**Part 1 – Introduction:**

**Briefly introduce the standard and its history.**

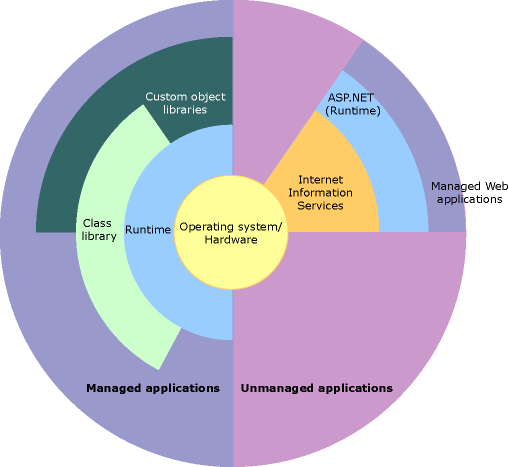
* **For what purpose (domain and/or business context) was the standard created?**
* **Describe the stakeholders: Who created it? Who (domain or community) uses it? How do they use it?**
* **Are there other related and/or competing standards? If so, list them and briefly explain how they differ from the standard you are analyzing.**

**Answer:**

# INTRODUCTION

1. For what purpose (domain and/or business context) was the standard created? (17’)

* The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.



**.NET Framework in context**

* NET is the result of many influences…



* The .NET Framework is an integral Windows component that supports building and running the next generation of applications and XML Web services. The .NET Framework is designed to fulfill the following objectives:
* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

1. Describe the stakeholders: Who created it? Who (domain or community) uses it? How do they use it?

### The .Net Framework for Users

* If you do not develop .NET Framework applications, but you use them, you do not need specific knowledge about the .NET Framework or its operation. For the most part, the .NET Framework is completely transparent to users.
* If you are using the Windows operating system, the .NET Framework may already be installed on your computer. In addition, if you install an application that requires the .NET Framework, the application's setup program might install a specific version of the .NET Framework on your computer. In some cases, you may see a dialog box that asks you to install the .NET Framework. If you have just tried to run an application when this dialog box appears and if your computer has Internet access, you can go to a webpage that lets you install the missing version of the .NET Framework.
* In general, you should not uninstall any versions of the .NET Framework that are installed on your computer, because an application you use may depend on a specific version and may break if that version is removed. Note that multiple versions of the .NET Framework can be loaded on a single computer at the same time. This means that you do not have to uninstall previous versions in order to install a later version.

### [[http://i.msdn.microsoft.com/Areas/Global/Content/clear.gif](javascript:void(0))The .NET Framework for Developers](javascript:void(0))

* The .NET Framework provides the following services for application developers:
* Memory management. In many programming languages, programmers are responsible for allocating and releasing memory and for handling object lifetimes. In .NET Framework applications, the CLR provides these services on behalf of the application.
* A common type system. In traditional programming languages, basic types are defined by the compiler, which complicates cross-language interoperability. In the .NET Framework, basic types are defined by the .NET Framework type system and are common to all languages that target the .NET Framework.
* An extensive class library. Instead of having to write vast amounts of code to handle common low-level programming operations, programmers can use a readily accessible library of types and their members from the .NET Framework Class Library.
* Development frameworks and technologies. The .NET Framework includes libraries for specific areas of application development, such as ASP.NET for web applications, ADO.NET for data access, and Windows Communication Foundation for service-oriented applications.
* Language interoperability. Language compilers that target the .NET Framework emit an intermediate code named Common Intermediate Language (CIL), which, in turn, is compiled at run time by the common language runtime. With this feature, routines written in one language are accessible to other languages, and programmers can focus on creating applications in their preferred language or languages.
* Version compatibility. With rare exceptions, applications that are developed by using a particular version of the .NET Framework can run without modification on a later version.
* Side-by-side execution. The .NET Framework helps resolve version conflicts by allowing multiple versions of the common language runtime to exist on the same computer. This means that multiple versions of applications can also coexist, and that an application can run on the version of the .NET Framework with which it was built.
* Multi-targeting. By targeting the .NET Framework Portable Class Library, developers can create assemblies that work on multiple .NET Framework platforms, such as the .NET Framework, Silverlight, Windows Phone 7, or Xbox 360.
* You can choose any programming language that supports the .NET Framework to create your application. Because the .NET Framework provides language independence and interoperability, you can interact with other .NET Framework applications and components regardless of the language with which they were developed.
* To develop .NET Framework applications or components, do the following:

