

# Getting Started with the .NET Framework

## .NET Framework (current version)

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 10, Windows Phone, and Xbox 360.

## The .NET Framework for users