Testsworld.com 

Microsoft Certification

MCAD Certification

The Microsoft Certified Application Developer (MCAD) credential provides industry recognition for professional developers who build powerful applications using Microsoft Visual Studio® .NET and Web services. Do something new for your career today.

Benefits of MCAD Certification

Get industry recognition for your ability to build Microsoft Visual Studio .Net and Web Services applications. This certifications demonstrate that you have the skills to build, deploy, and maintain Microsoft applications

MCAD Certification Track Details

• Core Exams (2 Exams Required)

• Elective Exams (1 Exams Required

1 Core Exams Application Development

Exam 70—305: Developing and Implementing Web Applications with Microsoft Visual Basic® .NET and Microsoft Visual Studio® .NET

OR

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

OR

Developing Exam 70—3 15:and Implementing Web Applications with Microsoft Visual C#TM .NET and Microsoft Visual Studio .NET

OR

Exam 70—316: Developing and Implementing Windows-based Applications with Microsoft Visual C# .NET and Microsoft Visual Studio .NET

1 Core Exams Server Component Development

Exam 70—3 10: Developing XML Web Services and Server Components with Microsoft

Visual Basic .NET and the Microsoft .NET Framework

OR

Exam 70—320: Developing XML Web Services and Server Components with Microsoft Visual C# and the Microsoft .NET Framework

Elective Exams (1 Exam Required)

Exam 70—229: Designing and Implementing Databases with Microsoft SQL

ServerTM 2000 Enterprise Edition
OR
Exam 70—230: Designing and Implementing Solutions with Microsoft BizTalk Server® 2000 Enterprise Edition
OR
Exam 70—234: Designing and Implementing Solutions with Microsoft Commerce Server 2000

Preparation Guide for Exam 70-305

Audience Profile

Candidates for this exam work on **a** team in **a** medium or large development environment that uses Microsoft Visual Studio .NET, Enterprise Developer Edition. Candidates should have at least one year of experience developing Web-based applications. Candidates should have a working knowledge of Microsoft Visual Basic .NET.

To top of page

Credit Toward Certification

When you pass the Developing and Implementing Web Applications with Microsoft
Visual Basic .NET and Microsoft Visual Studio .NET exam, you achieve Microsoft
Certified Professional status. You also earn:

- Core credit toward Microsoft Certified Solution Developer (MCSD) for Microsoft
.NET certification.
- Core or elective credit toward Microsoft Certified Application Developer (MCAD)
for Microsoft .NET certification.

- Elective credit toward Microsoft Certified Database Administrator (MCDBA) on Microsoft SQL Server 2000 certification.

Preparation Tools and Resources

We make **a** wealth of preparation tools and resources available to you, including courses, books, practice tests, and Microsoft Web sites. When you are ready to prepare for this exam, here's where you should start.

Instructor-led Courses for This Exam

- Course 2310: Developing Microsoft ASP.NET Web Applications Using Microsoft Visual Studio .NET
- Course 2389: Programming with Microsoft ADO.NET
- Course 2640: Upgrading Web Development Skills from ASP to Microsoft ASP.NET

E-Learning Courses for This Exam

- Course 2310: Developing Microsoft ASP.NET Web Applications Using Microsoft Visual Studio .NET
- Course 2389: Programming with Microsoft ADO.NET

* Course 2640: Upgrading Web Development Skills from ASP to Microsoft ASP.NET
This certification exam measures your ability to develop and implement Web-based applications with Web forms, ASP.NET, and the Microsoft .NET Framework. Before taking the exam, you should be proficient in the job skills listed in the following matrix. The matrix shows which Official Microsoft Learning Products may help you reach competency in the skills being tested in the exam.
The course includes material to prepare you for this task.
The course includes some material to prepare you for this task. You will need to supplement the course with additional work.
The course provides a general introductory overview of this task. You will need to supplement the course with additional work.

Preparation Guide for Exam 70-306

Audience Profile

Candidates for this exam work on a team in a medium or large development environment that uses Microsoft Visual Studio .NET, Enterprise Developer Edition. Candidates have at least one year of experience developing Windows-based applications. Candidates should have a working knowledge of Microsoft Visual Basic .NET.

Credit Toward Certification

When you pass the Developing and Implementing Microsoft Windows-based Applications with Microsoft Visual Basic .NET and Microsoft Visual Studio .NET exam, you achieve Microsoft Certified Professional status. You also earn:

* Core credit toward Microsoft Certified Solution Developer (MCSD) for Microsoft .NET certification.
* Core or elective credit toward Microsoft Certified Application Developer (MCAD) for Microsoft .NET certification.
* Elective credit toward Microsoft Certified Database Administrator (MCDBA) on Microsoft SQL Server 2000 certification.

