



## Course Outline

### Developing and Deploying Secure Microsoft .NET Framework Applications

Course 2350—Three days—Instructor-Led

#### Introduction

This three-day instructor-led course teaches developers to develop and deploy secure Microsoft .NET client applications by using Microsoft Visual Studio .NET and the Microsoft .NET Framework. The course provides an overview of security; discusses design issues, including threat modeling techniques and coding techniques that enhance security; and explains why type-safety verification is the cornerstone of Microsoft .NET Framework security. This course provides side-by-side coding examples and activities written in both Microsoft Visual Basic .NET and Microsoft Visual C#.

#### Audience

This course is intended for experienced professional software developers who work on development teams in a corporate enterprise or for independent software vendors. These developers may also develop enterprise level applications in a networked environment.

#### At Course Completion

After completing this course, students will be able to:

- Understand basic security and deployment problems that the .NET Framework can help prevent.
- Use the MSIL Disassembler to view assembly and type metadata.
- Use reflection to programmatically access assembly and type metadata.
- Use the STRIDE threat model to develop a threat mitigation strategy for an application.
- Encrypt and decrypt data by using symmetric and asymmetric encryption.
- Use permission requests to specify and to limit those permissions that are granted to code.
- Create a single-file assembly or a multifile assembly, depending on the requirements of the software development project.
- Create a private assembly or a shared assembly, depending on the requirements of the software development project.
- Create and access files that are written to isolated storage.
- Create a setup project that specifies where to install the files, what conditions must be met before installing certain files, and after the installation, what custom actions to perform.
- Describe the process that the runtime uses to ensure that it finds and binds to the correct version of an assembly.
- Create configuration files to modify the assembly version and the locations that the runtime searches to find assemblies.

#### Prerequisites

Developers who will gain the most from this course have a working understanding of the .NET Framework and some project experience writing .NET Framework client applications by using either Visual Basic .NET or Visual C#. Before attending this course, students must have:

- Experience developing applications by using the .NET Framework, including using:
  - The common language runtime
  - Managed execution
  - Basic file I/O classes, such as the FileStream class

Students can satisfy the prerequisites for this course by attending Course 2349: Programming with the Microsoft .NET Framework (Microsoft Visual C# .NET), or Course 2415: Programming with the Microsoft .NET Framework (Microsoft Visual Basic .NET).

- Programming experience with Visual Basic .NET or Visual C#, including:





## Course Outline

- Opening, building, running, and debugging projects in Visual Studio .NET
- Declaring variables and using loops and conditional statements
- Creating classes and methods
- Using attributes

Students can satisfy the prerequisites for Visual Basic .NET by completing Course 2559: Introduction to Visual Basic .NET Programming with Microsoft .NET, or Course 2373: Programming with Microsoft Visual Basic .NET.

Students can satisfy the prerequisites for Visual C# by completing Course 2124: Programming with C#, or Course 2609: Introduction to C# Programming with Microsoft .NET.

The following books and courses provide a depth of knowledge of the .NET Framework that is useful for this course:

- Programming Microsoft .NET by Jeff Prosise (Wintellect)
- Applied Microsoft .NET Framework Programming in Microsoft Visual Basic .NET by Jeffrey Richter & Francesco Balena (Wintellect)
- Course 2349: Programming with the Microsoft .NET Framework (Microsoft Visual C# .NET)
- Course 2415: Programming with the Microsoft .NET Framework (Microsoft Visual Basic .NET)

### Microsoft Certification exams

This is no Microsoft Certified Professional exam associated with this course.

### Course Materials

The student kit includes a comprehensive workbook and other necessary materials for this class.

### Course Outline

#### ***Module 1: Introduction to .NET Framework Security and Deployment***

This module introduces concepts and terminology, including a working definition of assembly, that are related to security and deployment in the Microsoft .NET Framework.

#### **Lessons**

- Introduction to .NET Assemblies
- Overview of Security Measures
- Overview of Deployment Concepts

There is no lab for this module

After completing this module, students will be able to:

- Define the term .NET Framework assembly.
- List the possible contents of a .NET Framework assembly.
- Describe common security and deployment problems that can be addressed by using .NET Framework assemblies.
- Describe the security architecture of the .NET Framework.
- Define terms and concepts used to describe deployment of applications that are built with the .NET Framework.
- Understand basic security and deployment problems that the .NET Framework can help prevent.



## Course Outline

### ***Module 2: Viewing Metadata and Using Reflection***

This module discusses metadata as it applies to assemblies and types. Reading metadata in Microsoft intermediate language (MSIL) code enables you to understand and troubleshoot assembly and type references. This module also discusses techniques for programmatically accessing metadata by using reflection.

#### **Lessons**

- Viewing Metadata
- Using Reflection

There is no lab for this module

After completing this module, students will be able to:

- Define assembly metadata.
- Define type metadata.
- Describe how the compiler uses assembly and type metadata to resolve references.
- Use the MSIL Disassembler to view assembly and type metadata.
- Use reflection to programmatically access assembly and type metadata.

### ***Module 3: Secure Coding and Type-Safety Verification***

This module provides an overview of security, discusses some design and coding techniques that enhance security, and then explains why type-safety verification is the cornerstone of Microsoft .NET Framework security.

