

This three-day course covers the essentials of the .NET Framework for programmers who already know the Visual Basic language and the fundamentals of Windows Forms. It is current to .NET 4.5 and Visual Studio 2012. The course focuses on core portions of the .NET Framework that are common across many application areas. The course starts with an introduction to the architecture and key concepts of .NET. It then discusses class libraries, assemblies, versioning, configuration, and deployment, which constitute a major advance in the simplicity and robustness of deploying Windows applications, ending the notorious “DLL hell. 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.

**Audience:** Experienced application developers or architects.

**Prerequisites:** The student should be an experienced application developer or architect with a working knowledge of Visual Basic, including building simple GUIs with Windows Forms.

**Number of Days:** 3 days

## 1. .NET Fundamentals

- 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
- JIT Compilation
- ASP.NET and Web Services
- The Role of XML
- Performance

## 2. Class Libraries

- 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
- Specifying Version Numbers

## 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
- Versioning Shared Components

- How the CLR Locates Assemblies
- Resolving an Assembly Reference
- Version Policy in a Configuration File
- Finding the Assembly
- Application Settings
- Application Settings Using Visual Studio
- My.Settings
- Application Configuration File
- User Configuration File
- 4. Metadata and Reflection**
- Metadata
- Reflection
- Sample Reflection Program
- System Reflection Assembly
- System.Type
- System.Reflection.MethodInfo
- Dynamic Invocation
- Late Binding
- 5. I/O and Serialization**
- Input and Output in .NET
- Directories
- Directory Example Program
- Files and Streams
- “Read” Command
- Code for “Write” Command
- Serialization
- Attributes
- 6. .NET Programming Model**
- Garbage Collection
- Finalize Method
- Dispose
- Finalize/Dispose Test Program
- Garbage Collection Performance
- Generations
- Processes
- Threads
- Asynchronous Calls
- Asynchronous Delegates
- Using a Callback Method
- BackgroundWorker
- Asynchronous Programs in VB 2012
- Task and Task(Of TResult)
- Async Methods
- Synchronous Call
- Async Call
- Threading
- Application Isolation
- Application Domain
- Application Domains and Assemblies
- AppDomain
- CreateDomain
- App Domain Events
- 7. .NET Threading**
- Threads
- Race Conditions
- Thread Synchronizations
- Monitor
- Using VB SyncLock Keyword
- Synchronization of Collections
- ThreadPool Class
- Starting a ThreadPool Thread
- Foreground and Background Threads
- Synchronizing Threads
- Task Parallel Library (TPL)
- Starting Tasks
- Waiting for Task Completion
- Data Parallelism
- 8. .NET Security**
- Fundamental Problem of Security
- Authentication
- Authorization
- The Internet and .NET Security
- Code Access Security
- .NET Security Concepts
- Permissions
- IPermission Interface
- IPermission Demand Method
- IPermission Inheritance Hierarchy
- Stack Walking
- Assert
- Demand
- Other CAS Methods
- Security Policy Simplification
- Simple Sandboxing API
- Setting up Permissions
- Creating the Sandbox
- Role-Based Security in .NET
- Identity Objects
- Principal Objects
- Windows Principal Information
- 9. 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
- 64-bit System Considerations
- Run the COM Client
- Implement the .NET Client Program
- The Client Target Platform Is 32-bit
- Import a Type Library Using Visual Studio
- Platform Invocation Services (Pinvoke)
- Marshalling *ByRef* Parameters
- Translating Types

## 10. **ADO.NET and LINQ**

- ADO.NET
- ADO.NET Architecture
- .NET Data Providers
- ADO.NET Interfaces
- .NET Namespaces
- Connected Data Access
- AcmePub Database
- Creating a Connection
- Using Server Explorer
- Performing Queries
- Connecting to a Database
- Database Code
- Connection String
- 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 DataTable
- 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

## **Appendix A: .NET Remoting**

- Distributed Programming in .NET
- Windows Communication Foundation
- .NET Remoting Architecture
- Remote Objects and Mobile Objects
- Object Activation and Lifetime
- Singleton and SingleCall

## **Appendix B: Learning Resources**