Preparation Tools and Resources

We make a wealth of preparation tools and resources available to you, including courses, books, practice tests, and Microsoft Web sites. When you are ready to prepare for this exam, here's where you should start.
Instructor-led Courses for This Exam

* Course 2389: Programming with Microsoft ADO .NET
* Course *2565:* Developing .NET Windows Applications (Microsoft Visual Basic .NET)

Course 2389: Programmin2 with Microsoft ADO .NET

Introduction

This course will teach developers to build data-centric applications and Web services with Microsoft ADO.NET, Microsoft SQL Server 2000, and the Microsoft .NET Framework

Audience

This course is designed for the professional platform developer who is responsible for designing and building data-centric, distributed applications for his or her organization. It is designed for developers who have component and Web application development skills, and who have previously built solutions by using Microsoft Visual Studio. Typically, these individuals will have the following skills: Experience with a .NET language such as Microsoft Visual Basic .NET, Microsoft Visual C#, or Microsoft Visual C++; an understanding of object-oriented concepts and terminology; experience developing distributed applications; and experience developing Web-based applications hosted on Internet Information Server.

At Course Completion

After completing the course, students will be able to:
Describe data-centric applications, ADO.NET architecture, and ADO.NET and XML.
Connect to SQL Server and other data sources.
Perform connected database operations including executing SELECT commands, database definition commands, dynamic SQL commands, and commands that return data from a SQL Server database in XML.
Build a DataSet schema, populate it with data, and modify the data pro grammatically.
Build a DataSet from an existing data source.

Use XML techniques while working with DataSets, including mapping tables and columns, creating XSD schemas, building strongly typed DataSets, and interacting with XMLDataDocuments.
Build a Web service that uses ADO.NET to query and update a data source. Troubleshoot errors within an ADO.NET application.

Prerequisites

The course assumes that students have the following skills:
Understanding of relational database concepts: table, row, column, primary
keys, foreign keys, constraints, and views

Data query and modification experience, including experience with SELECT,
INSERT, UPDATE, and DELETE commands
Exposure to XML documents, style sheets, and schemas
Experience with Visual Basic .NET, Visual Basic for Applications, or previous versions of Visual Basic
Experience building user interfaces, including Web applications or Microsoft
Windows applications

Course Materials

The course materials are yours to keep. You will be provided with the following software for use in the classroom:
Microsoft Windows 2000 Server
Microsoft SQL Server 2000 Developer Edition
XML for SQL Server 2000 Web Release 2 (WR2)
Microsoft Visual Studio .NET
Microsoft Access 2002

Course Outline

Module 1: Data-Centric Applications and ADO.NET
The following topics are covered in this module:
Design of Data-Centric Applications

ADO .NET Architecture
ADO.NET and XML
After completing this module, students will be able to:
Give examples of storage options.
Diagram the architecture of data-centric applications.
Choose a connected, disconnected, or mixed environment based on application requirements.
Use the System.Data namespaces in applications.
Diagram the ADO.NET object model.
Analyze typical business scenarios.
Explain how to use ADO.NET with XML.
Lab 1.1: Data-Centric Applications and ADO.NET
Module 2: Connecting to Data Sources
The following topics are covered in this module:
Choosing a .NET Data Provider

Defining a Connection
Managing a Connection
Handling Connection Exceptions
Connection Pooling
After completing this module, students will be able to:
Choose a .NET data provider.
Connect to SQL Server.
Connect to OLE DB data sources.
Manage a connection.
Handle common connection exceptions.
Implement and control connection pooling.
Lab 2.1: Connecting to Data Sources
Module 3: Performing Connected Database Operations
The following topics are covered in this module:
Working in a Connected Environment

Building Command Objects
Executing Commands That Return a Single Value
Executing Commands That Return Rows
Executing Commands That Do Not Return Rows
Using Transactions
After completing this module, students will be able to:
Build a command object.
Execute a command that returns a single value.
Execute a command that returns a set of rows, and process the result.
Execute a command that defines database structure and permissions by using
the data definition language (DDL) and data control language (DCL).
Execute a command that modifies data.
Use transactions.
Lab 3.1: Performing Connected Database Operations
Module 4: Building DataSets
The following topics are covered in this module:
Working in a Disconnected Environment
Building DataSets and DataTables
Binding and Saving a DataSet
Defining Data Relationships

Modifying Data in a DataTable
Sorting and Filtering
After completing this module, students will be able to:
Describe the disconnected environment.
Build a DataSet and a DataTable.
Bind a DataSet to a DataGrid.
Open and save a DataSet.
Define a data relationship.
Modify data in a DataTable.
Find and select rows in a DataTable.
Sort and filter a DataTable by using a DataView.
Lab 4.1: Building, Binding, Opening, and Saving DataSets
Lab 4.2: Manipulating DataSets

