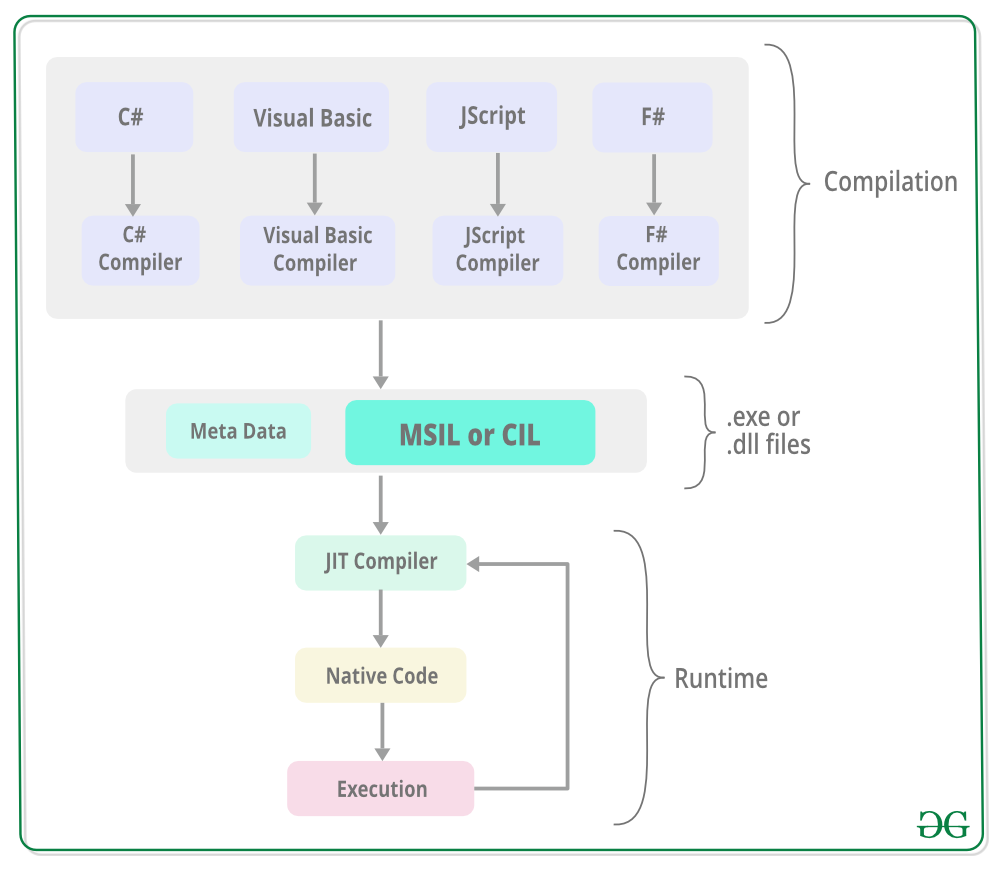
**.NET GLOSSARY DOCUMENTATION**

1. **WPF(Window Presentation Foundation)**
   1. WPF is a subset of .Net type located in **System.Windows** namespace.
   2. WPF uses XAML of what GUI should look like for frontend and C# language for backend programming(Similar to Win Form).
   3. WPF used to build window client application only runs on window OS.
2. **WCF(Window Communication Foundation)**
   1. It is a framework for building service-oriented application. Using WCF data can be send asynchronously from one endpoint to another.
   2. Service endpoint can be part of continuously available service hosted by IIS or it can be service hosted by application.
3. **AOT (Ahead of Time Compiler)**
   1. Compiler translates IL(Intermediate Language) to machine code.
   2. AOT compilation happens before the application is executed and is usually performed on the different machine.
   3. AOT compiler spends more time on optimizing.
   4. Performs cross-module linking and whole program analysis.
4. **ASP.NET**
   1. ASP.NET is a free web framework for building great websites and web applications using HTML,CSS and JavaScript.
   2. It is a server-side scripting language.
   3. ASP.NET is a cross-platform that runs on Window, MacOS, Linux and Docker.
   4. When a browser requests an ASP.Net file, the ASP engine reads the file, and returns the result to the browser.
5. **ASP.NET CORE**
   1. ASP.NET Core is a cross-platform,high performance open source framework for building modern,cloud based,internet connected applications.
