

CertifyExpress



Blank area for text or content.



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

Abstract

<table border=1 cellpadding=5 cellspacing=0 width=90%><tr><td>

This ExamNotes Exam Information Guide intends to provide you with information to prepare for the Microsoft MCAD 70-305 Exam.

What is MCAD?

"The MCAD for Microsoft .NET credential is appropriate for professionals who use Microsoft technologies to develop and maintain department-level applications, components, Web or desktop clients, or back-end data services or work in teams developing enterprise applications." (from Microsoft's web site)

MCAD candidates are required to pass two core exams and one elective exam in an area of specialization:

Core Exams (two required): Choose a language to leverage your existing skills through the .NET Framework. It can be either VB or C#. One exam focused on either Web Application Development or Windows Application Development, and another one for Web Services and Server Components.

Before you start

This study guide provides you with information on the many different aspects of "Microsoft MCAD Exam 70-305". Before you proceed with this subject, please make sure you are 100% comfortable with the concept of networking.

Do NOT rely solely on this study notes for the exam. By all means read more than one book on the subject and make sure you understand the material well enough so that you could be ready for the questions. There is no quick way to succeed for this topic. Ideally you must work things out and gain experience before even trying to sign up for the exam.

Your Study Track for MCAD 70-305

1, Know the ins and outs of web authoring and programming. Know ASP. Visit the following tutorial sites:

ASP:

<http://stardeveloper.com:8080/articles/041600-1.shtml>

<http://www.big-boys.com/toprated/tutorials/asptutorials.asp>

<http://www.manastungare.com/asp/practicalguide.asp>

<http://www.allarticles.com/default.asp?action=home&pid=40>

2, Make sure you are comfortable with the concept of web application development. You will find the following books useful:

A Methodology for Client/Server and Web Application Development

by Roger Fournier (Textbook Binding - September 1998)

Testing Applications on the Web: Test Planning for Internet-Based Systems

by Hung Quoc Nguyen (Paperback)

3, Make sure you understand .NET thoroughly. Reference the following books:

Building Web Services and .NET Applications (Application Development)

by Lonnie Wall, Andrew Lader (Paperback)

Visual Studio .NET: The .NET Framework Black Book

by Julian Templeman, David Vitter (Paperback)

Fundamentals of Web Applications Using .Net and XML

by Eric Bell (Editor), et al (Paperback)

4, Review the exam objectives for 70-305:

Creating User Services

Create ASP.NET pages.

Add and set directives on ASP.NET pages.

Separate user interface resources from business logic.

Add Web server controls, HTML server controls, user controls, and HTML code to ASP.NET pages.

Set properties on controls.

Load controls dynamically.

Apply templates.

Set styles on ASP.NET pages by using cascading style sheets.

Instantiate and invoke an ActiveX® control.

Implement navigation for the user interface.

Manage the view state.

Manage data during postback events.

Use session state to manage data across pages.

Validate user input.

Validate non-Latin user input.

Implement error handling in the user interface.

Configure custom error pages.

Implement Global.asax, application, page-level, and page event error handling.

Implement online user assistance.

Incorporate existing code into ASP.NET pages.

Display and update data.

Transform and filter data.

Bind data to the user interface.

Use controls to display data.

Instantiate and invoke Web services or components.

Instantiate and invoke a Web service.

Instantiate and invoke a COM or COM+ component.

Instantiate and invoke a .NET component.

Call native functions by using platform invoke.

Implement globalization.

Implement localizability for the user interface.
Convert existing encodings.
Implement right-to-left and left-to-right mirroring.
Prepare culture-specific formatting.
Handle events.

Create event handlers.
Raise events.
Implement accessibility features.

Use and edit intrinsic objects. Intrinsic objects include response, request, session, server, and application.

Retrieve values from the properties of intrinsic objects.
Set values on the properties of intrinsic objects.
Use intrinsic objects to perform operations.

Creating and Managing Components and .NET Assemblies

Create and modify a .NET assembly.
Create and implement satellite assemblies.
Create resource-only assemblies.
Create custom controls and user controls.