Module *5:* Reading and Writing XML with ADO.NET
The following topics are covered in this module:
Creating XSD Schemas
Loading Schemas and Data into DataSets
Writing XML from a DataSet
After completing this module, students will be able to:
Generate an XSD schema from a DataSet by using graphical tools.
Save a DataSet structure to an XSD schema file.
Create and populate a DataSet from an XSD schema and XML data.
Save DataSet data as XML.
Write and load changes by using a DiffGram.
Lab 5.1: Working with XML Data in ADO.NET
Building DataSets Module 6:from Existing Data Sources
The following topics are covered in this module:
Configuring a DataAdapter to Retrieve Information
Populating a DataSet Using a DataAdapter
Configuring a DataAdapter to Update the Underlying Data Source
Persisting Changes to a Data Source
How to Handle Conflicts
After completing this module, students will be able to:
Configure a DataAdapter to retrieve information.
Populate a DataSet by using a DataAdapter.
Configure a DataAdapter to modify information.

Persist data changes to a data source.
Manage data conflicts.
Lab 6.1: Retrieving Data into a Disconnected Application Lab 6.2: Retrieving and Updating Customers and Orders Data

Module 7: Building and Consuming a Web Service That Uses ADO.NET
The following topic is covered in this module:
Building and Consuming a Web Service That Returns Data
After completing this module, students will be able to:
Build a Web service.
Consume a Web service in a client application.
Troubleshoot errors in an ADO.NET application.
Lab 7.1: Troubleshooting an ADO.NET Application

Course *2565:* Developing .NET Windows Applications (Microsoft Visual Basic .NET)

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 application

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.

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
Helpting 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.

Course 2640: Upgrading Web Development Skills from ASP to Microsoft ASP.NET

This course can This course is ideal for anyone who wants build Web applications by using ASP.NET. also be used to prepare for exam:

Description:

The course provides the information that developers need to know to successfully upgrade their Web development skills from being able to create Active Server Pages (ASP) to Microsoft® ASP.NET. The material that comprises this course explains the differences in ASP versus ASP.NET, thereby creating an understanding of the efforts that are involved in getting an existing ASP Web application functioning in an ASP.NET environment. This course also reveals the new features of ASP.NET that can be used to improve an existing ASP Web application. Finally, this course provides students with several strategies for migrating ASP Web applications to ASP.NET, as well as a corresponding list of tasks that must be completed for successful migration. This course is intended for existing Web developers who are currently writing ASP Web applications.
Objectives:
At the end of the course, students will be able to:
In this course, you will upgrade your Web development skills from ASP to ASP.NET.
System Requirements:
To view this course, you need:
* Browser Netscape Communicator 4.0
* Browser Other: AOL 4.0
* Operating System MacOS *8.5*
* Browser Microsoft Internet Explorer 4.0
* Operating System MS-Windows *95*

Exam 70—315:and Implementing Web Applications with Microsoft Visual C#T41 .NET and Microsoft Visual Studio .NET Preparation Tools and Resources

We make a wealth of preparation tools and resources available to you, including courses, books, practice tests, and Microsoft Web sites. When you are ready to prepare for this exam, here's where you should start.

Instructor-led Courses for This Exam
Course 2310: Developing Microsoft ASP.NET Web Applications Using Microsoft Visual Studio .NET
Course 2389: Programming with Microsoft ADO.NET
Course 2640: Upgrading Web Development Skills from ASP to Microsoft ASP.NET

Course 2310: Developing Microsoft ASP.NET Web Applications Using Microsoft Visual Studio .NET

Lessons
Introduction to the .NET Framework
Overview of ASP.NET
Overview of the Lab Application
Resources
There is no lab for this module
After completing this module, students will be able to:
Explain the advantages of using the .NET Framework.
Understand the key functionality and purpose of using ASP.NET when developing Web applications.
Understand the basic functionality of the Web application that you will build in the labs throughout the course.
Module 2: Using Microsoft Visual Studio .NET
This module explains how to create new projects, and how to use the primary features that are available in Visual Studio .NET.

Lessons
Overview of Visual Studio .NET
Creating an ASP.NET Web Application Project
Lab 2: Using Microsoft Visual Studio .NET
Creating an ASP.NET Web Application Project Using Visual Studio .NET
After completing this module, students will be able to:
Navigate the Visual Studio .NET IDE.
Create, build, and view an ASP.NET Web application.
Module 3: Using Microsoft .NET-Based Languages
This module will introduce the various languages that support .NET. This module will focus on Visual Basic .NET and C#. Students will use Visual Studio .NET to create a class project and write code in either Visual Basic .NET or C#.
Lessons