1. [Install](http://msdn.microsoft.com/en-us/library/5a4x27ek.aspx) the version of the .NET Framework that your application will target. The most recent production version is the .NET Framework 4. The [.NET Framework 4.5 Developer Preview](http://go.microsoft.com/fwlink/?LinkID=227109) is also available for download.
2. Select type of application or service you want to develop. The types of applications and services :

* Console applications. See [Building Console Applications](http://msdn.microsoft.com/en-us/library/f492ca1w.aspx).
* Windows GUI applications (Windows Forms). See [Windows Forms](http://msdn.microsoft.com/en-us/library/dd30h2yb.aspx).
* Windows Presentation Foundation (WPF) applications. See [Introduction to WPF](http://msdn.microsoft.com/en-us/library/aa970268.aspx).
* ASP.NET applications. See [ASP.NET Web Pages](http://msdn.microsoft.com/en-us/library/fddycb06.aspx).
* Web services. See [ASP.NET Web Services](http://msdn.microsoft.com/en-us/library/t745kdsh.aspx).
* Windows services. See [Introduction to Windows Service Applications](http://msdn.microsoft.com/en-us/library/d56de412.aspx).
* Service-oriented applications using Windows Communication Foundation (WCF). See [Windows Communication Foundation](http://msdn.microsoft.com/en-us/library/dd456779.aspx).
* Workflow-enabled applications using Windows Workflow Foundation (WF). See [Windows Workflow Foundation](http://msdn.microsoft.com/en-us/library/dd489441.aspx).

1. Select the .NET Framework language or languages that you will use to develop your applications. A large number of languages are available, including Visual Basic, C#, F#, and C++ from Microsoft. (A programming language that allows you to develop applications for the .NET Framework adheres to the [Common Language Infrastructure (CLI) specification](http://go.microsoft.com/fwlink/?LinkId=199862).) For a list of available programming languages, see [Programming Languages for the .NET Framework](http://go.microsoft.com/fwlink/?LinkId=199863).
2. Select and install the development environment that you will use to create your applications and that supports your selected programming language or languages. The integrated development environment for .NET Framework applications available from Microsoft is [Visual Studio](http://go.microsoft.com/fwlink/?LinkId=199864). It is available in a number of retail and free editions.
3. Are there other related and/or competing standards? If so, list them and briefly explain how they differ from the standard you are analyzing.

|  |  |  |
| --- | --- | --- |
|  | .Net Architecture | Java Architecture |
| Definition | * .NET is Microsoft's platform for XML Web services. XML Web services allow applications to communicate and share data over the Internet, regardless of operating system or programming language | * Java EE is Oracle's enterprise java computing platform. The platform provides an API and runtime environment for developing and running enterprise software, including network and web services, and other large-scale, multi-tiered, scalable, reliable, and secure network applications. |
| Owned | * Microsoft started the development on the .NET Framework in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2000 the first beta versions of .NET 1.0 were released. | * Sun Microsoft, Inc., founded on February 24, 1982 |
| Component | * The **Common Language Runtime (CLR),** an application virtual machine that provides important services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework. * **Base Class Library** provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with the .NET Framework and other libraries. | * A J2EE component is a self-contained functional software unit that is assembled into a J2EE application with its related classes and files and communicates with other components. The J2EE specification defines the following J2EE components: * Application clients and applets are client components. Java Servlet and Java Server Pages technology components are web components. * Enterprise JavaBeans components (enterprise beans) are business components. * Resource adapter components provided by * EIS and tool vendors |
| Architecture | * Designs support multiple difference programming language. Currently, 30 languages support the .net framework. Runs primarily on Microsoft Windows. * The .NET CLR contains an Intermediate Language (IL) engine. Code and objects written in a language can be compiled into the IL runtime, once an IL compiler is developed for the language * CRL implement algorithm continuous distribution * Compiling works faster | * Though other language’s code can be converted run under JVM they don’t acquire true. Cross language capability. Run in multiple platforms. * Compiling source code into the Java "byte code.", as long as platforms have a Java Virtual Machine to execute byte code. * JVM implement algorithm non -continuous distribution |