

使用 Visual Studio 调试 C# 或 Visual Basic .NET Core Hello World 应用程序

2019/12/05 · 👤 👤 👤

本文内容

[“调试”生成配置](#)

[设置断点](#)

[设置条件断点](#)

[单步执行程序](#)

[生成“发布”版本](#)

[后续步骤](#)

至此，已按照[使用 Visual Studio 2019 创建第一个 .NET Core 控制台应用程序](#)中的步骤操作，创建和运行简单的控制台应用程序。编写和编译应用程序后，可以开始进行测试。Visual Studio 提供一整套调试工具，方便你在排除应用程序故障时使用。

“调试”生成配置

“调试”和“发布”是 Visual Studio 的两种默认生成配置。当前的生成配置显示在工具栏上。下面的工具栏图像显示 Visual Studio 配置为编译应用的“调试”版本：



首先运行应用的“调试”版本。“调试”生成配置会禁用大多数编译器优化，并在生成过程中提供更丰富的信息。

设置断点

运行程序，并尝试一些调试功能：

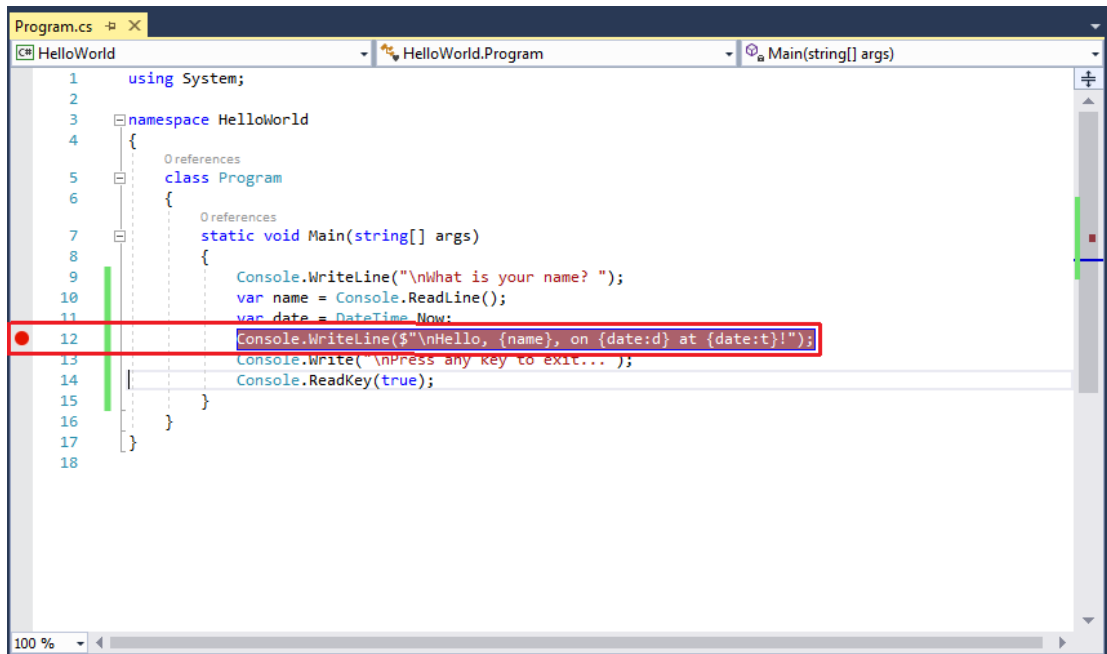
C# Visual Basic

1. 通过读取 `Console.WriteLine($"\\nHello, {name}, on {date:d} at {date:t}!");` 的行上单击代码窗口的左边距来设置该行的断点。也可以通过