Overview of the .NET-Based Languages
Comparison of the .NET-Based Languages
Creating a Component Using Visual Studio .NET
Lab 3: Building a Microsoft Visual Studio .NET Component
Create a new project in Visual Studio .NET for a Visual Basic class
After completing this module, students will be able to:
Identify the languages that support ASP.NET.
Choose an appropriate development language for their needs.
Create a component by using Visual Studio .NET.
Module 4: Creating a Microsoft ASP.NET Web Form
This module explains how to create and display an ASP.NET Web Form.
Lessons
Creating Web Forms
Using Server Controls
Lab 4: Creating a Microsoft ASP.NET Web Form
Creating the default.aspx Web Form
Creating the life.aspx Web Form

After completing this module, students will be able to:
Add a Web Form to an ASP.NET Web Application project.
Use the Visual Studio .NET toolbox to add server controls to a Web Form.
Module *5:* Adding Code to a Microsoft ASP.NET Web Form
This module explains how to add event procedures to an ASP.NET Web
application and add server controls on an ASP.NET Web Form. Examples
will be show in Visual Studio .NET.
Lessons
Using Code-Behind Pages
Adding Event Procedures to Web Server Controls
Using Page Events
Lab *5:* Adding Functionality to a Web Application
Creating a Page_Load Event Procedure
Creating a Click Event Procedure
After completing this module, students will be able to:

Use code-behind pages in an ASP.NET Web application.
Create event procedures for Web server controls.
Use Page events in an ASP.NET Web application.
Module 6: Tracing in Microsoft ASP.NET Web Applications
This module explains how to use the Trace feature and the Debug object in Visual Studio .NET. Students will learn about the two tracing techniques in
ASP.NET: page-level tracing and application-level tracing. Students will also learn how use the debugger to create breakpoints, set watch variables, and step between pages and components in a Web application.
Lessons
Understanding Tracing
Remote Debugging
Lab 6: Tracing in Microsoft ASP.NET Web Applications
Using Trace Statements
Tracing into a Component
After completing this module, students will be able to:

Use the Trace object to view runtime information about an ASP.NET Web application.
Debug Web applications remotely.
Module 7: Validating User Input
This module explains how to use the client-side and server-side validation
controls to screen data.
Lessons
Overview of User Input Validation
Using Validation Controls
Page Validation
Lab 7: Validating User Input
Using RequiredFieldValidator Controls
Using the ValidationSummary Control
Using the CompareValidator Control
Using the RegularExpressionValidator Control
After completing this module, students will be able to:
Identify when input validation is appropriate in Web Forms.

Use input validation controls to verify user input on a Web Form.
Verify that all validation controls on a page are valid.
Module 8: Creating User Controls
This module explains user controls and how to create them.
Lessons
Adding User Controls to an ASP.NET Web Form
Creating User Controls
Lab 8: Creating User Controls
Creating a User Control
Using the User Control
After completing this module, students will be able to:
Add a user control to an ASP.NET Web Form.
Create a user control.

Module 9: Accessing Relational Data Using Microsoft Visual Studio .NET
This module explains a conceptual overview of the objects in ADO.NET.
Lessons
Overview of ADO .NET
Creating a Connection to the Database
Displaying a DataSet in a List-Bound Control
Lab 9: Accessing Data Using Microsoft Visual Studio .NET
Connecting to a Database
Paging and Selection in a DataGrid Control
After completing this module, students will be able to:
Describe ADO.NET.
Create a connection to a database by using ADO.NET.
Display data in a Web Form by using a list-bound control.
Module 10: Accessing Data with Microsoft ADO.NET
This module explains how to manually add data access tools to a Web application.
Lessons
Introduction to Using ADO.NET
Connecting to a Database
Accessing Data with DataSets
Using Multiple Tables

Accessing Data with DataReaders
Lab 10: Accessing Data with Microsoft ADO.NET
Using a SqlDataReader
Viewing Data from the Database
After completing this module, students will be able to:
Describe the ADO.NET object model that is used for accessing data.
Create security-enhanced connections to a Microsoft SQL Server database by using the SqlConnection and SqlDataAdapter objects.
Use DataSet objects to support the local data storage and manipulation requirements of Web Forms.

Store multiple tables of data in a DataSet object, and then display that data in
DataGrid controls.
Programmatically read data from a SQL Server database by using a SqlDataReader object.
Module 11: Calling Stored Procedures with Microsoft ADO.NET
This module covers the more advanced and complicated features of ADO.NET.
Lessons
Overview of Stored Procedures
Calling Stored Procedures
Lab 11: Calling Stored Procedures with Microsoft ADO .NET
Calling Stored Procedures with Microsoft ADO.NET
After completing this module, students will be able to:
Explain what a stored procedure is and the reasons for using stored procedures
when accessing a database.
Call stored procedures.
Module 12: Reading and Writing XML Data
This module explains the methods that can be used for reading data from
XML files.
Lessons
Overview of XML Architecture in ASP.NET
XML and the DataSet Object
Working with XML Data

Using the XML Web Server Control
Lab 12: Reading XML Data
Reading a List from an XML File
Reading, Transforming, and Displaying XML
Nested Data
After completing this module, students will be able to:
Describe XML architecture in ASP.NET.
Read and write XML data into a DataSet object.

