**互操作性（C# 编程指南）**

互操作性使您能够保留和利用在现有非托管代码中的投入。 运行在公共语言运行时 (CLR) 的控制之下的代码称为“托管代码”，运行在 CLR 之外的代码称为“非托管代码”。 COM、COM+、C++ 组件、ActiveX 组件和 Microsoft Win32 API 都是非托管代码的示例。

.NET Framework 通过平台调用服务、[System.Runtime.InteropServices](https://msdn.microsoft.com/zh-cn/library/system.runtime.interopservices.aspx) 命名空间、C++ 互操作性和 COM 互操作性（COM 互操作）来实现与非托管代码的互操作性。

**本节内容**

[互操作性概述（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173185.aspx)

介绍在 C# 托管代码和非托管代码之间进行互操作的方法。

[如何：通过使用 Visual C# 功能访问 Office 互操作对象（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/dd264733.aspx)

描述 Visual C# 2010 中引入的功能以便于 Office 编程。

[如何：在 COM 互操作编程中使用索引属性（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ee310208.aspx)

描述如何使用索引属性以访问包含参数的 COM 属性。

[如何：使用平台调用播放波形文件（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173187.aspx)

介绍如何使用平台调用服务在 Windows 操作系统中播放 .wav 声音文件。

[演练：Office 编程（C# 和 Visual Basic）](https://msdn.microsoft.com/zh-cn/library/ee342218.aspx)

演示如何创建包含一个指向该工作簿的 Excel 工作簿和 Word 文档。

[COM 类示例（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/c3fd4a20.aspx)

演示如何将 C# 类作为 COM 对象公开。

# 平台调用

平台调用是一种服务，它使托管代码可以调用在动态链接库 (DLL)（如 Microsoft Win32 API 中的那些 DLL）中实现的非托管函数。此服务将查找并调用导出的函数，然后根据需要跨越互用边界封送其参数（整数、字符串、数组、结构等）。

有关更多信息，请参见[使用非托管 DLL 函数](https://msdn.microsoft.com/zh-cn/library/26thfadc.aspx)和 [如何：使用平台调用播放波形文件（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173187.aspx)。

| **https://wizardforcel.gitbooks.io/msdn-csharp/content/img/note.jpg 注意** |
| --- |
| [公共语言运行时 (CLR)](https://msdn.microsoft.com/zh-cn/library/8bs2ecf4.aspx) (CLR) 管理对系统资源的访问。调用 CLR 外部的非托管代码会避开此安全机制，因此会带来安全风险。例如，非托管代码可能会直接调用非托管代码中的资源，从而避开 CLR 安全机制，。有关更多信息，请参见 [.NET Framework Security](http://go.microsoft.com/fwlink/?LinkId=37122)（.NET Framework 安全性）。 |

## C++ Interop

可以使用 C++ Interop（又称为 It Just Works (IJW)）包装本机 C++ 类，使得用 C# 或其他 .NET Framework 语言编写的代码可以使用它。为此，可以编写 C++ 代码来包装本机 DLL 或 COM 组件。与其他 .NET Framework 语言不同，Visual C++ 支持互操作性，允许托管代码和非托管代码存在于同一个应用程序中，甚至存在于同一个文件中。然后，可以使用 **/clr** 编译器开关生成 C++ 代码，从而生成托管程序集。最后，在 C# 项目中添加一个对该程序集的引用，并像使用其他托管类那样使用被包装对象。

## 向 C# 公开 COM 组件

可以使用 C# 项目中的 COM 组件。一般步骤如下所示：

1. 找到要使用的 COM 组件并注册它。使用 regsvr32.exe 注册或注销 COM DLL。
2. 在项目中添加对 COM 组件或类型库的引用。

添加引用时，Visual Studio 会用到[Tlbimp.exe（类型库导入程序）](https://msdn.microsoft.com/zh-cn/library/tt0cf3sx.aspx)，后者将类型库作为输入并输出一个 .NET Framework 互操作程序集。该程序集又称为运行时可调用包装 (RCW)，其中包含了包装类型库中的 COM 类和接口的托管类和接口。Visual Studio 将生成组件的引用添加至项目。

1. 创建在 RCW 中定义的类的实例。而这样会创建 COM 对象的实例。
2. 像使用其他托管对象那样使用该对象。当垃圾回收对该对象进行回收后，COM 对象的实例也会从内存中释放出来。

有关更多信息，请参见 [向 .NET Framework 公开 COM 组件](https://msdn.microsoft.com/zh-cn/library/z6tx9dw3.aspx)。

## 向 COM 公开 C

COM 客户端可以使用已经正确公开的 C# 类型。公开 C# 类型的基本步骤如下所示：

1. 在 C# 项目中添加互操作特性。

可以通过修改 Visual C# 项目属性使程序集 COM 可见。有关更多信息，请参见 [“程序集信息”对话框](https://msdn.microsoft.com/zh-cn/library/1h52t681.aspx)。

1. 生成 COM 类型库并对它进行注册以供 COM 使用。

可以修改 Visual C# 项目属性以自动注册 COM interop 的 C# 组件。Visual Studio 使用 [Regasm.exe（程序集注册工具）](https://msdn.microsoft.com/zh-cn/library/tzat5yw6.aspx)，方法是使用 **/tlb** 命令行切换，其将管理的组件作为输入，以生成类型库。此类型库描述程序集中的 **public** 类型并添加注册表项，以便 COM 客户端可以创建托管类。

有关更多信息，请参见[向 COM 公开 .NET Framework 组件](https://msdn.microsoft.com/zh-cn/library/zsfww439.aspx)和 [COM 类示例（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/c3fd4a20.aspx)。

## 请参阅

[Improving Interop Performance](http://go.microsoft.com/fwlink/?LinkId=99564)

[Introduction to COM Interop](http://go.microsoft.com/fwlink/?LinkId=112406)

[Marshaling between Managed and Unmanaged Code](http://go.microsoft.com/fwlink/?LinkId=112398)

[与非托管代码交互操作](https://msdn.microsoft.com/zh-cn/library/sd10k43k.aspx)

[Advanced COM Interoperability](https://msdn.microsoft.com/zh-cn/library/bd9cdfyx.aspx)

[C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx)