

# CertifyExpress



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

## Abstract

This ExamNotes Exam Information Guide intends to provide you with information to prepare for the Microsoft MCAD/MCSD 70-320 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.

## What is MCSD.Net?

MCSD.Net is the next generation of the MCSD certification. An entirely new set of requirements are outlined by Microsoft:

*“Microsoft Certified Solution Developer (MCSD) for Microsoft .NET candidates are required to pass four core exams and one elective exam that provide a valid and reliable measure of technical proficiency and expertise in developing and maintaining enterprise applications based on Microsoft development tools, technologies, and platforms. The elective exam provides proof of expertise with a specific Microsoft server product.”*

According to Microsoft, MCSD.NET is:

*"... appropriate for professionals who design and develop leading-edge enterprise solutions with Microsoft development tools, technologies, platforms, and the Microsoft .NET Framework. The MCSD job role includes analyzing business and technical requirements, and defining the solution architecture, as well as the tasks typically conducted by MCADs—implementing the requirements and building, deploying, and maintaining the solution."*

**The 70-320 Developing XML Web Services and Server Components with Microsoft Visual C# .NET and the Microsoft .NET Framework exam can satisfy BOTH MCSD and MCAD.**

## Before you start

This study guide provides you with information on the many different aspects of "Microsoft MCAD/MCSD Exam 70-320". Before you proceed with this subject, please make sure you are 100% comfortable with the concept of networking, programming and development.

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.

**Exam 320 tests the combination of C# and XML, while exam 310 tests VB.NET and XML.**

## Your Study Track for MCAD/MCSD 70-320

**1, Know the ins and outs of web authoring and programming using COM and C#.**

The COM and COM+ Programming Primer -- by Alan Gordon; Paperback

Our Price: \$34.99 -- Or buy used from \$12.98

Applied Microsoft .NET Framework Programming -- by Jeffrey Richter; Paperback

Our Price: \$34.99 -- Or buy used from \$34.00

.NET and COM: The Complete Interoperability Guide  
by Adam Nathan (Paperback)

Microsoft Visual C# (Core Reference)  
by Mickey Williams (Paperback)

Programming Windows(r) with C# (Core Reference)  
by Charles Petzold (Hardcover - December 2001)

C# and the .NET Platform  
by Andrew Troelsen (Paperback - June 2001)

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

Beginning XML -- by David Hunter, et al; Paperback

Learning XML  
by Erik T. Ray (Paperback - February 2001)

Understanding Web Services: XML, WSDL, SOAP, and UDDI  
by Eric Newcomer (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 70-320:

- Creating and Managing Microsoft Windows Services, Serviced Components, .NET Remoting Objects, and XML Web Services
- Consuming and Manipulating Data
- Testing and Debugging
- Deploying Windows Services, Serviced Components, .NET Remoting Objects, and XML Web Services

## What is XML?

The Extensible Markup Language XML is:

*"... a specification developed by the W3C. XML is a pared-down version of SGML, designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations."*

<http://www.webopedia.com/TERM/X/XML.html>

**Remember, this 70-320 exam focuses on XML in the context of .NET.**

Please read the examnotes for 70-310 on the topics of:

- XmlReader
- XPath
- XPathNavigator
- XSD
- SOM
- DOM
- SOAP
- .NET

This exam tests heavily on .NET remoting object. This is an extremely important topic.

## XML Web Service

Dave Mendlen in his article "Visual Studio .NET: Build Web Applications Faster and Easier Using Web Services and XML" talks about the XML Web Service as below:

*"The heart of the Microsoft .NET Framework is a common language runtime that manages the needs of running code written in any Visual Studio programming language. This runtime supplies many services that help simplify code development and application deployment while also improving application reliability. The framework also supplies a set of class libraries that developers can use from any programming language. The framework provides specific support for building traditional Windows-based applications, Web applications, Web Services, and components. ..."*

*In Visual Studio .NET, you can easily expose any function—in any language—as a Web Service. There is no need to learn XML and SOAP to take advantage of Web Services. When you compile your business objects, Visual Studio .NET will automatically generate an XML file that describes the function, and when it is called the function will automatically send and receive XML packets.*