Consuming and Manipulating Data

Access and manipulate data from a Microsoft SQL Server™ database by creating and using ad hoc queries and stored procedures.

Access and manipulate data from a data store. Data stores include relational databases, XML documents, and flat files. Methods include XML techniques and ADO.NET.

Handle data errors.

Testing and Debugging

Create a unit test plan.
Implement tracing.
Add trace listeners and trace switches to an application.
Display trace output.
Debug, rework, and resolve defects in code.

Configure the debugging environment.

Create and apply debugging code to components, pages, and applications.

Provide multicultural test data to components, pages, and applications.

Execute tests.

Resolve errors and rework code.

Deploying a Web Application

Plan the deployment of a Web application.

Plan a deployment that uses removable media.

Plan a Web-based deployment.

Plan the deployment of an application to a Web garden, a Web farm, or a cluster.

Create a setup program that installs a Web application and allows for the application to be uninstalled.

Deploy a Web application.

Add assemblies to the global assembly cache.

Maintaining and Supporting a Web Application

Optimize the performance of a Web application.

Diagnose and resolve errors and issues.

Configuring and Securing a Web Application

Configure a Web application.

Modify the Web.config file.

Modify the Machine.config file.

Add and modify application settings.

Configure security for a Web application.

Select and configure authentication type. Authentication types include Windows® Authentication, None, forms-based, Microsoft Passport, Internet Information Services (IIS) authentication, and custom authentication.

Configure authorization. Authorization methods include file-based methods and URL-based methods.

Configure role-based authorization.

Implement impersonation.

Configure and implement caching. Caching types include output, fragment, and data.

Use a cache object.

Use cache directives.

Configure and implement session state in various topologies such as a Web garden and a Web farm.

Use session state within a process.

Use session state with session state service.

Use session state with Microsoft SQL Server.

Install and configure server services.

Install and configure a Web server.

Install and configure Microsoft FrontPage® Server Extensions

ASP

ASP is a specification for a dynamically created Web page. It utilizes ActiveX scripting. When a browser requests an ASP page, the Web server generates a page with HTML code and sends it back to the browser.

Visual Basic

It is a programming language and environment that provides a graphical programming environment and a paint metaphor for developing user interfaces. It is not truly object-oriented, but is event-driven in a sense that each object can react to different events such as a mouse click.

VB Script

Shorts for Visual Basic Scripting Edition, it is a scripting language supported by Microsoft's Internet Explorer Web browser that enables Web authors to include interactive controls on their Web pages. It can be thought of as the "lite" version of Visual Basic.

However, ASP.NET no longer emphasizes the use of VBScript. Instead, VB and C# are used as the primary languages.

C#

It is a hybrid of C and C++ developed by Microsoft. It is object-oriented, and is designed to be used with XML -based Web services on the .NET platform. It boasts the following features:

- type-safety
- garbage collection
- simplified type declarations
- versioning and scalability support

.NET

According to webopedia.com,

"It is a Microsoft operating system platform that incorporates applications, a suite of tools and services and a change in the infrastructure of the company's Web strategy. The objective of .NET is to bring users into the next generation of the Internet by conquering the deficiencies of the first generation and giving users a more enriched experience in using the Web for both personal and business applications.

This is Microsoft's most ambitious undertaking since the release of Windows 3.0."

The four main principles of .NET are:

- erasing boundaries between applications and the Internet.
- allowing software rental as a hosted service over the Internet.
- allowing user access to their information on the Internet from any device, anytime, anywhere.
- introducing new ways to interact with application data.

