

Using the Command Line Compiler in C#

The command line compiler (csc.exe) allows you to compile C# programs without the need for an integrated development environment (IDE) like Visual Studio. This can be particularly useful for quick tests, automation, and learning purposes.

Step 1: Write Your C# Code

First, you need a C# source file. You can create this file using any text editor.

1. Create a Source File:

- Open your favorite text editor (e.g., Notepad, Visual Studio Code).
- Write the following C# code and save the file as `HelloWorld.cs`:

```
using System;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
        }
    }
}
```

Step 2: Open Command Prompt or Terminal

Next, open Command Prompt (Windows) or Terminal (macOS/Linux).

Step 3: Navigate to the Directory Containing Your Source File

Use the `cd` command to navigate to the directory where you saved your `HelloWorld.cs` file. For example:

```
""sh
cd C:\Path\To\Your\Source\File
""
```

Step 4: Compile the C# Program

Use the C# compiler (`csc.exe`) to compile your program. The `csc.exe` compiler is part of the .NET SDK. Make sure you have the .NET SDK installed and properly configured in your system's PATH.

1. Compile the Program:

- Run the following command to compile `HelloWorld.cs`:

```
```sh
csc HelloWorld.cs
```
```

This will generate an executable file named `HelloWorld.exe` in the same directory.

Step 5: Run the Compiled Program

1. Execute the Program:

- Run the generated executable by typing the following command:

```
```sh
HelloWorld.exe
```
```

You should see the output:

```
```
Hello, World!
```
```

Example: Compiling and Running a Program with Command Line Arguments

1. Create a Source File:

- Write the following C# code and save it as `ArgsExample.cs`:

```
using System;

namespace ArgsExample
{
    class Program
    {
        static void Main(string[] args)
        {
            if (args.Length > 0)
            {
                Console.WriteLine("Arguments passed to the program:");
                foreach (string arg in args)
                {
                    Console.WriteLine(arg);
                }
            }
            else
            {

```

```
        Console.WriteLine("No arguments were passed.");
    }
}
}
```

2. Compile the Program:

- Use the command:

```
``sh
csc ArgsExample.cs
``
```

3. Run the Program with Arguments:

- Execute the program with arguments:

```
``sh
ArgsExample.exe arg1 arg2 arg3
``
```

You should see the output:

```
``
Arguments passed to the program:
arg1
arg2
arg3
``
```

Handling Compilation Errors

If there are errors in your code, the compiler will display error messages. You can use these messages to debug and fix your code.

Summary

Using the command line compiler `csc.exe` is a straightforward way to compile and run C# programs without an IDE. It is particularly useful for quick tests, learning, and automation scripts.

By following these steps, you can compile and run C# programs, handle command line arguments, and debug any compilation errors that arise.