Identify how to store, retrieve, and transform XML data by using XMLDataDocument and XslTransform objects.
Use the XML Web server control to display, load, and save XML data.
Module 13: Consuming and Creating XML Web Services
This module explains the steps that are necessary to access a Web service
from an ASP.NET page and then incorporate that data into the Web application.
Lessons
Overview of Using XML Web Services
Calling an XML Web Service by HTTP
Using a Proxy to Call an XML Web Service
Creating an XML Web Service
Lab 13: Creating a XML Web Service
Create an XML Web service
Create an XML Web service method
Consume an XML Web service method
After completing this module, students will be able to:
Describe the purpose and process behind calling an XML Web service from a
Web Form.
Call an XML Web service directly from a browser by using Hypertext Transfer Protocol (HTTP).
Create a Web reference proxy for an XML Web service Web method and call
the method from a Web From.
Use the templates in Visual Studio .NET to create an XML Web service.

Module 14: Managing State
This module explains the several methods that are available for storing application and session data, for both short- and long-term storage.
Lessons
State management
Application and Session Variables
Cookies and Cookieless Sessions
Lab 14: Storing Application and Session Data

Using Session Variables
Using Cookies
Using Application Variables
Storing Session Variables in a Database
After completing this module, students will be able to:
Describe state management and its different types of options that are available to manage state in an ASP.NET Web application.
Use application and session variables to manage state in ASP.NET Web applications.
Use cookie and cookieless sessions to manage state in ASP.NET Web applications
Module 15: Configuring, Optimizing, and Deploying a Microsoft ASP.NET
Web Application
This module explains how to configure and deploy an ASP.NET Web application.
Lessons
Using the Cache Object
Using ASP.NET Output Caching
Configuring an ASP.NET Web Application
Deploying an ASP.NET Web Application
Lab 15: Configuring, Optimizing, and Deploying a Microsoft ASP.NET
Application
Using the Cache object
Using the Page Output Cache
Partial Page Caching
Using Dynamic Properties

Deploying Your Site
After completing this module, students will be able to:
Use the Cache object to store information.
Use ASP.NET output caching to store Web pages and Web page fragments.
Configure an ASP.NET Web application by using the Machine.config and
Web.config files.
Deploy an ASP.NET Web application.

Module 16: Helping to Protect a Microsoft ASP.NET Web Application
This module explains how to help protect a Web application by using a variety of technologies.
Lessons
Web Application Security Overview
Working with Windows-Based Authentication
Working with Forms-Based Authentication
Overview of Microsoft Passport Authentication
Lab 16: Securing a Microsoft ASP.NET Web Application
Securing Your Web Site Using Windows-Based Authentication
Securing Your Web Site Using Forms-Based Authentication
Registering New Users
Permitting Users to Sign Out
After completing this module, students will be able to:
Describe the ASP.NET and Internet Information Services (ITS) authentication
methods.
Use Microsoft Windows-based authentication to help protect ASP.NET Web
applications.
Use Forms-based authentication to help protect ASP.NET Web applications.
Use Microsoft Passport to help protect ASP.NET Web applications.

Preparation Guide for Exam 70-316

Preparation Tools and Resources

We make **a** wealth of preparation tools and resources available to you, including courses, books, practice tests, and Microsoft Web sites. When

you are ready to prepare for this exam, here's where you should start.
Instructor-led Courses for This Exam
Course 2389: Programming with Microsoft ADO .NET
Course *2555:* Developing Microsoft .NET Applications for Windows (Visual
C# .NET)

Course *2555:* Developing Microsoft .NET Applications for Windows (Visual C# .NET)

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 C# .NET curriculum and is intended to provide C# 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, Web Services consumption, debugging, security, and deployment issues for desktop applications.

Course Content:

Introducing Windows Forms
Creating a Form, Adding Controls to a Form, Creating an Inherited Form, Organizing Controls on a Form, and Creating MDI Applications
Working with Controls
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
Building Controls
Extending and Creating Controls, Adding Design-Time Support for Controls, Licensing a Control
Using Data in Windows Forms Applications

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, and Persisting Data
Interoperating with Managed Objects
Using .NET and COM Components in a Windows Forms Application, Calling
Win32 APIs from Windows Forms Applications
Printing and Reporting in Windows Forms Applications
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
Asynchronous Programming
The .NET Asynchronous Programming Model, The Asynchronous Programming Model Design Pattern, How to Make Asynchronous Calls to Any Method, Helping Protect State and Data in a Multithreaded Environment
Enhancing the Usability of Applications
Adding Accessibility Features, Adding Help to an Application, and Localizing an Application
Deploying Windows Forms Applications
.NET Assemblies, and Deploying Windows Forms Applications
Securing Windows Forms Applications
Security in the .NET Framework, Using Code Access Security, and Using Role-Based Security

Exam 70—310: Developing XML Web Services and Server Components with Microsoft Visual

Preparation Guide for course 2524

Introduction

The goal of this course is to provide students with the knowledge and skills that are required to develop Extensible Markup Language (XML) Web services-based solutions to solve common problems in the distributed

application domain. The course focuses on using Microsoft Visual Studio .NET and Microsoft ASP.NET to enable students to build, deploy, locate, and consume Web services.

