

Microsoft (70-536)
.NET 2.0 Framework
Application Development Foundation

 Smarter
Training

This LearnSmart exam manual will equip you with all the knowledge necessary to successfully complete the Microsoft .NET 2.0 Development exam (70-536). By studying this manual, you will become familiar with an array of exam-related content, including:

- Developing applications that use system types and collections
- Implementing service processes, threading and application domains in a .NET Framework application
- And more!

Give yourself the competitive edge necessary to further your career as an IT professional and purchase this exam manual today!

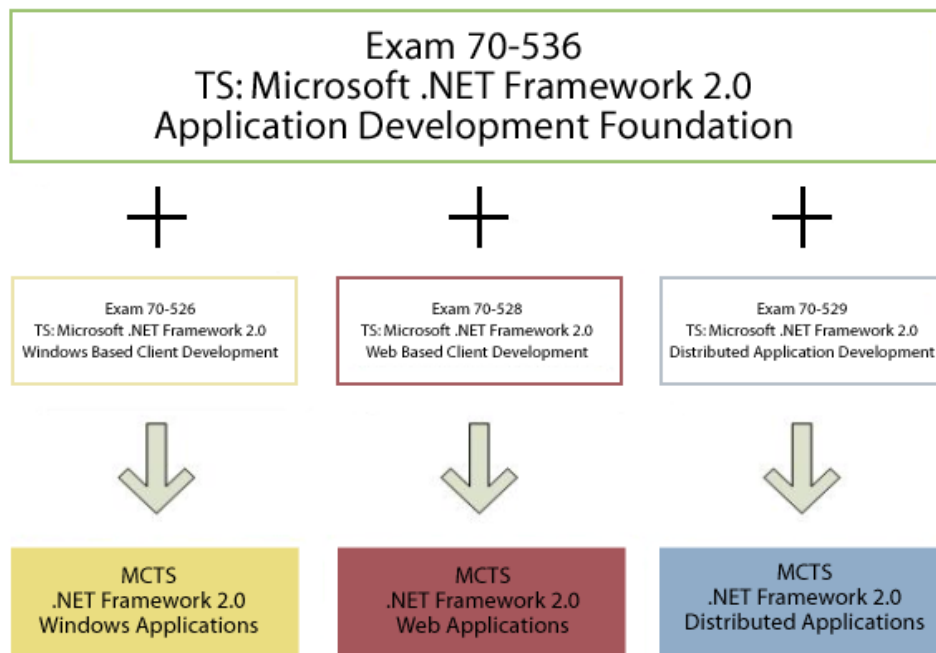
Overview of new features, enhancements, and capabilities of the .NET Framework 2.0

1.1 Microsoft .NET Framework 2.0 Certification Path

This Exam Manual will help prepare you for the certification exam for 70-536:TS: Microsoft .NET Framework 2.0—Application Development Foundation. This exam is one of the examinations required for one of the following three Microsoft Certified Technology Specialist (MCTS) certifications:

- Microsoft Certified Technology Specialist: .NET Framework 2.0 Web Applications
- Microsoft Certified Technology Specialist: .NET Framework 2.0 Windows Applications
- Microsoft Certified Technology Specialist: .NET Framework 2.0 Distributed Applications

Microsoft Technology Specialists are capable of building, implementing, troubleshooting, and debugging applications using Microsoft technologies. The above certifications require that a candidate pass the 70-536 exam and one additional exam specific to each certification path. The certification path for all three options is follows:



MCTS Certification Path Options

Candidates for the 70-536 exam can select the programming language in which all code examples will appear. When the exam begins, they must select from one of the following:

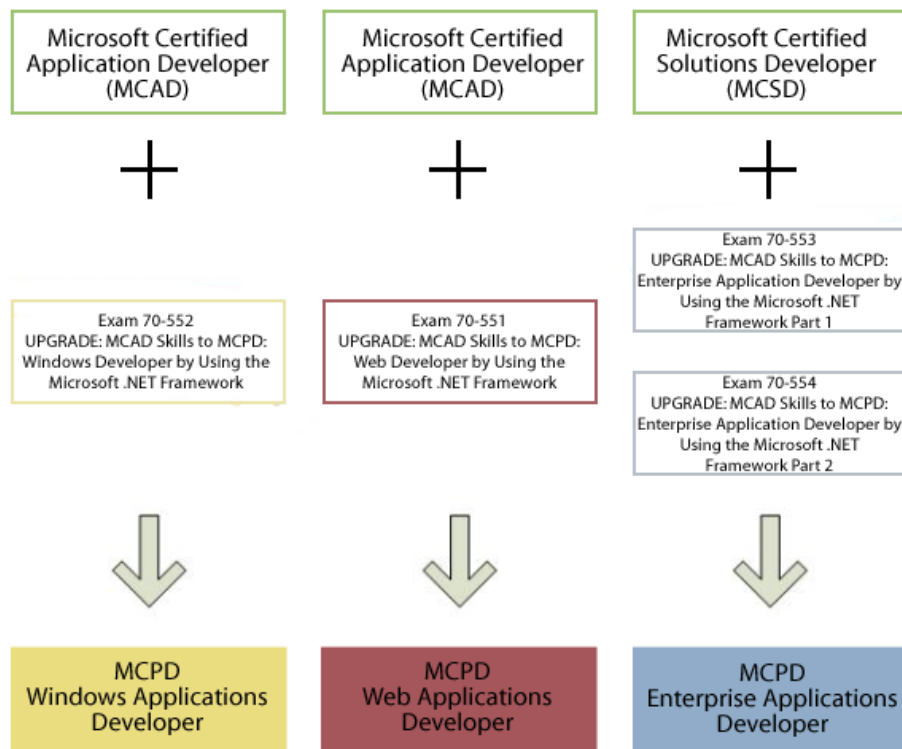
- Microsoft Visual Basic 2005
- Microsoft Visual C# 2005
- Microsoft Visual C++ 2005

The 70-536 exam consists of 150 questions that measure candidates' knowledge of the fundamentals of the .NET Framework 2.0. It covers new features and enhancements in the following areas, specifically, the enhanced capabilities of the namespaces.

