What is the Microsoft .NET Framework, and why is it installed on the PC?

If you are interested in knowing everything, let's discover what .NET is and why so many applications need it in the following article.

If you have been using Windows for a long time, you may have heard about Microsoft .NET, possibly because an application requires you to install it, or you see it in your list of installed programs. me Unless you are a developer, you don't need much knowledge to use it, as long as it works. But if you are interested in knowing everything, let's explore what .NET is and why so many applications need it in the following article.

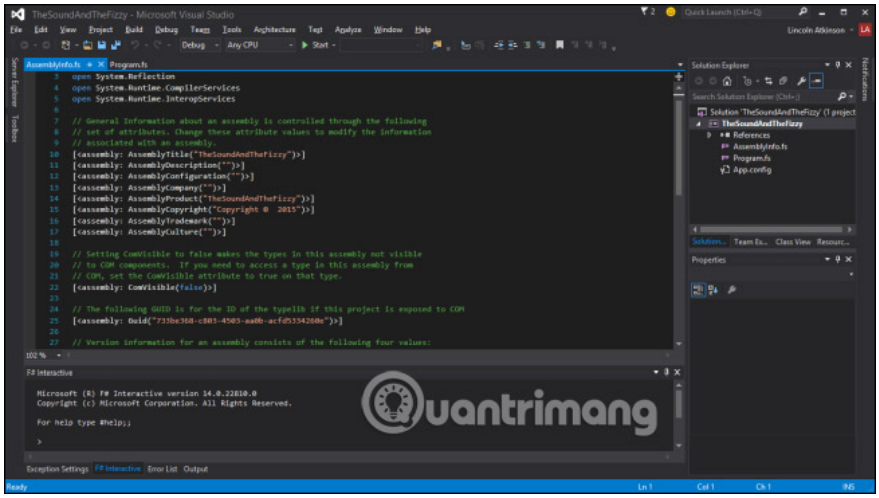
What is the .NET Framework?

The name '.NET Framework' may cause a bit of confusion. A framework (in programming) is actually a set of **Application Programming Interfaces (APIs)** and a shared code library, which developers can use when developing applications. so don't have to write code from scratch. In the .NET Framework, the shared code library is named Framework Class Library (FCL). The code bits in the shared library can perform all the different functions. For example, a developer wants their application to query another IP address on the network. Instead of writing the code yourself, and then writing all the bits and sections to explain the meaning of the query results, they can use the code from the library to perform that function.

That's just a small example. .NET Framework contains tens of thousands of code. These code sections make developer work much easier, because they don't have to write code from scratch for some common functionality on the application. Instead, they can focus on writing their own application code and for the user interface, which binds everything together. Using a code sharing framework like this also helps to set some standards between applications. Other developers can understand what a program is doing more easily and application users can see things like the **Open and Save As** dialog boxes that work the same in different applications.

So, why does the framework name cause confusion?

Because in addition to function as a framework of shared code, .NET also provides an environment for running applications. The running environment provides a sandbox just like a virtual machine, where the application will run. Many development platforms provide the same thing. For example, Java and Ruby on Rails both provide their own application runtime environment. In the .NET world, the application runtime environment is named Common Language Runtime (CLR). When a user runs an application, the code for that application is actually compiled into machine language at runtime and then executed. CLR also provides a number of other services, such as memory management and processors, handling program exceptions and security management. The application running environment is really the way to get applications from the actual hardware that the application is running.



There are several advantages when applications work within an application-specific environment. The biggest plus is portable. Developers can write code using any supported language, including popular programming languages ??like C #, C ++, F #, Visual Basic and several dozen other languages. The code can then be run on any hardware that .NET is supported. Although this platform is designed to support hardware, not Windows-based computers. However, its monopoly makes it virtually used for Windows applications.

Microsoft has deployed .NET in many ways to help solve this problem. Mono is a free and open source project, designed to provide compatibility between .NET applications and other platforms, especially Linux. .NET Core deployment, also a free and open source framework, is designed to bring lightweight modular applications to multiple platforms. .NET Core is designed to support Mac OS X, Linux and Windows (including support for popular Windows platform applications).

