

## 03 Setting up the development environment (e.g., Visual Studio)

Setting up a development environment for .NET Core typically involves installing the necessary tools and configuring your preferred Integrated Development Environment (IDE). Here's a step-by-step guide to setting up the development environment with .NET Core, using Visual Studio as the IDE:

### Prerequisites:

1. **Operating System:** Ensure that your operating system (Windows, macOS, or Linux) meets the requirements for running .NET Core. .NET Core is cross-platform and can run on Windows, macOS, and various distributions of Linux.
2. **.NET Core SDK:** Download and install the .NET Core SDK from the official .NET website (<https://dotnet.microsoft.com/download>). The SDK includes the necessary tools for building and running .NET Core applications.

### Setting Up Visual Studio:

1. **Download Visual Studio:** Go to the Visual Studio website (<https://visualstudio.microsoft.com/>) and download the version of Visual Studio that best suits your needs. Visual Studio Community is free and offers a comprehensive set of features for individual developers and small teams.
2. **Install Visual Studio:** Run the downloaded installer and follow the on-screen instructions to install Visual Studio on your machine. During installation, you can customize the components and workloads to include .NET Core development tools.
3. **Launch Visual Studio:** Once installed, launch Visual Studio from the Start menu or desktop shortcut.
4. **Select Workload:** In the Visual Studio installer or when you first launch Visual Studio, select the ".NET Core cross-platform development" workload. This ensures that the necessary components and tools for .NET Core development are installed.
5. **Create or Open a Project:** You can create a new .NET Core project by selecting "File" > "New" > "Project" and choosing the desired project template (e.g., Console Application, Web Application). Alternatively, you can open an existing .NET Core project by selecting "File" > "Open" > "Project/Solution".
6. **Configure Target Framework:** In the project properties or project file ( `*.csproj` ), ensure that the target framework is set to a compatible version of .NET Core. You can specify the target framework version based on your project requirements.

### Additional Tools and Extensions:

- **Visual Studio Code:** Alternatively, you can use Visual Studio Code, a lightweight and versatile code editor, for .NET Core development. Install the C# extension for Visual Studio Code to add support for C# development (<https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.csharp>).
- **NuGet Package Manager:** Visual Studio includes NuGet Package Manager for managing dependencies in your .NET Core projects. You can use NuGet Package Manager to search for, install, and update packages from the NuGet Gallery.
- **Azure DevOps:** If you're working in a team or require version control and project management capabilities, consider using Azure DevOps for source control, continuous integration/continuous deployment (CI/CD), and agile planning.

Once you've set up your development environment, you can start building and running .NET Core applications using Visual Studio. Make sure to regularly update your tools and extensions to access the latest features and improvements in .NET Core development.