International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

# Versions of the .Net Framework's Architecture Run From 2.0 to 4.5

**AzraJabeen Mohamed Ali**

Independent Researcher, California, USA

**Abstract:**
This paper discusses .Net framework architecture and its versions. The study's main research question explores the intricate relationship between different versions of .Net Framework architecture and recognized standards in order to comprehend their synergy and impacts on the user experience. From a methodological perspective, this study is based on a detailed analysis of existing industry practices, as well as a comprehensive evaluation of .net frameworks and emerging standards. Through meticulous observation and critical analysis, the study dissects key components of framework architecture, shedding light on the fine line that separates innovation from adherence to established conventions. The study's key findings provide insight into the intricate connection between standard compliance and creative design. The article describes how these factors affect user engagement, website performance, and the overall viability of digital platforms. Designers and developers may navigate the difficult terrain of modern frameworks with the aid of insights gleaned from in-depth research. In addition to theoretical concerns, the study's implications offer practical guidance for business professionals. It's crucial to understand how .Net architecture and standards work together as the version evolves. The findings emphasize how important it is to strike a balance between originality and norm compliance to foster a digital environment where innovation meets user expectations.

**Keywords:** Net Framework, CLR, Microsoft Visual Studio, industry standards, CLR (Common Language Runtime), FCL (Framework Class library).

## INTRODUCTION

### Net Framework Architecture:

The.NET framework is a mainstay in the rapidly evolving field of software development, offering a stable and adaptable platform for creating a vast range of applications. The architecture of the.NET framework is crucial in determining how developers plan and carry out their projects, whether they are desktop software, online applications, or something else entirely. We will examine the main elements, advantages, and effects of the.NET architecture on application development in this research paper.

Microsoft's.NET framework has completely changed how software applications are created and implemented. It gives developers the ability to construct a vast array of apps for Windows environments by offering a comprehensive and unified platform that supports multiple programming languages. The.NET architecture's primary goals are to improve maintainability, productivity, and code reuse.

The.Net framework was initially released in 2002 as version 1.0. To put it simply, it is a virtual machine that can compile and run programs written in many languages, such as C#, VB.Net, and others. Web-based apps, Web services, and Form-based applications are all developed with it. The.Net platform supports a

wide range of programming languages, with C# and VB.Net being the most widely used. It is employed in the development of Windows, phone, web, and other apps. It meets industrial standards and offers a wide range of functionalities. Many versions are released as part of the evolution of architecture like Version 2.0, 3.5, 4.0, 4.5

**Net Framework 2.0:**

The framework's condensed version is called.Net 2.0. The goal behind the release of this version was to increase developer productivity. In this release, new ADO.Net technologies such user-defined types (UDT), XML data types, and asynchronous database operations were added. Additionally, additional ASP.Net capabilities were added, including caching, an improved code-behind paradigm, data controls, and more.

**Fundamental Components in architecture:**

The fundamental components of the architecture are CLR (Common Language Runtime), FCL (.Net Framework Class Library), CLS (Common Language Specification), CTS (Common Type System), CLI (Common Language Infrastructure).

**CLR (Common language Runtime):**

Visual Basic, C#, F#, C++, Cobra, Jscript.Net, VB.Net, Oxygene are the programming languages used to create.NET applications. Code is compiled into a Common Intermediate Language (CIL), which is independent of language. Assemblies, which are files with a.dll or.exe file extension, contain compiled code.

When an application runs, the CLR takes the assembly and converts it into machine code that can operate on the computer architecture by using a just-in-time compiler (JIT). The execution engine responsible for managing active applications is CLR. Among the services it offers are exception handling, type-safety, garbage collection, and thread management.

**FCL (.Net Framework Class Library):**

Developers can speed up application development by utilizing the pre-built classes, methods, and components in the.NET framework Class Library (FCL). From file processing to database connectivity, the FCL provides a wide range of features to manage intricate jobs.

**CTS (Common Type System):**

The.NET framework's base for defining and working with data types is the Common Type System (CTS). It creates a standardized system type that facilitates interoperability and data transmission between languages.