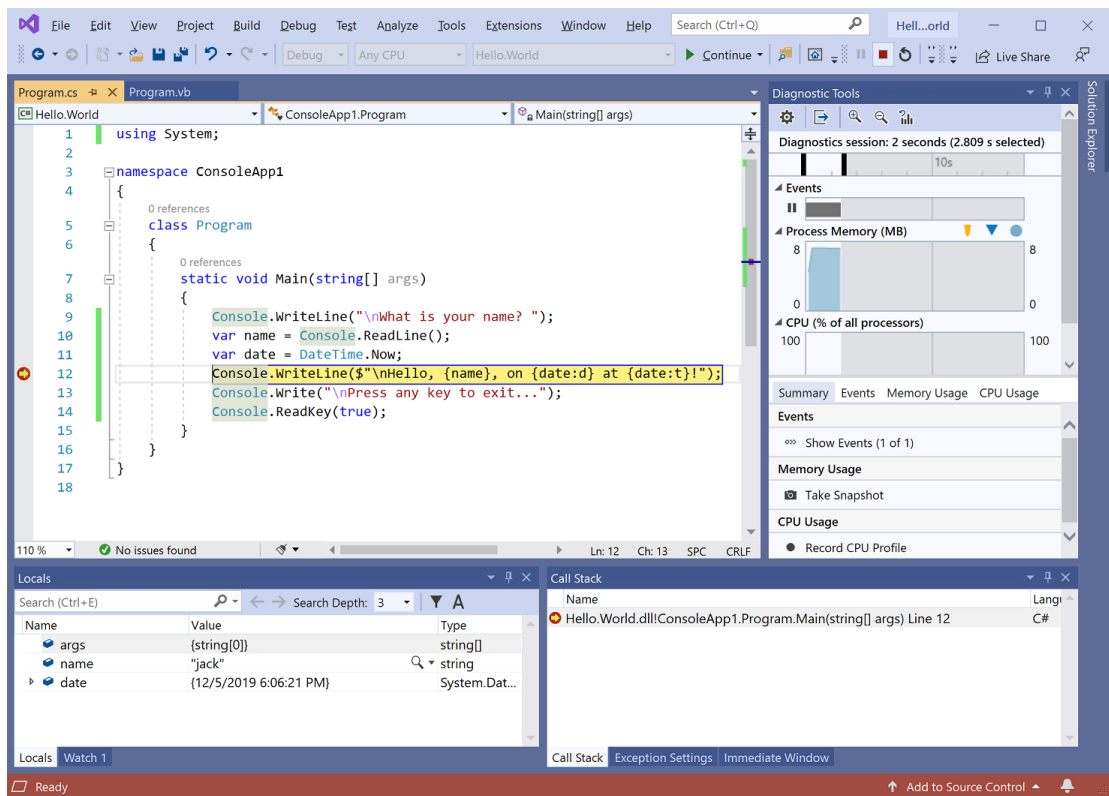
将插入符号置于代码行中，然后按 F9 或从菜单栏中选择“调试” > “切换断点”来设置断点。