Course Materials

The student kit includes **a** comprehensive workbook and other

necessary
materials for this class.
Course Outline
Module 1: The Need for XML Web Services
This module provides students with an understanding of the problem space that Web services address. The module compares various approaches to implementing distributed applications. Because the Web services in this course are implemented by using Microsoft ASP.NET and the Microsoft .NET Framework, alternate options for implementing distributed applications by using the .NET Framework are discussed to better define what kinds of solutions Web services are appropriate for.
After completing this module, you will be able to explain how Web services emerged as a solution to the problems with traditional approaches to designing distributed applications. This includes:
Describing the evolution of distributed applications.
Identifying the problems with traditional distributed application architectures and technologies.
Describing Web services and briefly explaining how they address the design problems in traditional distributed applications.
Listing the alternate options for distributed application development.
Identifying the kinds of scenarios where Web services are an appropriate solution.

Module 2: XML Web Service Architectures
This module broadly describes the service-oriented architecture, which is a conceptual architecture. Then, the module explains the roles and how Web service architectures are a type of service-oriented architecture.
After completing this module, you will be able to describe the architecture of a Web services-based solution. This includes:
Identifying how Web service architectures are a type of service-oriented architecture.
Describing the elements of a Web service architecture and explaining their roles.
Describing the Web service programming model.
Module 3: The Underlying Technologies of XML Web Services
After completing this module, you will be able to describe the

underlying technologies of Web services and explain how to use the .NET Framework to implement Web services by using these technologies.
This includes:
Describing the structures of an HTTP request and response.
Issuing HTTP POST and GET requests and processing the responses by using
the .NET Framework.
Describing data types by using the XML Schema Definition language (XSD).
Explaining how to control the way a .NET Framework object is serialized to
XML.
Describing the structures of a Simple Object Access Protocol (SOAP) request
and response.
Issuing a SOAP request and processing the response by using the .NET Framework.
Module 4: Consuming XML Web Services
After completing this module, you will be able to implement a Web service consumer by using Visual Studio .NET.

This includes:
Explaining the structure of a Web Service Description Language (WSDL) document.
Explaining the Web services discovery process.
Locating service contracts by using Disco.exe.
Generating Web service proxies by using Wsdl.exe.
Implementing a Web service consumer by using Visual Studio .NET.
Invoking a Web service synchronously and asynchronously by using a Web service proxy.
Module *5:* Implementing a Simple XML Web Service
This module provides students with the skills that are required to implement and debug a Web service by using Visual Studio .NET.
This includes:
Creating a Web service project.
Implementing Web service methods, exposing them, and controlling

their behavior.
Managing state in an ASP.NET-based Web service.
Debugging Web services.
Module 6: Publishing and Deploying XML Web Services
This module teaches students how to deploy and publish Web services and locate Web services by using the Microsoft Universal Description, Discovery, and Integration (UDDI) software development kit (SDK). A local development UDDI registry is used in the demonstrations for this module, but the mechanics of publishing and finding Web services is no different on the public UDDI registry nodes.
After completing this module, you will be able to publish and deploy a Web
service. This includes:
Explaining the role of UDDI in Web services.
Publishing a Web service in a UDDI registry by using the UDDI SDK.
Searching a UDDI registry to locate Web services by using the UDDI SDK.
Explaining the various options for publishing a Web service on an intranet.

Explaining some of the options for modifying the default configuration of a Web service.
Module 7: Securing XML Web Services
This module teaches students how to use the security services of the Microsoft
Windows operating system, Microsoft Internet Information Services (ITS), and
the.NET Framework and common language runtime to secure Web services.
After completing this module, you will be able to secure a Web service. This includes:
Identifying the differences between authentication and authorization.
Explaining how to use the security mechanisms that Microsoft Internet Information Services (ITS) and Windows provide for authentication.
Using SOAP headers for authentication in a Web service.
Using role-based security and code access security for authorization in a Web service.

Encrypting the communication between a Web service consumer and a Web service.

Module 8: Designing XML Web Services

This module teaches students which design issues to consider when designing real-world Web services. The issues discussed are related to data type constraints, performance, reliability, versioning, deployment in Internet Service Provider (ISP) and Application Service Provider (ASP) scenarios, and aggregating Web services. The module also discusses HTML screen scraping as a pseudo Web service.