**CLS (Common Language Specification):**

A subset of the Common Type System is the CLS. Programming languages must adhere to a set of rules and principles defined by the Common Language Specification (CLS) in order to be compatible with the .NET environment. It promotes language-agnostic development by guaranteeing that code written in several languages can work together seamlessly.

**CLI (Common Language Infrastructure):**

NET application development and execution are standardized by the Common Language Infrastructure (CLI). Cross-platform compatibility is made possible via the Just-In-Time (JIT) compiler, which converts Intermediate Language (IL) code into native machine code.

Applications are stacked in the architecture on top of the fundamental components.

**Application Types:**

The following three categories mostly include the apps developed using the.Net framework: WinForms,

ADO.Net, ASP.Net. Winforms : Applications that are based on forms fall under this category. This category includes, in essence, client-based apps that have the ability to read and write to the file system. ASP.NET: This category includes web-based apps. Web applications, websites, and web services can be developed with the help of the web framework ASP.Net, which offers a fantastic integration of HTML, CSS, and JavaScript. The addition of web services was made.Net Framework 2.0 and regarded as a component of web applications built using ASP.NET. New ASP.NET customization options, including support for webparts, master pages, themes, and skins. ADO.NET: It comprises programs like Oracle, MS SQL Server, and others that are designed to interact with databases. Classes that can be used connect, retrieve, insert, and remove data make up the majority of it.

**Features of .Net Framework 2.0:**

Released Visual Studio .Net 2005 IDE, Released CLR Version 2.0, Introduced new functionality for ADO.Net and ASP.Net, supported "Access Control List", API for data protection was made available, FTP Support was given, provided programmatic control over caching features. For native programs that want to host a.NET runtime instance, a new hosting API is available: The new API provides fine-grained control over the runtime's memory allocation, assembly loading, and multithreading behavior. It was first created to effectively host the runtime in Microsoft SQL Server, which has its own memory manager and scheduler. Two service packs are released with updates.

**.Net Framework 3.0:**

On November 6, 2006,.NET Framework 3.0—previously known as WinFX—was made available. It comes with a fresh set of managed code APIs that are essential to Windows Server 2008 and Vista. It can also be downloaded for Windows Server 2003 and Windows XP SP2. Since.NET Framework 3.0 uses the same CLR as.NET Framework 2.0, there aren't any significant architectural changes included with this release.[51] There was no.NET Compact Framework release to go along with this version, in contrast to the earlier major.NET releases. Windows Vista came with the.NET Framework version 3.0. Additionally, it came with Windows Server 2008 as an add-on (by default disabled). On top of .Net framework 2.0, there comes the new features of .Net framework 3.0 like WPF (Windows Presentation Foundation), WCF (Windows Communication Foundation), WF (Windows Workflow Foundation), WCS (Windows Card Space).

**WPF (Windows Presentation Foundation):**

Microsoft's Windows Presentation Foundation (WPF) is a graphical subsystem that renders user interfaces (UI) in Windows-based applications using DirectX. WPF, formerly known as "Avalon," was first made available in 2006 as a component of.NET Framework 3.0.

**WCF (Windows Communication Foundation):**

Previously known as Indigo, Windows Communication Foundation (WCF) is a service-oriented messaging framework that enables programs to work together locally or remotely in a manner akin to web services. A framework called Windows Communication Foundation (WCF) is used to create service-oriented applications. Data can be sent as asynchronous messages between service endpoints using WCF. A service endpoint may be a service hosted within an application or it may be a component of an IIS-hosted continuously accessible service. A client of a service that makes a data request to a service endpoint might be considered an endpoint. The messages can be sent as a stream of binary data or as basic as a single letter or word in XML format.

**WF (WorkFlow Foundation):**

Using workflows, Windows Workflow Foundation (WF) enables the creation of integrated transactions

and task automation.

**WCS (Windows Card Space):**

Previously known as InfoCard, Windows CardSpace is a software component that safely saves an individual's digital identities and offers a single interface for selecting the identity for a specific operation, like entering into a website.

