

Developing Microsoft .NET Applications for Windows (Visual Basic .NET)

Course 2565—Five days—Instructor-led

Introduction

This five-day instructor-led course provides students with the skills required to build Microsoft Windows Forms applications by using the Microsoft .NET Framework. This course is a part of the Microsoft Visual Basic .NET curriculum and is intended to provide Visual Basic programmers with the skills required to create Windows Forms applications by using the .NET Framework. The course will cover the major topics for Windows client application programming on the .NET Framework. These topics include: Windows Forms, GDI+, simple data access, interoperating with unmanaged code, threading and asynchronous programming issues, simple remoting, Web access, XML Web services consumption, debugging, security, and deployment issues for desktop applications.

Audience

This course is intended for the intermediate programmer who is responsible for designing and building Windows-based applications with the .NET Framework. It is designed for developers who have Visual Basic development skills. Students are required to have the following skills:

- Experience with a .NET language such as Visual Basic .NET
- Experience developing basic applications with MFC and/or Microsoft Visual Basic 6.0

Typically, these individuals perform the following key activities:

- Help create functional specifications.
- Design and develop user interfaces.
- Create and test prototypes.
- Write Windows Forms applications.

At Course Completion

After completing this course, students will be able to:

- Create and populate Windows Forms.
- Organize controls on Windows Forms.
- Create menus in a Windows Forms application.
- Add code to form and control event procedures in a Windows Forms application.
- Create Multiple Document Interface (MDI) applications.
- Use dialog boxes in Windows Forms applications.
- Validate user input in a Windows Forms application.
- Create and use user controls in a Windows Forms application.
- Create licenses for controls.
- Bind Windows Forms applications to various data sources by using Microsoft ADO.NET.
- Consume XML Web services from Windows Forms applications.
- Use .NET and COM components in a Windows Forms application.
- Call Microsoft Win32 APIs from a Windows Forms application.
- Migrate Visual Basic 6.0 applications to Visual Basic .NET.
- Print documents in a Windows Forms application.
- Make asynchronous calls to methods from a Windows Forms application.
- Debug a Windows Forms application.
- Incorporate accessibility features in a Windows Forms application.
- Localize a Windows Forms application.
- Add support for Help to localize a Windows Forms application.
- Create Help files in a Windows Forms application.
- Deploy a Windows Forms application.
- Implement code access and role-based security in a Windows Forms application.
- Add deployment flexibility to applications by using shared assemblies.

Prerequisites

The course assumes that students have the following skills:

- Understanding of Visual Basic programming
- Experience building applications

The following prerequisites are recommended for this course:

- Course 2559A—Introduction to Visual Basic .NET programming
Or
Equivalent knowledge

Microsoft Certification exams

This course helps you prepare for the following Microsoft Certified Professional exam:

- Exam 70-306: Developing and Implementing Windows-based Applications with Microsoft Visual Basic .NET and Microsoft Visual Studio .NET

Student Materials

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

Course Outline

Module 1: Introducing Windows Forms

This module introduces Windows forms and controls in the .NET Framework. Students will learn how to create and populate base forms and inherited forms by using the Microsoft Visual Studio .NET Framework. They will also learn how to add controls to a form and how to create Multiple Document Interface (MDI) applications.

Lessons

- Creating a Form
- Adding Controls to a Form
- Creating an Inherited Form
- Organizing Controls on a Form
- Creating MDI Applications

Lab 1.1: Creating Windows Forms

- Creating a New Windows Form
- Inheriting a New Form from an Existing Windows Form

After completing this module, students will be able to:

- Create a form and add controls to it.
- Create an inherited form by using Visual Inheritance.
- Organize controls on a form.
- Create Multiple Document Interface (MDI) applications.

Module 2: Working With Controls

This module explains how to create event procedures (handlers) in your application that will run in response to user actions. Students will learn how to add programming logic to the event procedures of a control, how to use the Windows Forms intrinsic controls, dialog boxes, and menus, and how to validate the data entered by users of your application.

Lessons

- Creating an Event Handler for a Control
- Using Windows Forms Controls
- Using Dialog Boxes in a Windows Forms Application
- Adding Controls at Run Time
- Creating Menus
- Validating User Input

Lab 2.1: Working with Controls

- Creating and Using Controls

After completing this module, students will be able to:

- Create an event handler for a control.
- Select and use the appropriate controls in a Windows Forms application.
- Use dialog boxes in a Windows Forms application.
- Add controls to a form at run time.
- Create and use menus in a Windows Forms application.
- Validate user input in a Windows Forms application.

Module 3: Building Controls

This module explains how to extend the functionality of an existing Windows control, combine multiple existing controls into a composite control, and build a new custom control. Students will also learn how to add design-time licensing support to a control.

Lessons

- Extending and Creating Controls
- Adding Design-Time Support for Controls
- Licensing a Control

Lab 3.1: Building Controls

- Declare an Event and Raising It from an Extended Control
- Creating a Composite Control
- Adding Design-Time Support

After completing this module, students will be able to:

- Extend an existing control.
- Create a composite control by combining functionality of several existing Windows Forms controls.
- Describe the design-time support options for components provided by Visual Studio .NET.
- Add attributes that provide information to the Visual Designer.
- Create and validate licenses for controls.