After completing this module, you will be able to evaluate the trade-offs and issues that are involved in designing a real-world Web service. This includes:

Identifying the restrictions that are imposed on data types by the various Web services protocols.

Explaining how the use of Application and Session state can affect the performance and scaling of Web services.

Explaining how to use output and data caching to improve Web service performance.

Implementing caching in a Web service.

Explaining how asynchronous Web service methods can improve performance.

Explaining the need for instrumenting Web services.

Identifying the components of a Web service that can be versioned.

Explaining how to implement a virtual Web service by using screen scraping. Implementing a Web service that uses multiple Web services.

Identifying the trade-offs in the techniques that are used for exposing aggregated Web services.

Module 9: Global XML Web Services Architecture

This module teaches students how to use the security services of the Microsoft Windows operating system, ITS, and the .NET Framework and common language runtime to secure Web services.

After completing this module, you will be able to:

Describe limitations inherent to the specifications with which today's Web services are built.

Describe the design principles and specifications of Global XML Web services Architecture (GXA).

Describe Web service application scenarios made possible by Web Services Routing Protocol (WS-Routing) and Web Services Referral Protocol (WSRefelTal).
Explain how to use Web Services Security Language (WS-Security) and Web Services License Language (WS-License) to perform authentication and authorization for Web services.
Design Web services that anticipate and can leverage the features that GXA will offer when released.

Preparation Guide for course *2557*

Introduction

This five-day, instructor-led course provides students with the knowledge and skills to effectively build scalable, distributed applications that use Microsoft .NET Enterprise Services and the Microsoft .NET Framework.

Course Materials

The student kit includes a comprehensive workbook and other necessary materials for this class.The following software is provided in the student kit:
Evaluation copy of Microsoft Windows XP Professional for classroom use only
Course Outline
Module 1: Introduction to COM+ Services
This module covers the evolution of applications from monolithic applications to client/server applications to component-based applications and the supporting application infrastructure that COM+ services provides. The module also covers the COM+ runtime architecture and how it uses surrogates, context, and interception to provide services to components.
Lessons
History of Server-Based Applications
The COM+ Runtime Architecture
After completing this module, students will be able to:
Describe the history of server-based applications.
Describe the COM+ runtime architecture.
Module 2: Configuring Just-in-Time Activation and Synchronization
This module describes the attributes that you can assign to components

and how to write a serviced component. This module also describes how to access the object context from within code, ITT activation, synchronization, the relationship between synchronization and ITT activation, and how you can set ITT activation and synchronization for a component.
Lessons
The .NET Enterprise Services Programming Model
ITT Activation
Synchronization
Lab 2: Configuring Just-in-Time Activation

Exercise 1: Creating a Serviced Component
Exercise 2: Using a Serviced Component
After completing this module, students will be able to:
Use attributes to configure an assembly as a COM+ application.
Create components that use JIT activation.
Create components that are synchronized.
Module 3: Using ADO.NET to Work With Data
This module describes how to run a query and retrieve a result set by using
ADO.NET. The module also covers how to pass parameters to a stored procedure, create typed DataSet objects, and use construction strings to specify connection information to establish a connection to a data source.
Lessons
The ADO.NET Architecture
Accessing a SQL Server Database
Lab 3: Using ADO.NET in a Serviced Component
Exercise 1: Creating a New Typed Dataset
Exercise 2: Updating the PurchasingSelect Component
Exercise 3: Updating the OrderProcessing Component
Exercise 4: Modifying the PlaceOrder Web Page
After completing this module, students will be able to:
Describe the ADO.NET architecture and namespace classes.
Use the classes provided by the SqlClient namespace to retrieve and update data from a Microsoft SQL Server 2000 database.
Module 4: Transaction Services

This module describes transaction processing, how it is implemented in .NET Enterprise Services, and how you add attributes to code to enable transaction processing.

Lessons

Introduction to Transaction Processing

.NET Enterprise Services Transactions

Lab 4: Using Transaction Services

Exercise 1: Creating a Transactional Component

Exercise 2: Updating the OrderProcessing Component

Exercise 3: Updating the OrderApp Web Application

Exercise 4: Testing with the OrderApproval Client

After completing this module, students will be able to:

Describe transaction processing and how it is implemented in .NET Enterprise Services.

Use the classes defined in the EnterpriseServices namespace to implement transaction processing.

Module *5:* Securing Enterprise Applications

This module explains how to implement COM+ role-based security in serviced components by using .NET Enterprise Services.

Lessons

Introduction to Application Security

Implementing COM+ Role-Based Security

Authentication and Impersonation

Lab *5:* Securing Enterprise Applications

Exercise 1: Updating the PurchasingUpdate component

Exercise 2: Updating the OrderProcessing Component

Exercise 3: Updating the OrderApproval Client

After completing this module, students will be able to:

Describe the security model offered by COM+ and how it is used with other security mechanisms.