**Features of .Net Framework 3.0:**

Windows XP, Server 2003, Vista, Server 2008, and Server 2008 R2 all support .NET Framework 3.0.WPF, WCF, WF, WCS are the new features part of version 3.0. Two service packs are released with updates.

**.Net Framework 3.5:**

Version 3.5 of the .NET Framework was published on 19 November 2007. Version 3.5 utilizes the Common Language Runtime (CLR) 2.0, which is the same version as.NET Framework version 2.0, just like.NET Framework 3.0. Additionally,.NET Framework 3.5 installs.NET Framework 2.0 SP1 and 3.0 SP1 (using the later 3.5 SP1 instead of 2.0 SP2 and 3.0 SP2). This enables the BCL classes in version 2.0 to have certain methods and properties that are necessary for features like Language Integrated Query (LINQ) in version 3.5. Version 2.0 programs are unaffected by these modifications. On top of .Net framework 3.0, there comes the new features of LINQ (Language Integrated Query), Entity Framework.

**LINQ (Language Integrated Query):**

A group of solutions based on the direct integration of query capabilities into the C# language are referred to as Language-Integrated Query (LINQ). Conventionally, searches against data are written as straightforward strings that lack IntelliSense assistance and type verification at compile time. Additionally, each type of data source—such as SQL databases, XML documents, different Web services, and so forth—requires a different query language. Like classes, methods, and events, a query is a first-class language construct in LINQ.

**Entity Framework:**

A collection of ADO.NET technologies called the C# Entity Framework facilitates the creation of data-driven software. The Object Relational Mapping (ORM) framework C# Entity provides developers with an automated method for storing and retrieving databases. Developers may work with data at a greater level of abstraction thanks to the Entity Framework. Compared to traditional applications, Entity Framework enables you to write less code for data-oriented programs.

**Features of .Net Framework 3.5:**

LINQ, Entity Framework are the new features. Improved assistance with cryptography procedures, Better support for AJAX programming in ASP.NET, Offered the ability to network with peers, LatencyModes are supported during trash collection, facilitated the interoperability of web services, Improved performance of the thread pool, Released Visual Studio .Net 2008 IDE. Two service packs are released, which System.Web.Abstraction and System.Web.Routing are two new web development assemblies introduced that are utilized in the ASP.NET MVC framework.

**.Net Framework 4.0:**

The final release of Microsoft Visual Studio 2010 coincided with the introduction of the.NET Framework 4.0. Vista (with Service Pack 1 or higher), Windows XP (with Service Pack 3), Windows Server 2003 (with Service Pack 2), Server 2008, 7, and Server 2008 R2 all support.NET Framework 4.0. As part of new features PLINQ (Parallel LINQ) and Task Parallel Library (TPL) are introduced.

**PLINQ (Parallel LINQ):**

The Language-Integrated Query (LINQ) paradigm is implemented in parallel-by-Parallel LINQ (PLINQ). The complete set of LINQ standard query operators is implemented by PLINQ as System extension methods. It has extra operators for parallel processing and uses the System.Linq namespace. PLINQ combines the capability of parallel programming with the readability and simplicity of LINQ syntax. There are many similarities between a PLINQ query and a non-parallel LINQ to Objects query. PLINQ queries feature deferred execution, meaning they don't start running until the query is enumerated. They work on any in-memory IEnumerable or IEnumerable data source, exactly as sequential LINQ queries. The main distinction is that PLINQ aims to utilize every processor in the system to its maximum potential. To do this, the data source is divided into pieces, and the query is then run on each segment using different worker threads running concurrently on several processors. Parallel execution frequently results in much faster query execution.