Common Language Runtime:

- CLR
- forms the foundation of the .NET Framework
- manages the execution of .NET code
- verifies that code is type-safe
- enforces Code Access Security
- ensure version compatibility between application components
- manages memory and threads.
- includes the just-in-time compilers
- all code created to target .NET is referred to as "managed code"

Microsoft Intermediate Language:

- MSIL
- CPU-independent
- tokenized set of instructions
- when executed, invoke the CLR

.NET compilers:

- VB.NET
- C++ with Managed Extension
- JScript.NET
- C#

Common Type System:

- CTS
- defines a subset of common data types
- defines how new data types can be declared
- allows all languages to share a basic set of types and rules for extending types
- allows .NET to support cross language integration

Assemblies:

- building blocks of .NET applications
- portable executable DLL or EXE
- self-describing with the help of meta data
- contain the following elements:
 - ◆ PE Header
 - ◆ Unmanaged stub
 - ◆ Meta data
 - ◆ MSIL code
 - ◆ Resources

.NET Framework Class Library

- an object-oriented library of classes and other types
- comes with the .NET Framework SDK
- classes are organized hierarchically in namespaces
- supports all .NET applications, regardless of the language or the application type
- provides access to system functionality
- implements the classes upon which .NET Web Services are built

ASP.Net

- the .NET rewrite of ASP
- essentially a web service
- requires IIS 5.0 to run
- ASP.NET applications are common language run-time managed applications
- can be written in any .NET compatible language

Essential Elements

Declarations

- introduces the named entities
- represent the "meaning" of the program.
- entity that can contain other entities defines a declaration space
- other entities are introduced into such a declaration space through either declarations or inheritance
- it is invalid for declarations to introduce identically named entities of the same kind into the same declaration context
- a declaration space may never contain different kinds of entities with the same name
- two namespace declarations with the same fully qualified name contribute to the same declaration space

Inheritance

- a relationship in which one type derives from another
- the derived type's declaration space implicitly contains all of the accessible nonconstructor type members of the base type

Implementation

- exists when a type declares that it implements an interface and the type implements all the type members of the interface
- a type that implements a particular interface is convertible to that interface
- a type implementing an interface with multiply inherited type members must implement those methods, although such implementation can be deferred

Polymorphism

- provides the ability to vary an Overridable method's implementation
- allows the same base method to perform different actions
- decision made based on the run-time type of the instance that invokes the method
- when a non-overridable method is invoked, the compile-time type of the instance is the determining factor.
- overridable method may also be MustOverride
- MustOverride methods are only allowed in MustInherit classes
- a MustOverride method declaration cannot include either the Shared nor Overridable modifiers.

Access Types

- specified by a declaration
- does not change the scope of an entity's name
- five access types:
 - ◆ Public
 - ◆ Protected
 - ◆ Friend
 - ◆ Protected Friend
 - ◆ Private
- access type for a declaration is specified via an optional access modifier
- the default access type depends on the declaration context
- the permitted access types also depend on the declaration context

Scope

- the set of all declaration spaces within which it is possible to refer to that name without qualification
- an entity's declaration may contain nested declarations of entities with the same name
- shadowing through nesting is allowed

Attributes

- programmer may specify declarative information about entities defined in the program
- programmers may invent new kinds of declarative information
- new kinds of declarative information are defined through the declaration of attribute classes

Grammar Summary

Available at Microsoft's MSDN site:

<http://msdn.microsoft.com/library/en-us/vbls7/html/vblrfvbspec12.asp>

Section by section Language Reference:

<http://msdn.microsoft.com/library/en-us/vbls7/html/vbSpecStart.asp>

Hands-on

Obtaining Visual Studio.NET

<http://msdn.microsoft.com/vstudio/howtobuy/default.asp>

Code examples of VB.NET:

<http://abstractvb.com/code.asp?F=1>

VB.NET Beta 2 Code Samples:

<http://msdn.microsoft.com/vbasic/downloads/samples/default.asp>

VB.NET add-ins and components:

<http://msdn.microsoft.com/vbasic/downloads/addins.asp>

Further Readings

Coding Techniques for Microsoft(r) Visual Basic(r) .NET

by John Connell

Coding Techniques for Microsoft(r) Visual Basic(r) .NET

by John Connell

Microsoft Visual Basic .NET Language Reference

by Microsoft Corporation (Editor), Microsoft Corporation

Microsoft(r) Visual Basic(r) .NET Step by Step

by Michael Halvorson

Professional VB.NET Transactions

by Matthew Bortniker, James Conard