*After the Web Service has been built, both the compiled code and the XML file describing the public methods of the service are published to the Web server. The Web Service can now be invoked via HTTP, and XML will automatically be used to pass data to and from the service.*

*In Visual Studio .NET, you can drag any exposed Web Service directly into your application. Doing so enables Visual Studio to treat the Web Service as a class. Calling the Web Service is as simple as creating a new instance of the Web Service class and then calling its exposed methods."*

<http://msdn.microsoft.com/msdnmag/issues/0900/vsnet/vsnet.asp>

Below is a summary of the XML Web services:

- fundamental building block of distributed computing on the Internet
- platform for application integration - applications are constructed using multiple XML Web services from various sources that work together regardless of where they reside or how they were implemented.
- can be built on XML, SOAP, WSDL, and UDDI specifications - baseline specifications:
- Baseline XML Web Services Specifications are:
  - ◆ SOAP
  - ◆ UDDI
  - ◆ WSDL
- Global XML Web Services Specifications are:
  - ◆ WS-Inspection
  - ◆ WS-Referral
  - ◆ WS-Routing
  - ◆ WS-Security

To know more about the XML Web Service, refer to the following links:

Web Services: Building Reusable Web Components with SOAP and ASP .NET

<http://msdn.microsoft.com/msdnmag/issues/01/02/webcomp/webcomp.asp>

Visual Studio .NET: Build Web Applications Faster and Easier Using Web Services and XML

<http://msdn.microsoft.com/msdnmag/issues/0900/vsnet/vsnet.asp>

Web Services Specifications

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/wsspeccover.asp>

## What is the relationship between XML and C#?

C# 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

As said by Microsoft,

*"C# programmers can leverage an extensive framework for building applications on the Microsoft .NET platform. C# includes built-in support to turn any component into an XML Web service that can be invoked over the Internet—from any application running on any platform.*

*Even better, the XML Web services framework can make existing XML Web services look just like native C# objects to the programmer, thus allowing developers to leverage existing XML Web services with the object-oriented programming skills they already have.*

*There are more subtle features that make C# a great Internet programming tool. For instance, XML is emerging as the standard way to pass structured data across the Internet. Such data sets are often very small. For improved performance, C# allows the XML data to be mapped directly into a struct data type instead of a class. This is a more efficient way to handle small amounts of data."*

**More information on C# can be obtained from**

<http://msdn.microsoft.com/vstudio/techinfo/articles/upgrade/Csharpintro.asp>

**Also visit the Community and Launch site at:**

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncscol/html/csharp03122002.asp>

## Abstract Base Classes

- 2 .NET Framework XML abstract classes: XmlReader and XmlWriter.
- XmlReader provides a fast, forward-only, read-only cursor for processing an XML document stream.
- XmlWriter provides an interface for producing XML document streams that conform to the W3C's XML 1.0 + Namespaces Recommendations.
- Applications that want to process XML documents consume XmlReader
- Applications that want to produce XML documents consume XmlWriter
- Both XmlReader and XmlWriter define the functionality that all derived classes must support.
- Three concrete implementations of XmlReader—XmlTextReader, XmlNodeReader, and XslReader—as well as two concrete implementations of XmlWriter: XmlTextWriter and XmlNodeWriter

Read this MSDN article at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmag01/html/xml0101.asp>

# What is .NET Remoting?

MSDN describes .NET Remoting as a framework that

*"... provides a rich and extensible framework for objects living in different AppDomains, in different processes, and in different machines to communicate with each other seamlessly. .NET Remoting offers a powerful yet simple programming model and runtime support for making these interactions transparent. In this article we will take a look at the different building blocks of the Remoting architecture, as well as explore some of the common scenarios in which .NET Remoting can be leveraged."*