#### **Lessons**

- Security Basics
- Creating and Using a Threat Model
- Type-Safety Verification

There is no lab for this module

After completing this module, students will be able to:

- Describe the concept of security as it applies to traditional security measures and to applications written by using the .NET Framework.
- Use the STRIDE threat model to develop a threat mitigation strategy for an application.
- Describe how type-safety verification forms the basis of .NET Framework application security.

### ***Module 4: Cryptography and Digital Signing***

This module discusses cryptography and digital signing. These technologies involve the protection of data and code. You can encrypt data to prevent unauthorized users from viewing it, and you can sign both data and code to prevent tampering and to identify the sender. The Microsoft .NET Framework provides extensive support for cryptography and data signing.



## Course Outline

### Lessons

- Cryptography and Signing Basics
- Encrypting and Decrypting Data with a Symmetric Algorithm
- Encrypting, Decrypting, and Signing Data with an Asymmetric Algorithm
- Signing Code

### Lab: Encrypting and Decrypting Text with a Password

This lab covers how to perform the following tasks:

- Generate a key for a symmetric algorithm from a password and a random number.
- Encrypt data by using a symmetric algorithm.
- Decrypt data by using a symmetric algorithm.

After completing this module, students will be able to:

- Describe symmetric and asymmetric encryption, hashing, and digital signing.
- Encrypt and decrypt data by using a password and symmetric encryption.
- Encrypt, decrypt, and sign data by using asymmetric encryption.
- Hash data.
- Sign and delay-sign an assembly with a strong name.

### Module 5: Code Access Security

This module discusses code access security. This feature of the .NET Framework allows the developer and the systems administrator to exercise precise control over the resources that code is given permission to access. You can use tools and classes that are provided with the Microsoft .NET Framework to view and modify how code access security is implemented in your application.

### Lessons

- Overview of Code Access Security
- Modifying Security Policy
- Security Operations Basics
- Performing Imperative Security Operations
- Performing Declarative Security Operations
- Adding Permission Requests

### Lab: Using Code Access Security

This lab covers how to perform the following tasks:

- Perform demand and assert operations by using imperative code access security.
- Add minimum and optional permission requests to an assembly.

After completing this module, students will be able to:

- Describe how the .NET Framework security system uses code access security to control the amount of permission to access computing resources that is granted to code.
- Modify security policy that is applied to assemblies.
- Use code to assert and to demand permissions imperatively.
- Use attributes to assert and to demand permissions declaratively.
- Use permission requests to specify and to limit those permissions that are granted to code.

## Course Outline

### ***Module 6: Role-Based Security***

This module discusses programming techniques for implementing role-based security by using the Microsoft .NET Framework.

#### **Lessons**

- Role-Based Security Basics
- Role-Based Security with Principal and Identity Objects
- Role-Based Security with Permission Objects

#### **Lab: Role-Based Security**

This lab covers how to perform the following tasks:

- Perform a role-based security check by using a principal object
- Perform a role-based security check by using a permission object
- Perform a role-based security check by using a permission attribute

After completing this module, students will be able to:

- Describe how role-based security is implemented by the .NET Framework.
- Perform role-based security checks with principals and identities.
- Perform role-based security checks with permission objects.

### ***Module 7: Isolated Storage***

This module discusses isolated storage, what it is, the advantages of using it, and how to use it.

#### **Lessons**

- Isolated Storage Basics
- Using Isolated Storage

There is no lab for this module

After completing this module, students will be able to:

- Describe the types of isolated storage.
- Describe the scenarios for using isolated storage.
- List the security permissions that are required for using isolated storage.
- Open a store.
- Create, read, and write files and folders in a store.

### ***Module 8: Creating an Assembly***

This module describes why and how to deploy an assembly either as a single file or as multiple files. It then describes why and how to deploy an assembly privately or as a shared assembly.



## Course Outline

### Lessons

- Creating Single-File and Multifile Assemblies
- Creating Privately Deployed and Shared Assemblies

There is no lab for this module

After completing this module, students will be able to:

- Create a single-file assembly or a multifile assembly, depending on the requirements of the software development project.
- Create a private assembly or a shared assembly, depending on the requirements of the software development project.

### ***Module 9: Deploying .NET Framework Applications***

This module discusses specific reasons for using each deployment option. It also describes how to create deployment projects and how to customize deployment.

### Lessons

- Overview of Deployment
- Creating a Setup Project

There is no lab for this module

After completing this module, students will be able to:

- Choose whether to use the XCOPY command, Microsoft Windows( Installer, or a Cab project to deploy an assembly.
- Create a setup project that specifies where to install the files, what conditions must be met before installing certain files, and what custom actions to perform after installation is complete.

### ***Module 10: Assembly Binding and Configuration***

This module covers how to configure assembly binding by using the Microsoft .NET Framework. The ability to manage assembly binding allows you to perform the following deployment tasks:

- Deployment of an updated shared component across an enterprise.
- Allow a specific application to continue to use an earlier version of a shared assembly.
- Enforce binding policy across the enterprise without exception.

### Lessons

- Versioning and Assembly Binding Basics
- Configuration File Syntax
- Creating Policy Configuration Files

There is no lab for this module

After completing this module, students will be able to:



## Course Outline

- Describe the process that the runtime uses to ensure that it finds and binds to the correct version of an assembly.
- Identify the XML element in a configuration file that modifies binding instructions.
- Create configuration files to modify the assembly version and the locations that the runtime searches to find assemblies.