As you can imagine, a framework like .NET can bring real benefits, on the development side of everything. It allows developers to write code in their preferred language and ensures that the code can run wherever the framework is supported. Users benefit from consistent applications and the fact that many applications may not be developed, if developers do not have access to the framework.

How .NET is installed on the system?

The .NET Framework has a somewhat complicated history, and it has had many versions over the years. Typically, the latest .NET version will be included in each new release of Windows. .NET versions have backward compatibility (so an application written for version 2 can run on version 3), but it doesn't work as well as on the previous version. Not all applications work with newer versions. In particular, on systems running Windows XP and Vista, you often see many different versions of .NET installed on your PC.

Basically, there are three ways that any specific .NET Framework version will be installed:

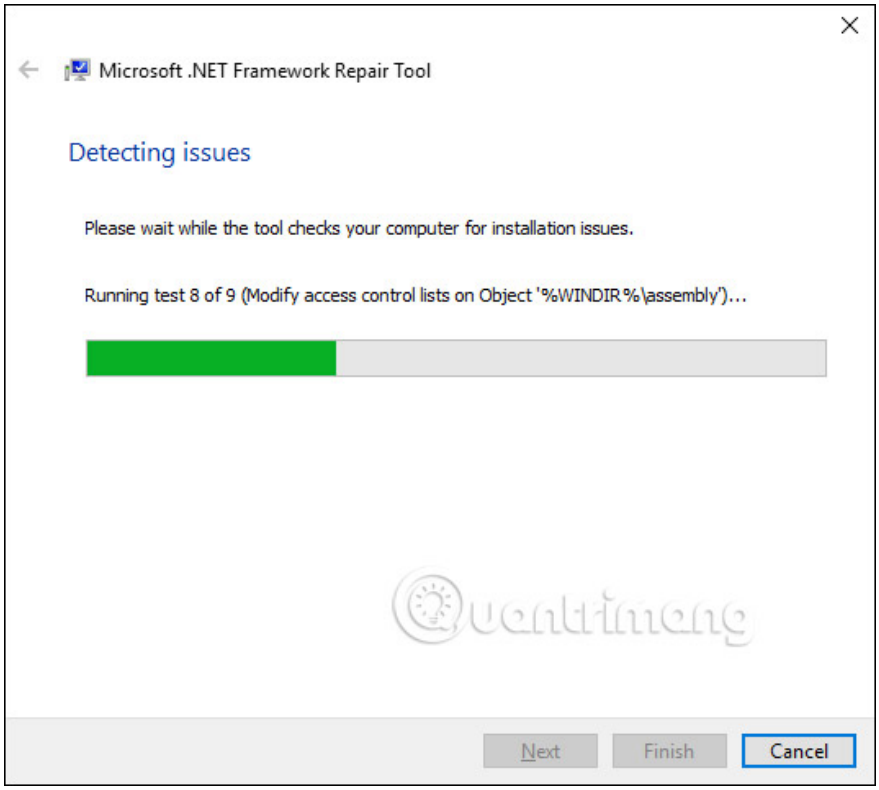
1. Your version of Windows may already have .NET Framework in its default settings.
2. An application requires a specific version, the .NET Framework can be installed during the installation of that application.
3. Some applications will even redirect you to a separate download site to get and install a specific version of the .NET Framework.

Fortunately, things become easier in modern Windows versions. In Windows Vista's 'heyday', two important things happened. First, the .NET Framework 3.5 is released. That version has been redesigned, including components from versions 2 and 3. Applications that require previous versions will still work if you install version 3.5. Secondly, upgrading to the .NET Framework has finally begun to be distributed via Windows Update.

Taken together, these two means that developers can now rely heavily on users - objects with the right components pre-installed and no longer need to ask users to perform the installation. put additional.

When Windows 8 was released, there was a redesigned .NET Framework 4 version that completely came with it. Version 4 (and later versions) does not have backward compatibility with older versions. It is designed to run parallel with version 3.5 on the same PC. Applications written from version 3.5 or later will require installation of version 3.5 and applications written for version 4 or higher will need to install version 4. Good news is that users do not really have to worry about those settings too. Windows will handle it all for you.