断点会在执行包含断点的代码行之前 暂时中断执行应用程序。

如下图所示，Visual Studio 通过突出显示此代码行，并在其左侧边缘显示红色圆圈来指明该行设置了断点。



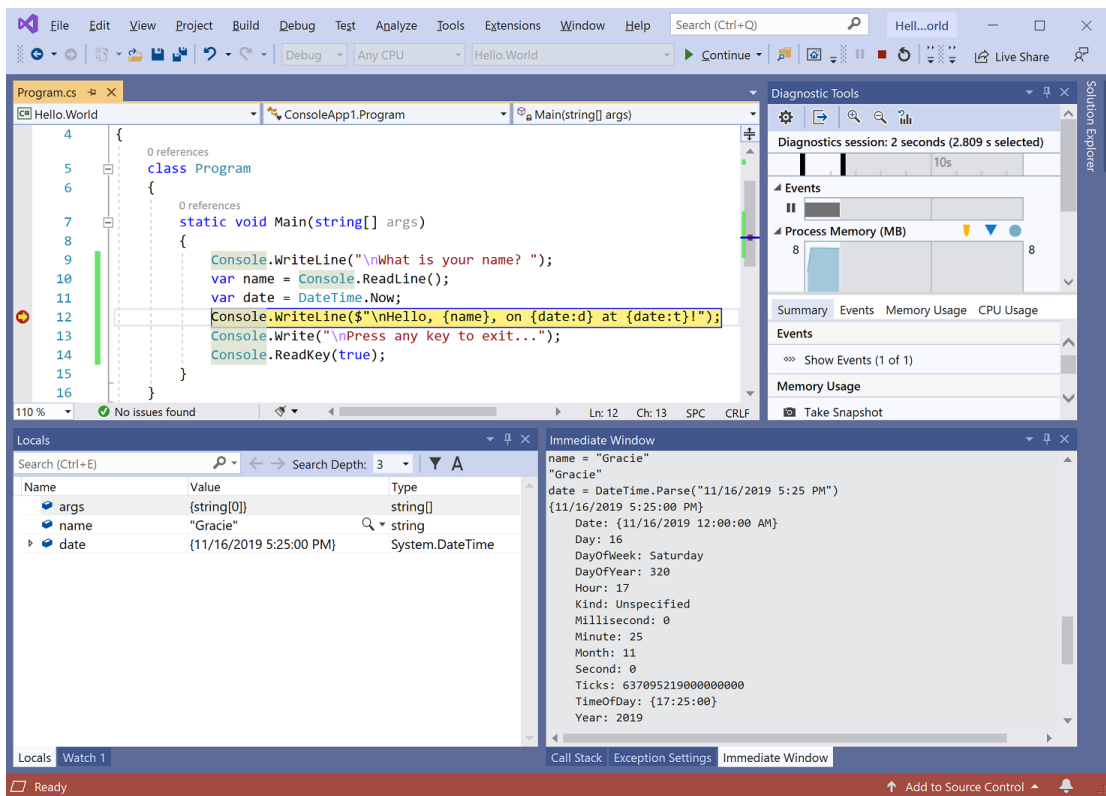
2. 选择工具栏上含绿色箭头的“HelloWorld” 按钮、按 F5 或选择“调试” > “启动调试”，在“调试”模式下运行程序。
3. 当程序提示输入名称时，在控制台窗口中输入字符串，然后按 Enter 。
4. 到达断点时，程序停止执行，然后执行 Console.WriteLine 方法。“局部变量”窗口显示当前正在执行的方法中定义的变量值。



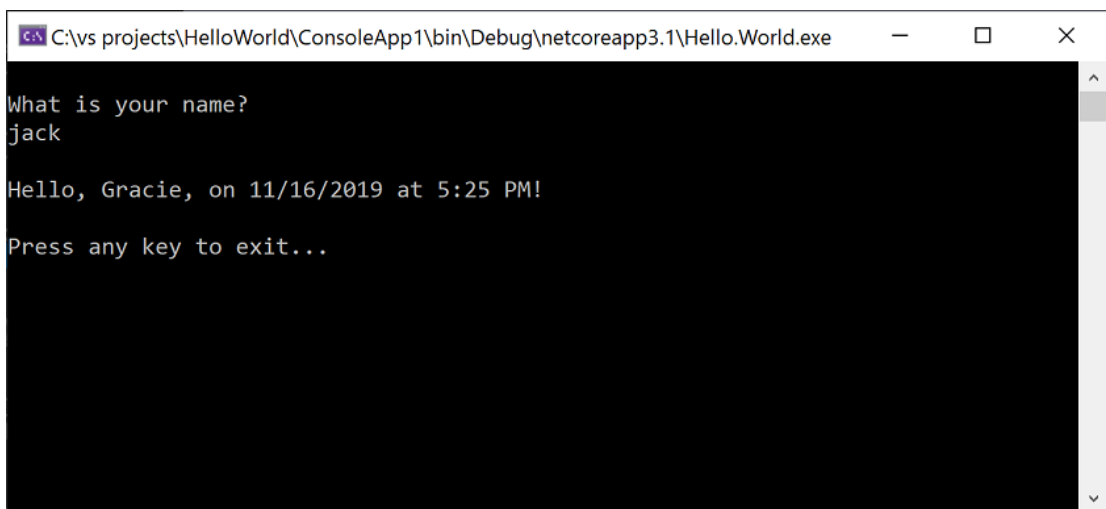
5. 在“即时”窗口中，可以与正在调试的应用程序进行交互。可以通过交互方式更改变量值，看看这样会对程序产生哪些影响。