**Task Parallel Library (TPL):**

The Task Parallel Library (TPL) is a collection of public types and APIs in System.Threading and System.Threading.Tasks namespaces. The TPL aims to increase developer productivity by streamlining the process of integrating concurrency and parallelism into applications. To make the best use of all the available processors, the TPL dynamically scales the level of concurrency. TPL also manages state management, cancellation support, thread scheduling on the ThreadPool, job partitioning, and other low-level tasks. With TPL, you may optimize your code's performance while concentrating on the tasks your program is intended to complete. The TPL is the recommended method for writing parallel and multithreaded programming in the.NET Framework 4. But not every piece of code can be parallelized. For instance, if a loop doesn't run for many iterations or only does a little amount of work on each iteration, the overhead of parallelization may result in slower code execution. Additionally, parallelization complicates the execution of your application, just like any multithreaded code. In order to utilize the TPL efficiently, even if it simplifies multithreaded scenarios, we advise that you have a fundamental understanding of threading principles, such as locks, deadlocks, and race conditions.

**Features of .Net Framework 4.0:**

Supports parallel computing using Task Parallel library, PLINQ, offers language services for dynamic languages with its "Dynamic Language Runtime," which operates atop CLR, Assistance with Code Contracts, New types are added to support complex numbers (System.Numerics.Complex) and arbitrary-precision arithmetic (System.Numerics.BigInteger), Released Common Language Runtime (CLR) 4.0, Introduced new features for the Visual Basic,.NET, and C# languages, including named parameters, dynamic dispatch, implicit line continuations, and optional parameters, Released Visual Studio .Net 2010 IDE.

**.Net Framework 4.5:**

On August 15, 2012, NET Framework 4.5 was made available; it included a number of enhanced or new capabilities. Windows Vista or later supports NET Framework 4.5. Common Language Runtime 4.0 is used by the.NET Framework 4.5, along with a few other runtime features. .NET Framework 4.5 is supported on Windows Vista (with Service Pack 2), Server 2008 (with Service Pack 2), 7 (with Service Pack 1), Server 2008 R2 (with Service Pack 1), 8, Server 2012, 8.1 and Server 2012 R2. There are two subsets of the.NET Framework that can be used to create Metro-style applications in Visual Basic or C#: .NET APIs for Windows 8.x Store apps is one for Windows 8 and Windows 8.1. .NET APIs for UWP is another for the Universal Windows Platform (UWP).

**.NET API For Store/UWP Apps:**

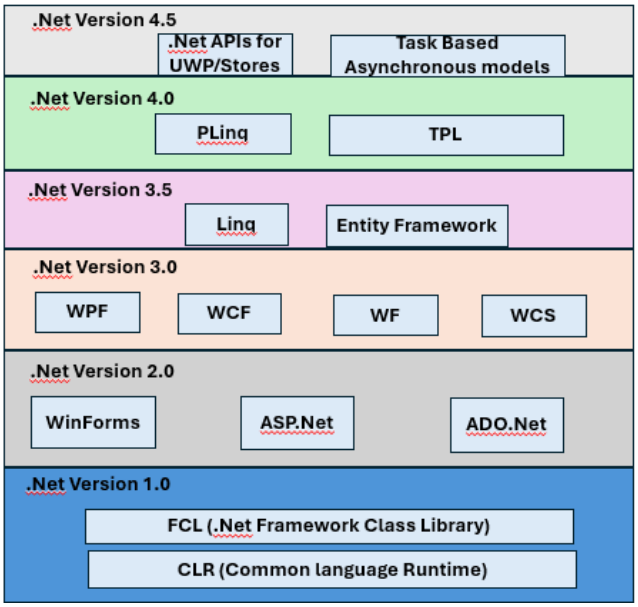
Microsoft introduced a few APIs in 2012 that allow developers to use C# or VB to create UWP (Universal Windows Platform) applications for Windows. Asp.Net Web API is an open-source framework for web services that makes it easier to create and use RESTful services.

**Task-Based Asynchronous Model:**

In.NET, For new development, the task-based asynchronous pattern is the suggested asynchronous design pattern. It is predicated on the Task and Task<TResult> types in System.Threading.Task namespace to carry on asynchronous operations. TAP represents the start and finish of an asynchronous action using a single method. This stands in contrast to the Event-based Asynchronous Pattern (EAP) and the Asynchronous Programming Model (APM or IAsyncResult) pattern.