The .NET Remoting Framework Overview document says that

*"The framework provides a number of services, including activation and lifetime support, as well as communication channels responsible for transporting messages to and from remote applications. Formatters are used for encoding and decoding the messages before they are transported by the channel. Applications can use binary encoding where performance is critical, or XML encoding where interoperability with other remoting frameworks is essential. All XML encoding uses the SOAP protocol in transporting messages from one application domain to the other. Remoting was designed with security in mind, and a number of hooks are provided that allow channel sinks to gain access to the messages and serialized stream before the stream is transported over the channel."*

## .NET Remoting Objects

Activation and deactivation of remote objects:

- Client-activated objects - under the control of a lease-based lifetime manager that ensures that the object is garbage collected when its lease expires
- Server-activated objects – you can have a choice of selecting either a "single call" or "singleton" model - lifetime of singletons are also controlled by lease-based lifetime.
- all remote objects have to be registered with the remoting framework before clients can access them
- Object registration is normally done by a hosting application that:
  - ◆ starts up
  - ◆ registers one or more channels with ChannelServices
  - ◆ registers one or more remote objects with RemotingConfiguration
  - ◆ waits until it is terminated
- registered channels and objects are only available while the process that registered them is alive.
- information required when registering a remote object with the framework:
  - ◆ assembly name in which the class is contained.
  - ◆ type name of the remote object.
  - ◆ object URI that clients will use to locate the object.
  - ◆ object mode required for server activation

- remote object can be registered by:
  - ◆ calling RegisterWellKnownServiceType
  - ◆ passing the information above as parameters
  - ◆ storing the above information in a configuration file and then calling Configure

#### .NET Remoting Objects:

- Single Call
  - ◆ service one and only one request coming in
  - ◆ useful in scenarios where the objects are required to do a finite amount of work
  - ◆ usually not required to store state information
  - ◆ cannot hold state information between method calls
  - ◆ can be configured in a load-balanced fashion
  
- Singleton Objects
  - ◆ service multiple clients
  - ◆ share data by storing state information between client invocations
  - ◆ useful in cases in which data needs to be shared explicitly between clients
  - ◆ useful in which the overhead of creating and maintaining objects is substantial
  
- CAO
  - ◆ Client-activated objects
  - ◆ server-side objects
  - ◆ activated upon request from the client
  - ◆ When the client submits a request for a server object using "new" operator, an activation request message is sent to the remote application
  
- Passing Objects with .NET Remoting:
  - ◆ As parameters in method calls
  - ◆ Return Value of method calls
  - ◆ Values resulting from property
  - ◆ Values resulting from field access of a .NET component
    - objects that are Marshal By Value (MBV) - complete copy of the object is made when the object is passed
    - objects that are Marshal By Reference (MBR) - reference to the object is made when passing

- Any object can be changed into a remote object by deriving it from MarshalByRefObject
- When a client activates a remote object, it receives a proxy to the remote object. All operations on this proxy are appropriately indirected to enable the remoting infrastructure
- In cases where the proxy and remote objects are in different application domains, all method call parameters on the stack are converted into messages and transported to the remote application domain, where the messages are turned back into a stack frame and the method call is invoked.

## Proxy Objects

- created when a client activates a remote object
- acts as a representative of the remote object
- ensures that all calls made on the proxy are forwarded to the correct remote object instance
  - ◆ when a client activates a remote object, the framework creates a local instance of the class TransparentProxy that contains a list of all classes, as well as interface methods of the remote object
  - ◆ all method calls on the proxy are intercepted by the runtime
  - ◆ the call is examined to determine if it is a valid method of the remote object and if an instance of the remote object resides in the same application domain as the proxy.
  - ◆ a simple method call is routed to the actual object if in the same domain
  - ◆ if the object is in a different application domain, the call parameters on the stack are packaged into an IMessage object and forwarded to a RealProxy class by calling its Invoke method

## ObjRef

- A remote object is registered in an application domain on a remote machine. The object is marshaled to produce an ObjRef.
- ObjRef contains all the information required to locate and access the remote object from anywhere on the network.
- Information includes:
  - ◆ the strong name of the class
  - ◆ the class's hierarchy (its parents)
  - ◆ the names of all the interfaces the class implements
  - ◆ the object URI
  - ◆ details of all available channels that have been registered
- remoting framework uses the object URI to retrieve the ObjRef instance created for the remote object when it receives a request for that object.
- A client activates a remote object by calling new or one of the Activator functions like CreateInstance.
- In the case of server-activated objects, the TransparentProxy for the remote object is produced in the client application domain and returned to the client - no remote calls are made at all.
- During unmarshaling, the ObjRef is parsed to extract the method information of the remote object and both the TransparentProxy and RealProxy objects are created. The content of the parsed ObjRef is added to the internal tables of the TransparentProxy before the latter is registered with the CLR.