Windows 8 and Windows 10 include versions 3.5 and 4 (the latest version is currently 4.6.1). When installing an application, it will need one of those versions and Windows will automatically add the .NET Framework. You can manually add them to Windows by accessing the optional features of Windows. You have the option to add version 3.5 and version 4.6 separately.



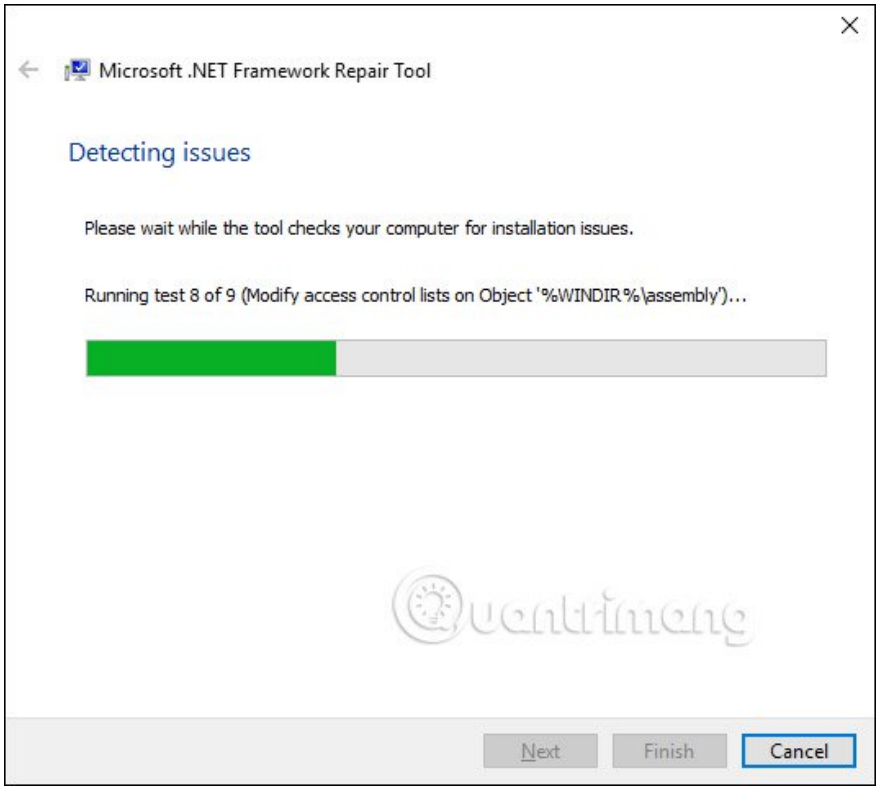
That means, there is no real reason to add them yourself to your Windows installation unless you are developing applications. The first time you install an application will need one of the available versions and Windows will automatically add it to you.

What to do if you have problems with .NET?

You may not have problems with .NET on current Windows versions. Since both required versions are in Windows and installed when needed, the application installation is quite seamless. On older Windows versions (like XP and Vista), you often have to uninstall and reinstall different versions of .NET. You also have to go through checks to make sure that the appropriate .NET versions are installed for the applications that need them. Now, Windows will handle those things for you.

That is, if you have a problem and you think it is related to the .NET framework, you can do a few steps here.

First, you should ensure that Windows has all the latest updates. If there is an update of the .NET Framework, that might be the way to solve your problem. You can also try to remove the .NET Framework versions from your computer and then add them again. If both steps do not work, you can try scanning for corrupted system files in Windows. This process does not take a long time and can restore corrupted or lost system files.



If the above methods don't work, try downloading and running Microsoft's .NET Framework Repair tool. This tool supports all current versions of the .NET Framework. It helps you troubleshoot common problems with setting up or updating to new .NET and can automatically fix any problems you encounter.

The above is all the information about the .NET Framework we want to mention in this article. Hope they will be useful for you !.

See more:

1. Enable .Net Framework 3.5 on Windows 8
2. Fix error 0x800F081F when installing .Net Framework 3.5
3. Fix the error of not installing the .NET Framework 3.5 on Windows

You finished reading the article "**What is the Microsoft .NET Framework, and why is it installed on the PC?** " edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.