

Getting Started with the .NET Framework

.NET Framework 4.5 255 out of 325 rated this helpful

The .NET Framework is a runtime execution environment that manages applications that target the .NET Framework. It consists of the common language runtime, which provides memory management and other system services, and an extensive class library, which enables programmers to take advantage of robust, reliable code for all major areas of application development.

What is the .NET Framework?

The .NET Framework is a managed execution environment that provides a variety of services to its running applications. It consists of two major components: the common language runtime (CLR), which is the execution engine that handles running applications; and the .NET Framework Class Library, which provides a library of tested, reusable code that developers can call from their own applications. The services that the .NET Framework provides to running applications include the following:

- **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.
- **Multitargeting.** By targeting the .NET Framework Portable Class Library, developers can create

assemblies that work on multiple .NET Framework platforms, such as Windows 7, Windows 8, Windows 8.1, Windows Phone, and Xbox 360.

The .NET Framework for users

If you do not develop .NET Framework applications, but you use them, you are not required to have any 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. There are two reasons for this:

- If an application that you use depends on a specific version of the .NET Framework, that application may break if that version is removed.
- Some versions of the .NET Framework are in-place updates to earlier versions. For example, the .NET Framework 3.5 is an in-place update to version 2.0, and the .NET Framework 4.5.2 is an in-place update to versions 4, 4.5, and 4.5.1. For more information, see [.NET Framework Versions and Dependencies](#).

If you do choose to remove the .NET Framework, always use **Programs and Features** from Control Panel to uninstall it. Never remove a version of the .NET Framework manually.

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.

The .NET Framework for developers

If you are a developer, 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 the version of the .NET Framework that your application will target. The most recent production version is the [.NET Framework 4.5.2](#). There are additional .NET Framework packages that are released out of band. For information about these packages, see [The .NET Framework and Out-of-Band Releases](#).
2. Select the .NET Framework language or languages that you will use to develop your applications. A number of languages are available, including Visual Basic, C#, Visual 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](#).) For a list of available programming languages, see [Visual Studio Languages](#).

- 3. Select and install the development environment that you will use to create your applications and that supports your selected programming language or languages. The Microsoft integrated development environment for .NET Framework applications is [Visual Studio](#). It is available in a number of retail and free editions.

Related Topics

Title	Description
Overview of the .NET Framework	Provides detailed information for developers who build applications that target the .NET Framework.
The .NET Framework and Out-of-Band Releases	Describes the .NET Framework out-of-band releases and how to use them in your app.
.NET Framework System Requirements	Lists the hardware and software requirements for running the .NET Framework.
Roadmap for the .NET Framework	Lists resources for understanding, evaluating, and deploying the .NET Framework, and for developing apps that target the .NET Framework.
Quick Technology Finder for the .NET Framework	Provides links to individual technologies that are included in the .NET Framework.
Installing the .NET Framework 4.5, 4.5.1, and 4.5.2	Provides information about installing the .NET Framework.

See Also

- Concepts
 - [What's New in the .NET Framework 4.5, 4.5.1, and 4.5.2](#)
- Other Resources
 - [.NET Framework 4.5](#)
 - [.NET Framework Class Library](#)
 - [.NET Framework Development Guide](#)