**Features of .Net Framework 4.5:**

The ASP.NET Web API was released as part of the .NET Framework 4.5, along with many other new features and tools. The C# and Visual Basic languages now have new asynchronous functionality. These features implement futures and promises and add a task-based mechanism for asynchronous operations. Zip compression is implemented natively; earlier iterations only supported the compression technique, not the archive format. Versioning of comparative data and cultural string ordering is supported. By using the CustomReflectionContext class, it is possible to modify a reflection context and override the default reflection behavior. Improved resource retrieval performance. Released Visual Studio .Net 2012 IDE. The new platform and development model for Metro-style apps, Windows Runtime, includes this version of the.NET Framework together with the runtime and libraries needed for Metro-style apps. It is an ecosystem that contains a variety of platforms and languages, such as HTML5 with JavaScript, C++, and the.NET Framework.




**Conclusion**

.Net Framework offers several advantages which are listed below. The object-oriented programming (OOP) paradigm is the foundation of the.NET framework. By encouraging code modularity, reusability, and maintainability, this method helps developers produce scalable and reliable programs. A rich framework with many functionalities is offered by the.NET architecture. These characteristics include

secure code execution, strong exception handling, and automatic memory management via garbage collection. To speed up development, developers can take advantage of the comprehensive.NET framework Class Library. Writing repeated code is less necessary when pre-built classes and components are available, which saves crucial development time. Since source code and HTML are centralized, pages created with.NET are naturally maintainable. As a result, the entire software lifecycle is improved and development, maintenance, and debugging become simpler. The CLR efficiently manages memory and resources while guaranteeing consistency in code execution. Additionally, the runtime environment keeps an eye out for irregularities in applications, automatically resolving problems like memory leaks and excessive resource utilization.

Applications can be complicated to deploy, but the.NET framework makes it easier. Framework-based applications can be packed with all required dependencies, simplifying deployment. Furthermore, the framework guarantees version compatibility, enabling applications to function flawlessly across various.NET runtime versions.

**References**

1. Microsoft "Get started with.Net framework" https://learn.microsoft.com/en-us/dotnet/framework/get-started/?WT.mc_id=dotnet-35129-website#Introducing (Sep 21, 2022 )
2. Microsoft "Task-based asynchronous pattern (TAP) in .NET: Introduction and overview" https://learn.microsoft.com/en-us/dotnet/standard/asynchronous-programming-patterns/task-based-asynchronous-pattern-tap (Mar 11,2022)
3. Microsoft "Task Parallel Library (TPL)" https://learn.microsoft.com/en-us/dotnet/standard/parallel-programming/task-parallel-library-tpl (Oct 04, 2022)
4. Microsoft "Introduction to PLINQ" https://learn.microsoft.com/en-us/dotnet/standard/parallel-programming/introduction-to-plinq (Sep 15, 2021)
5. Microsoft ".NET Framework Programming" https://learn.microsoft.com/en-us/previous-versions/dotnet/netframework-2.0/ms229284(v=vs.80) (Dec 04, 2006)
6. Microsoft ".Net Framework 3.0" https://learn.microsoft.com/en-us/previous-versions/dotnet/netframework-3.0/aa338210 (Nov 03, 2006)
7. Microsoft ".Net Framework 3.5" https://learn.microsoft.com/en-us/previous-versions/dotnet/netframework-3.5/w0x726c2(v=vs.90) (Nov 16, 2012)
8. Microsoft "Windows Presentation foundation" https://learn.microsoft.com/en-us/previous-versions/dotnet/netframework-3.5/ms754130(v=vs.90) (Nov 16, 2012)
9. Microsoft "Windows Communication foundation" https://learn.microsoft.com/en-us/previous-versions/dotnet/netframework-3.5/ms735119(v=vs.90) (Mar 22, 2010)
10. Microsoft "Windows Workflow Foundation" https://learn.microsoft.com/en-us/previous-versions/dotnet/netframework-3.5/ms735967(v=vs.90) (Mar 03, 2010)
11. Microsoft ".Net framework 4.0" https://learn.microsoft.com/en-us/previous-versions/dotnet/netframework-4.0/w0x726c2(v=vs.100) (Oct 22, 2012)