- 如果“即时”窗口不可见，请选择“调试” > “Windows” > “即时”来显示它。
- 在“即时”窗口中输入 `name = "Gracie"`，然后按 Enter 键。
- 在“即时”窗口中输入 `date = DateTime.Parse("11/16/2019 5:25 PM")`，然后按 Enter 键。

“即时”窗口显示字符串变量的值和 [DateTime](#) 值的属性。此外，“局部变量”窗口中也会更新变量值。



6. 选择工具栏中的“继续”按钮，或选择“调试” > “继续”，继续执行程序。控制台窗口中显示的值对应于在“即时”窗口中所做的更改。



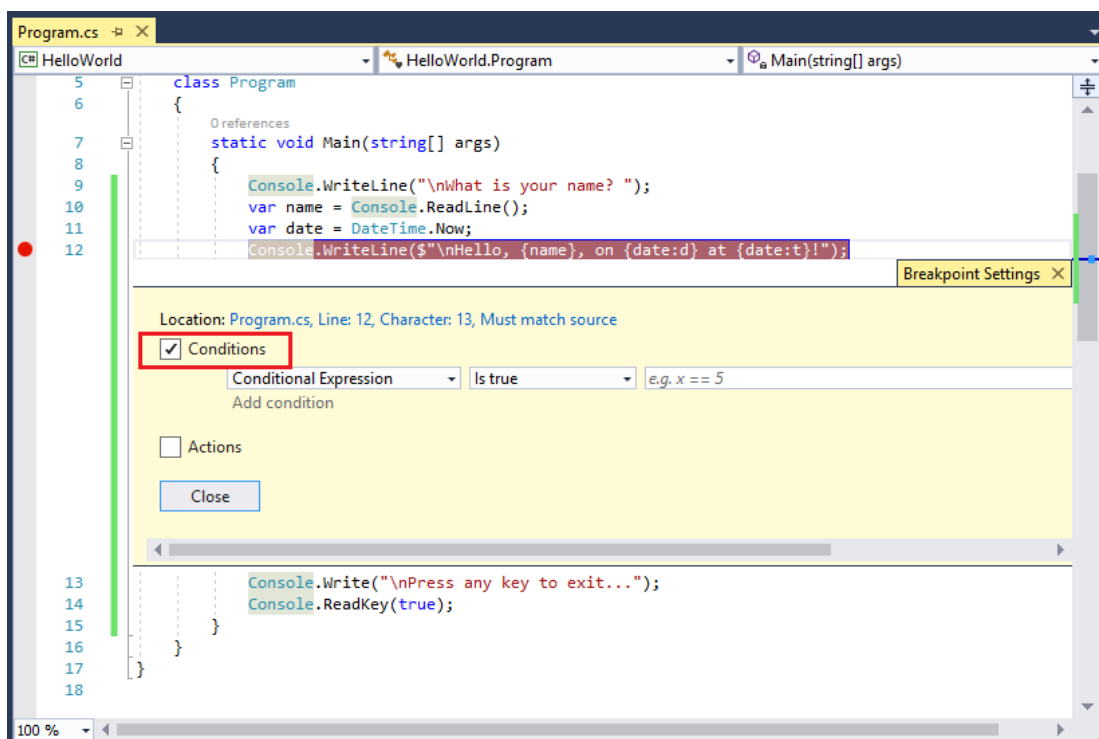
7. 按任意键，退出应用程序并停止调试。

设置条件断点

程序显示用户输入的字符串。如果用户没有输入任何内容，情况又如何呢？可使用名为“条件断点”的实用调试功能来测试这种情况，该功能可在满足一个或多个条件时中断程序执行。

若要设置条件断点，并测试用户无法输入字符串时的情况如何，请执行以下操作：

1. 右键单击表示断点的红点。在上下文菜单中，选择“条件”，打开“断点设置”对话框。选择“条件”框（如果尚未选择）。



2. 对于“条件表达式”，将“e.g. x == 5”替换为以下内容：

| | |
|----------------------------|----|
| C# | 复制 |
| String.IsNullOrEmpty(name) | |