- TransparentProxy is an internal class that cannot be replaced or extended.
- The RealProxy and ObjRef classes are public
- The RealProxy and ObjRef classes can be extended and customized when necessary.
- The RealProxy class is an ideal candidate for performing load balancing

## Channels

- used to transport messages to and from remote objects
- parameters and details related to the calls are transported through the channel to the remote object
- client can select any of the channels registered on the "server" to communicate with the remote object
- developers can select the channels that best suit their needs
- it is possible to customize any existing channel or build new ones that use different communication protocols
- Clients can communicate with a remote object using any registered channel.
- All channels derive from IChannel and implement either IChannelReceiver or IChannelSender, depending on the purpose of the channel.
- Most channels implement both the receiver and sender interfaces to enable them to communicate in either direction.
- Channel selection is subject to the following rules:
  - ◆ At least one channel must be registered with the remoting framework before a remote object can be called.
  - ◆ Channels must be registered before objects are registered.
  - ◆ Channels are registered per application domain.
  - ◆ There can be multiple application domains in a single process.
  - ◆ When a process dies, all channels that it registers are automatically destroyed.
  - ◆ It is illegal to register the same channel that listens on the same port more than once.
- Keep in mind that remote objects share channels, that
  - ◆ a remote object does not own a channel
  - ◆ server applications that host remote objects have to register the channels they require as well as the objects they wish to expose with the remoting framework
  - ◆ when a channel is registered, it automatically starts listening for client requests at the specified port
  - ◆ when a request comes in on a channel, the remoting framework examines the message to determine the target object and checks the table of object references to locate the reference in the table

- HTTP Channel:
  - ◆ transports messages to and from remote objects using the SOAP protocol
  - ◆ all messages are passed through the SOAP formatter, where the message is changed into XML and serialized, and the required SOAP headers are added to the stream
  - ◆ you may configure the HTTP Channel to use the binary formatter
- TCP Channel:
  - ◆ uses a binary formatter to serialize all messages to a binary stream
  - ◆ it transports the stream to the target URI using the TCP protocol
  - ◆ you may configure the TCP channel to the SOAP formatter.

## What is UDDI?

UDDI = Universal Description, Discovery, and Integration

As stated in <http://uddi.microsoft.com/about/default.aspx>,

*"... UDDI involves the shared implementation of a Web Service based on the UDDI specifications. This Web Service -- the UDDI Business Registry -- is an Internet directory of businesses and the applications they have exposed as Web Services for trading partners and customers to use. Business programs will use the UDDI Business Registry to determine the specifications for programs at other companies in a manner similar to how people use Web search engines today to find websites. This automated application-to-application discovery and integration over the Internet will help eliminate many of the configuration and compatibility problems that are preventing businesses from more widely adopting B2B, despite B2B's potential for cost savings and improved efficiency."*

Microsoft is using UDDI as a core building block of the .NET Platform and plan to integrate support for UDDI into Visual Studio 7, BizTalk Server, Passport, bCentral and the .NET Framework.

Note that UDDI is NOT a technology. It is a comprehensive industry initiative that enables businesses to

- (i) define their business
- (ii) discover other businesses
- (iii) share information about how they interact in a global registry.

It is also an architecture for web services integration:

- (i) It contains standards-based specifications for service description and discovery.
- (ii) It takes advantage of W3C and IETF standards
- (iii) It addresses cross platform programming features by adopting early versions of the proposed SOAP messaging specifications found at the W3C Web site.

