed and Unmanaged Code
- COM Interop and PInvoke
- Calling COM Components from Managed Code

- The TlbImp.exe Utility
- TlbImp Syntax
- Using TlbImp
- Register the COM Server
- OLE/COM Object Viewer
- Run the COM Client
- Implement the .NET Client Program
- Import a Type Library Using Visual Studio
- Platform Invocation Services (Pinvoke)
- Marshalling out Parameters
- Translating Types
- 9. ADO.NET and LINQ**
- ADO.NET
- ADO.NET Architecture
- .NET Data Providers
- ADO.NET Interfaces
- .NET Namespaces
- Connected Data Access
- Creating a Connection
- Using Server Explorer
- Performing Queries
- Connecting to a Database
- Database Code
- Using Commands
- Creating a Command Object
- Using a Data Reader
- Generic Collections
- Executing Commands
- Parameterized Queries
- DataSet
- DataSet Architecture
- Why DataSet?
- DataSet Components
- DataAdapter
- Data Access Class
- Retrieving the Data
- Filling a DataSet
- Accessing a DataSet
- Using a Standalone Data Table
- Adding a New Row
- Searching and Updating a Row
- Deleting a Row
- Row Versions

Row State
 Iterating Through DataRows
 Command Builders
 Updating a Database
 Data Binding
 DataGridView Control
 Language Integrated Query (LINQ)
 Bridging Objects and Data
 Object Relational Designer
 IntelliSense
 Basic LINQ Query Operators
 Obtaining a Data Source
 Filtering
 Ordering
 Aggregation
 Obtaining Lists and Arrays
 Deferred Execution
 Modifying a Data Source
 Performing Inserts via LINQ to SQL
 Performing Deletes via LINQ to SQL
 Performing Updates via LINQ to SQL

10. **Debugging Fundamentals**

Compile-Time Errors
 Runtime Errors
 Debugging
 Project Configurations
 Release Configuration
 Creating a New Configuration
 Build Settings for a Configuration
 Customizing a Toolbar
 Using the Visual Studio Debugger
 Overflow Exception
 Just-in-Time Debugging
 Standard Debugging – Breakpoints
 Standard Debugging – Watch Variables
 Stepping with the Debugger
 The Call Stack
 JIT Debugging in Windows Apps
 Configuration File

11. **Tracing**

Instrumenting an Application
 Order Application
 Debugging Review
 Tracing
 Debug and Trace Classes
 Viewing Trace Output

Debug Statements
 Debug Output
 Assert
 More Debug Output
 WriteLine Syntax
 Event Logs
 Viewing Event Logs
 Event Log Entry Types
 .NET EventLog Component
 Retrieving Entries from an Event Log
 Handling EventLog Events

12. **More about Tracing**

Trace Switches
 BooleanSwitch
 Using a Configuration File
 TraceSwitch
 SwitchDemo
 Trace Listeners
 DefaultTraceListener
 A Stream Listener
 A Custom Listener
 Trace Output to a Window
 An Event Log Listener
 Tracing in the Order Application