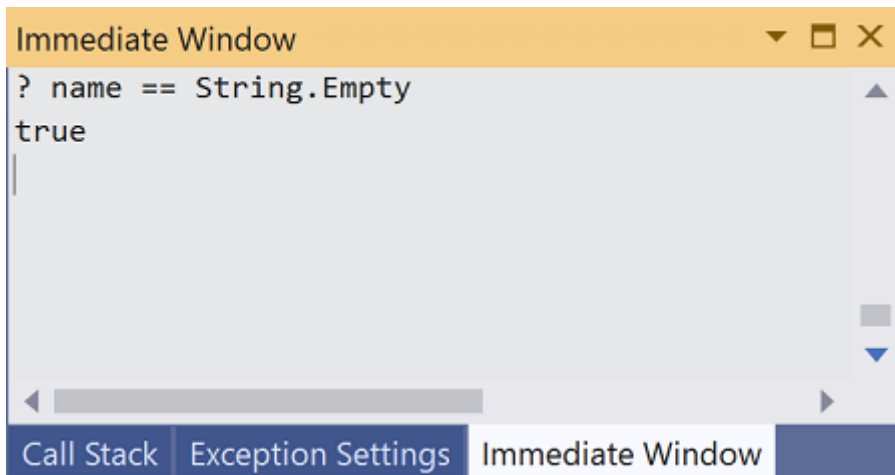
要测试的代码条件是 `String.IsNullOrEmpty(name)` 方法调用是否为 `true`，因为 `name` 尚未分配有值或值为空字符串（`""`）。与条件表达式不同，你还可以指定命中次数，这样程序就会在语句的执行次数达到指定值时中断执行；也可以指定筛选条件，这样就可以根据诸如线程标识符、进程名称或线程名称之类的属性来中断程序执行。

3. 选择“关闭”以关闭对话框。
4. 通过按 F5 调试来启动程序。
5. 在控制台窗口中，在看到输入名称的提示时按 Enter 键。
6. 由于符合指定的条件（`name` 为 `null` 或 `String.Empty`），因此程序在到达断点时以及在 `Console.WriteLine` 方法执行前停止执行。

7. 选择“局部变量”窗口，其中显示当前正在执行的方法的局部变量值。在这种情况下，Main 是当前正在执行的方法。请注意，name 变量的值为 "" 或 [String.Empty](#)。

8. 在“即时”窗口中输入下面的语句并按 Enter，确认值为空字符串。结果为 true。

| | |
|-----------------------------------|----|
| C# | 复制 |
| <pre>? name == String.Empty</pre> | |



9. 选择工具栏上的“继续”按钮，继续执行程序。

10. 按任意键，关闭控制台窗口并停止调试。

11. 单击代码窗口左侧边缘上的点，或选中该代码行，选择“调试”>“切换断点”，从而清除断点。

单步执行程序

使用 Visual Studio，还可以单步执行程序，并监视其执行情况。通常可以设置断点，并使用此功能单步执行程序流，尽管是一小部分程序代码。因为程序很小，因此可以单步执行整个程序：

