

What is a Namespace in C#?

In C#, a **namespace** is a way to organize and group related classes, interfaces, structs, enums, and delegates into a logical structure. Namespaces help manage the complexity of large software projects by avoiding name collisions and providing a hierarchical organization for code.

Why Use Namespaces?

1. **Organization:** Namespaces help to logically group related types (e.g., classes, interfaces) together, making the codebase easier to manage and navigate.
2. **Avoiding Name Collisions:** They prevent naming conflicts by allowing you to use the same class names in different namespaces.
3. **Code Readability:** Namespaces improve the readability of the code by providing context for the types used.

Syntax

1. Defining a Namespace:

```
namespace NamespaceName
{
    // Class, interface, or other types
}
```

2. Using a Namespace:

- To use types from a namespace, you can use the using directive at the beginning of your file:

```
using NamespaceName;
```

Example of Using Namespaces

Here's a simple example to illustrate the concept of namespaces:

File 1: Library.cs

```
// Define a namespace called MyLibrary
namespace MyLibrary
{
    public class Book
    {
        public string Title { get; set; }
        public string Author { get; set; }

        public void DisplayInfo()
        {
            Console.WriteLine($"Title: {Title}");
            Console.WriteLine($"Author: {Author}");
        }
    }
}
```

File 2: Program.cs

```
using System;
using MyLibrary; // Import the MyLibrary namespace
```

```
class Program
{
    static void Main(string[] args)
    {
        // Create an instance of the Book class from MyLibrary namespace
        Book myBook = new Book();
        myBook.Title = "To Kill a Mockingbird";
        myBook.Author = "Harper Lee";

        // Display book information
        myBook.DisplayInfo();
    }
}
```

Explanation:

1. Defining the Namespace (MyLibrary):

- In Library.cs, the Book class is defined within the MyLibrary namespace. This groups the Book class logically and avoids naming conflicts with other Book classes that might exist in different namespaces.

2. Using the Namespace:

- In Program.cs, the using MyLibrary; directive makes the Book class accessible without needing to specify the full namespace path (MyLibrary.Book).

3. Creating and Using Types:

- An instance of the Book class is created and used in the Main method. The Book class is fully qualified because the MyLibrary namespace has been imported.

Summary

Namespaces in C# provide a way to organize code and prevent naming conflicts. By grouping related types together, namespaces make it easier to manage and navigate complex projects. You can define and use namespaces to improve code readability and maintainability.