Module 4: Using Data in Windows Forms Applications

This module explains how to bind Windows forms to various data sources by using ADO .NET. Students will learn about binding Windows forms to data from sources such as databases and XML files. Students will get an overview of the XML Web services programming model and learn how to create applications that use XML Web services. The module also provides an overview of how to persist data to and read data from files and isolated storage.

Lessons

- Adding ADO.NET Objects to and Configuring ADO.NET Objects in a Windows Forms Application
- Accessing and Modifying Data by Using DataSets
- Binding Data to Controls
- Overview of XML Web Services
- Creating a Simple XML Web Services Client
- Persisting Data

Lab 4.1: Accessing Data with ADO.NET

- Generating and Populating DataSets
- Modifying a DataSet
- Updating a DataSet to a DataSource
- Binding and Formatting Data in Controls

Lab 4.2: Calling an XML Web Service

- Calling an XML Web Service

After completing this module, students will be able to:

- Describe the objects in the ADO.NET object model.
- Add and configure ADO.NET objects in a Windows Forms application.
- Access and modify data from a database by using DataSets.
- Bind data to controls.
- Describe the XML Web services model and the roles of HTML, SOAP, and XML in the XML Web services model.
- Create and test a simple XML Web service client application.
- Persist data to files, serialize objects, use isolated storage, and persist application settings.

Module 5: Interoperating with Managed Objects

This module explains how to use .NET and COM components in a Windows Forms application. Students will also learn how to call Win32 APIs in their Windows Forms application.

Lessons

- Using .NET and COM Components in a Windows Forms Application
- Calling Win32 APIs from Windows Forms Applications
- Upgrading Visual Basic 6.0 Applications to Visual Basic .NET

Lab 5.1: Interoperating with COM and Calling Win32 APIs

- Using a COM Component in a .NET-Based Application
- Calling Win32 APIs from a .NET-Based Application

After completing this module, students will be able to:

- Use .NET and COM components in a Microsoft .NET Framework Windows Forms application.
- Call Win32 APIs from a Windows Forms application.
- Upgrade Visual Basic 6.0 applications to Visual Basic .NET.

Module 6: Printing and Reporting in Windows Forms Applications

This module explores how to implement printing in a Windows Forms application and how to create reports in a Windows Forms application by using Crystal Reports for Visual Studio .NET.

Lessons

- Printing From a Windows Forms Application
- Using the Print Preview, Page Setup, and Print Dialogs
- Constructing Print Document Content by Using GDI+
- Creating Reports by Using Crystal Reports

Lab 6.1: Printing Formatted Documents

- Adding Print Support to an Application
- Creating Printed Output by Using GDI+

After completing this module, students will be able to:

- Print documents in a Windows Forms application.
- Use the Visual Studio .NET printing dialog boxes in a Windows Forms application.
- Use GDI+ to construct print document content.
- Create and format reports by using Crystal Reports.

Module 7: Asynchronous Programming

This module explains how to use the techniques of asynchronous programming and multithreading to avoid blocking the user interface of an application.

Lessons

- The .NET Asynchronous Programming Model
- The Asynchronous Programming Model Design Pattern
- How to Make Asynchronous Calls to Any Method
- Helping to Protect State and Data in a Multithreaded Environment

Lab 7.1: Making Asynchronous Calls to an XML Web Service

- Converting Synchronous Calls to Asynchronous Calls

After completing this module, students will be able to:

- Describe the .NET Framework asynchronous programming model.
- Modify a client application to use built-in .NET Framework support for asynchronous calls to methods.
- Describe how to add explicit support for asynchronous calls to any method.

Module 8: Enhancing the Usability of Applications

This module covers how to use the accessibility, Help, and localization features available in the .NET Framework.

Lessons

- Adding Accessibility Features
- Adding Help to an Application
- Localizing an Application

Lab 8.1: Enhancing the Usability of an Application

- Adding Support for Accessibility
- Adding Help to an Application
- Adding ToolTips to an Application
- Localizing the User Interface of an Application
- Localizing Resources in an Application

After completing this module, students will be able to:

- Use .NET Framework features to add and enable accessibility features in an application.
- Add support for context-sensitive Help, Help menus, and ToolTips to an application.
- Use localization properties and resource files to create a localized version of a .NET Framework Windows Forms application.

Module 9: Deploying Windows Forms Applications

This module explains assemblies and how to use strong-named assemblies and the Global Assembly Cache (GAC) in the .NET Framework. Students will also learn how to configure and deploy .NET-based applications.

Lessons

- .NET Assemblies
- Deploying Windows Forms Applications

Lab 9.1: Deploying an Application

- Building and Referencing a Strong-Named Assembly
- Installing a Strong-Named Assembly into the Global Assembly Cache
- Deploying a .NET Application
- Using an Application Configuration File

After completing this module, students will be able to:

- Use strong-named assemblies in .NET applications.
- Use application configuration files to configure and use Microsoft Windows Installer 2.0 to package and deploy .NET applications.



Module 10: Securing Windows Forms Applications

This module explains the .NET Framework security model. Students will learn how to use .NET Framework security features in Windows Forms applications.

Lessons

- Security in the .NET Framework
- Using Code Access Security
- Using Role-Based Security

Lab 10.1: Adding and Testing Permission Requests

- Adding and Testing Permission Requests

After completing this module, students will be able to:

- Describe the .NET Framework security model.
- Use code access security to help protect an application.
- Use role-based security to help control access to an application.