   2. It was designed to provide an optimized development framework for apps that are deployed to the cloud or run on- premises.
6. **ASSEMBLY**
   1. An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality.
   2. Assemblies can take the form of .exe(executable file) or .dll(dynamic link library).
   3. Assemblies can be static or dynamic.
7. **CLR(Common Language Runtime)**
   1. CLR handles memory allocation and management.
   2. It converts the code to the native code which can be further executed by the CPU.
   3. CLR handles exceptions, provides type-safety, provides security, memory management, improve performance, language independent, platform independent, garbage collection, etc.
8. **CoreCLR(.Net Core Common Language Runtime)**
   1. CoreCLR is the .Net execution engine in .Net Core performing functions such as garbage collection and compilation to machine code.
   2. Core CLR is a part of .Net Core and represents a simplified version of CLR.
   3. It is a cross-platform runtime.
9. **CoreFX**
   1. CoreFX is the foundational class libraries for .Net Core.
   2. It includes types for collections, console, JSON, XML, file systems, async and many others.
10. **CoreRT(.Net Core Runtime)**
    1. It doesn't include a JIT.
    2. It includes GC(Garbage Collector) and the ability for runtime type identification(RTTI) and reflection.
11. **Cross-Platform**
    1. The ability to develop and execute an application that can be used on multiple different operating systems such as Linux, Windows and Mac etc.
12. **Garbage Collector(GC)**
    1. The garbage collector (GC) manages the allocation and release of memory.
    2. Serves as an automatic memory manager.
13. **Intermediate Language(IL)**
    1. It is used to convert high level .Net language into low level instruction set.
    2. Also known as MSIL(Microsoft Intermediate Language) or CIL(Common Intermediate Language). It is platform independent.
14. **Just-in-Time(JIT compiler)**
    1. It is similar to AOT this compiler translate the IL to machine code that processor understand.
    2. It is responsible for performance optimization at runtime
    3. At runtime it interact with CLR to convert code into native machine code.



1. **Library**
   1. It is collection of Api that can be called by apps or other libraries
   2. A .Net library is composed of one or more assemblies
2. **MetaPackage**
   1. A metapackage contain link to existing package. They keep package as a dependencies.
   2. A NuGet package that has only dependencies, do not contain library of its own.
   3. It is known as virtual package which are loosely grouped together.
3. **.Net Core**
   1. A cross platform(support all system), high- performance open source implementation of .NET.
   2. It includes CoreCLR, CoreAOT, CoreRT, CoreSDK, Core Basic Class Library
4. **.Net Core CLI**
   1. It is a new cross-platform tool chain for creating, restoring package, building running and publishing .Net Application.
   2. It is like Command prompt installed with .Net Core SDK. It accepts command from used end and then process the same.
   3. The CLI has many commands. Each command can be followed by arguments and options.
5. **.Net Core SDK** 
   1. It is a set of libraries and tools that allow developer to create core application
   2. It contain .Net core CLI for building apps, class libraries and dotnet executable that runs application and CLI commands.
6. **.Net Framework**
   1. .Net framework provide the programming environment where software can be developed, installed and executed.
   2. It provides easy to use data structure and API. Many programming languages is integrated in one framework.
7. **Package**
   1. It is a .zip file with one or more assemblies of the same name along with additional metadata
   2. The .zip file has extension .nupkg and contain assets such as .xml files, etc.
   3. The assets define implementation of class library.
8. **Stack**
   1. It contains set of programming technologies that used together to build and run application.
   2. The .Net stack refer to .Net standard and all implementation of .Net.
9. **Target Framework**
   1. The application uses only those functionalities that is available in the specified framework version only.
   2. For example, .Net standard 2.0
10. **TFM(Target Framework Moniker)**
    1. A standard token format for specifying target framework for .Net app or library
    2. The TFM can be short name or long name Example: .Net Framework, Version = 5.6.2
11. **UWP(Universal Window Platform)**
    1. Implementation of an application can be used in different type of devices like phone, tablet, pc, etc.
    2. UMP provide different service such as centralized app, execution environment.
    3. App can be written in C++,C#, VB.NET, JavaScript.
12. **IIS(Internet Information Services)**
    1. It is used to host and provide internet-based services to asp.net and asp web application
    2. IIS support HTTP,HTTPS,FTP,SMTP and many more protocol.