Help protect your application by using COM+ role-based security.

Configure authentication and impersonation levels to balance security requirements with performance and flexibility requirements.

Module 6: State Management
This module describes how to manage state in .NET Enterprise Services. It
explains how to use the shared property manager (SPM) to store state, use
ASP.NET applications to store application and session state, and use ASP.NET caching.
Lessons
Introduction to State Management
Using the Shared Property Manager

Using ASP.NET to Store State
Lab 6: Managing Component State
Exercise 1: Updating the OrderProcessing Component
After completing this module, students will be able to:
Describe the need for state management and the techniques for implementing state management.
Maintain state by using the SPM.
Maintain state by using ASP.NET application and session state and ASP.NET caching.
Module 7: Compensating Resource Managers
This module describes the architecture of compensating resource managers
(CRMs) and how to implement CRMs.
Lessons
Introduction to Compensating Resource Managers
Implementing Compensating Resource Managers
Lab 7: Implementing Compensating Resource Managers
Exercise 1: Creating the OrderDocCRM Component
Exercise 2: Updating the OrderProcessing Component
After completing this module, students will be able to:
Describe the architecture of CRMs.
Implement a CRM.
Module 8: Loosely Coupled Events
This module describes the architecture of Loosely Coupled Events (LCEs) and the LCE system. This module also describes how to configure and implement publishers, subscribers, and event classes.

Lessons
Introduction to Loosely Coupled Events
COM+ Events
Using Loosely Coupled Events
Lab 8: Using Loosely Coupled Events
Exercise 1: Creating an Event Class
Exercise 2: Creating a Publisher

Exercise 3: Creating a Subscriber
After completing this module, students will be able to:
Describe why LCEs are needed.
Describe the architecture of the LCE system.
Configure the LCE system programmatically and by using the Component
Services administrative tool.
Implement publishers, subscribers, and event classes.
Module 9: Queued Components
The following topics are covered in this module:
Lessons
Introduction to Queuing
Developing Queued Components
Queued Components and Transactions
Lab 9: Creating a Queued Component
Exercise 1: Creating a Queued Component
Exercise 2: Calling the Queued Component
After completing this module, students will be able to:
List the advantages of using asynchronous messaging in a distributed systems
environment.
Explain the purposes of the recorder, listener, and player in the Queued Components architecture.
List additional component design considerations introduced by asynchronous
messaging.
Install and configure a queued component in a COM+ application.
Instantiate a queued component by using the queue and new monikers.
Module 10: Debugging COM+ Applications

In this module, you will learn how to debug applications that use .NET Enterprise Services.
Lessons
Debugging Tools
Common Debugging Scenarios
Lab 10: Debugging COM+ Applications
Exercise 1: Debugging DCOM Problems

Exercise 2: Debugging Transaction Problems
Exercise 3: Debugging Security Problems
After completing this module, students will be able to:
Debug COM+ applications by using tools such as the Microsoft Visual Studio
debugger.
Debug some common problems in COM+ applications.
Module 11: Deploying and Administering COM+ Applications
This module introduces the COMAdmin objects and additional techniques and tools for deploying and administering COM+ applications. It describes the advantages and drawbacks of each technique so that students can select the one most appropriate for your application.
Lessons
Deploying a COM+ Application Built Using .NET Enterprise Services
sing COMAdmin Objects in WSH Scripts
Lab 11: Administering COM+ Applications
Exercise 1: Creating a COM+ Application
Exercise 2: Adding a Role to a COM+ Application
Exercise 3: Deleting a COM+ Application
Exercise 4 (Optional): Exposing a COM+ Application as an XML Web Service
Exercise *5* (Optional): Creating the OrderDispatch Service
After completing this module, students will be able to:
Deploy a COM+ application.
Use Microsoft Windows Script Host (WSH) scripts to do common deployment and administrative tasks, such as creating an application or adding components to an application.
Module 12: COM+ *1.5* Enhancements

This module explains how to use new COM+ version 1.5 features that help you manage, scale, and maximize the uptime of your COM+ application more efficiently. The module also explains how to use .NET Enterprise Services without having to implement serviced components.
Lessons
S calability and Availability Enhancements

Manageability Enhancements
Other COM+ Features
After completing this module, students will be able to:
Use COM+ 1.5 features to improve the scalability and availability of COM+ applications.
Use COM+ 1.5 features to improve the manageability of COM+ applications. Use COM+ features such as SWC, BYOT, and Phase Zero notification in your applications.

Preparation Guide for course 2663

Introduction

This three-day instructor-led his course teaches developers how to program with XML in the Microsoft .NET Framework in order to build Microsoft Windows or Web-based integrated applications that are easier to build, scalable, and flexible.

pict0.jpg

pict1.jpg