C# Visual Basic

1. 在菜单栏上，选择“调试”>“单步执行”，或按 F11。Visual Studio 会在要执行的下一行旁边突出显示一个箭头。

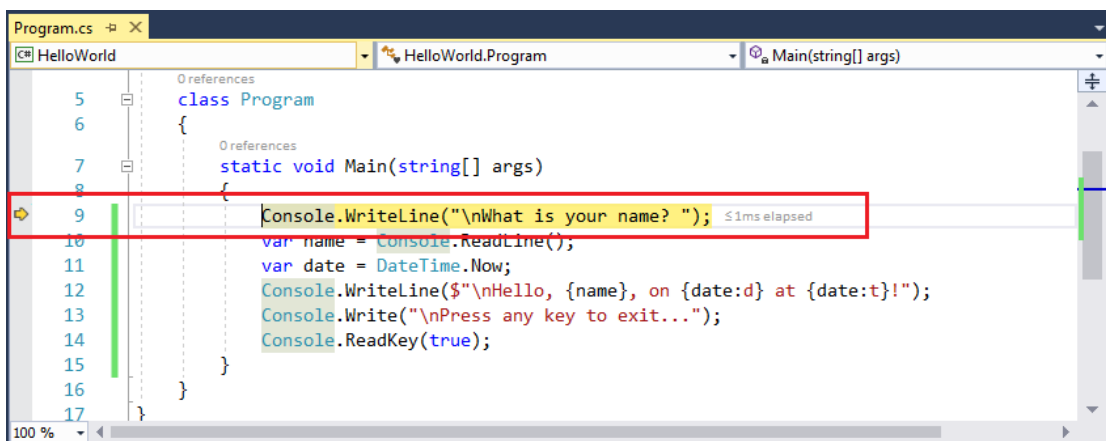


```
Program.cs
HelloWorld
HelloWorld.Program
Main(string[] args)

5  class Program
6  {
7      static void Main(string[] args)
8      {
9          Console.WriteLine("\nWhat is your name? ");
10         var name = Console.ReadLine();
11         var date = DateTime.Now;
12         Console.WriteLine($"Hello, {name}, on {date:d} at {date:t}!");
13         Console.Write("\nPress any key to exit...");
14         Console.ReadKey(true);
15     }
16 }
17
```

此时，“局部变量”窗口显示程序只定义了一个变量，即 `args`。由于尚未向程序传递任何命令行自变量，因此它的值是一个空字符串数组。此外，Visual Studio 还打开了一个空白控制台窗口。

2. 选择“调试” > “单步执行”，或按 F11。Visual Studio 现在突出显示要执行的下一行。如图所示，从上一语句执行到该行代码花费了不到 1 毫秒的时间。`args` 仍然是唯一声明的变量，控制台窗口仍为空白。



```
Program.cs
HelloWorld
HelloWorld.Program
Main(string[] args)

5  class Program
6  {
7      static void Main(string[] args)
8      {
9          Console.WriteLine("\nWhat is your name? ");
10         var name = Console.ReadLine();
11         var date = DateTime.Now;
12         Console.WriteLine($"Hello, {name}, on {date:d} at {date:t}!");
13         Console.Write("\nPress any key to exit...");
14         Console.ReadKey(true);
15     }
16 }
17
```

3. 选择“调试” > “单步执行”，或按 F11。Visual Studio 突出显示包含 `name` 变量赋值的语句。“局部变量”窗口显示 `name` 为 `null`，控制台窗口显示字符串“`What is your name?`”。
4. 在控制台窗口中输入字符串，然后按 Enter，从而响应提示。控制台无响应，输入的字符串未显示在控制台窗口中，但 `Console.ReadLine` 方法将捕获输入。
5. 选择“调试” > “单步执行”，或按 F11。Visual Studio 突出显示包含 `date` 变量赋值的语句。“局部变量”窗口显示 `Console.ReadLine` 方法调用返回的值。控制台窗口还显示在提示符处输入的字符串。
6. 选择“调试” > “单步执行”，或按 F11。“局部变量”窗口显示通过 `DateTime.Now` 属性赋值后的 `date` 变量值。控制台窗口保持不变。
7. 选择“调试” > “单步执行”，或按 F11。Visual Studio 调用 `Console.WriteLine(String, Object, Object)` 方法。控制台窗口会显示格式化的

字符串。

8. 选择“调试” > “跳出”，或按 Shift + F11。这将停止单步执行。控制台窗口会显示一条消息，并等待用户按任意键。
9. 按任意键，关闭控制台窗口并停止调试。

生成“发布”版本

测试应用程序的“调试”版本后，还应该编译并测试“发布”版本。发布版本包含编译器优化，有时可能会对应用程序的行为产生不良影响。例如，旨在提升性能的编译器优化可能会在多线程应用程序中创建争用条件。

若要生成和测试控制台应用程序的发布版本，请将工具栏上的生成配置从“调试”更改为“发布”。



按 F5 或选择“生成”菜单中的“生成解决方案”后，Visual Studio 会编译应用程序的“发布”版本。可像测试“调试”版本一样测试“发布”版本。

后续步骤

调试完应用程序后，下一步是发布应用程序的可部署版本。若要了解如何执行此操作，请参阅[使用 Visual Studio 发布 .NET Core Hello World 应用程序](#)。

此页面有帮助吗？

是 否