After achieving a Microsoft Certified Technology Specialist (MCTS) certification, candidates can enhance their certification level by becoming a Microsoft Certified Professional Developer (MCPD). The MCPD certification is targeted to a candidate's specific technical expertise. The three MCPD certifications are:

- Microsoft Certified Professional Developer: Web Applications Developer
- Microsoft Certified Professional Developer: Windows Applications Developer
- Microsoft Certified Professional Developer: Enterprise Applications Developer

Each of the above certifications requires a candidate to pass one additional exam specific to a certification path (the MCPD: Enterprise Applications Developer requires all three MCTS exams in addition to the MCPD exam). The certification path for all three options is displayed below:



Upgrade Certification Path Options

1.2 Fundamentals of the .NET Framework

This manual assumes that you are familiar with the .NET Framework and have been developing applications (Windows Forms/Windows GUI Applications, ASP.NET Web applications, Mobile applications, Windows Services, Console Applications, XML Web Services, etc.). The Exam 70-536:TS: Microsoft .NET Framework 2.0—Application Development Foundation primarily covers enhancements made to the .NET Framework, specific to version 2.0, but we will briefly review the .NET Framework. For more detailed information on the .NET Framework and information on each of the examination options available, refer to the Introduction.

The .NET Framework is a language-neutral component library and execution environment that provides the infrastructure for building applications using .NET. Application developers can build powerful, enterprise, integrated applications regardless of platform or language. The .NET Framework 2.0 does not introduce a new development framework or paradigm, implementation model, or scripting model from the previous release of .NET Framework 1.0 or .NET Framework 1.1 (the .NET Framework Release 1.0 was initially released in January of 2002). But because it is more than an incremental release, it introduces some significant enhancements to the .NET Framework.

The .NET framework is an object-oriented hierarchy of classes that are ubiquitous to all Microsoft Windows operating systems, regardless of whether the user is developing applications for Windows Forms, Web applications, mobile applications, services, or any other task. When creating a .NET application, a developer creates

a class, and defines the class properties, events, and methods that build the functionality of the application. With .NET, these classes support object-oriented features such as polymorphism, inheritance and encapsulation. For example, when executing an ASP.NET application, the ASP.NET runtime engine will transform the source code the .aspx page into an instance of a .NET framework class that inherits from the *Page* base class. This fundamental concept is why this exam (Exam 70-536:TS:Microsoft .NET Framework 2.0—Application Development Foundation) must be passed in tandem with any of three other exams to earn a Microsoft Certified Technology Specialist certification (with a focus on either .NET Framework 2.0 Web Applications, .NET Framework 2.0 Windows Applications, or .NET Framework 2.0 Distributed Applications). The .NET Framework has two main components:

- The common language runtime (CLR)
- The .NET Framework class library

The common language runtime (CLR) is at the foundation of the .NET Framework. The CLR provides a runtime environment and run-time services for .NET applications and is responsible for code management, access security, language interpretation, memory management, thread management, process management, compilation and code accuracy, strict type and code verification (the common type system of CTS) to ensure that all managed code is self-describing. Code management is an important principle in the .NET Framework. Code targeting the runtime is managed code (custom object libraries, class libraries), while code not targeting the runtime is unmanaged code (Internet Information Services, ASP.NET web applications). The CLR also provides a management environment that automatically provides common services (garbage collection and security). In addition, ASP.NET provides the environment to incorporate both managed and unmanaged features, such as an unmanaged component or application such as Internet Explorer with an embedded managed component or Windows Forms control.

As an application developer or system administrator, you are concerned with productivity, reliability and performance. Because the runtime automatically handles object layout as well as managed references to objects in memory, they are automatically released when no longer being used. This feature helps eliminate memory leaks and invalid object memory references. Finally, the Just In Time compiler (JIT) allows for managed code to run in the system's native machine language. Also, because the CLR supports development in multiple development languages, application developers can script in their language of choice while still accessing a common runtime and class library. Finally, the CLR also provides other fundamental services which are outside the scope of this book.

However, this book will focus on the .NET Framework class library, specifically, on the features, enhancements, and capabilities of the class libraries available in the .NET Framework 2.0. Microsoft defines the .NET Framework as:

"...a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services."

In short, the .NET Framework contains the object-oriented, hierarchical, extensible classes, interfaces, and value types that provide the framework to build .NET applications, components and controls. They provide the standard functionality that allows developers to utilize string manipulation, security management, network communications, thread management, input/output controls and user interface design features. The .NET Framework provides class libraries (APIs) that replace the libraries and frameworks previously used by application developers based upon their programming language (Microsoft Foundation Classes for C++ developers, Visual Basic runtime for VB developers, etc.). The .NET Framework simplifies those issues by providing a common set of APIs for all the supported programming languages (at the time of this writing, more than 25 languages are supported). The .NET Framework supports cross-language inheritance (you can inherit classes across language boundaries), error handling and debugging. A class written in one