**捕捉非 CLS 异常**

包括 C++/CLI 在内的某些 .NET 语言允许对象引发不是由 [Exception](https://msdn.microsoft.com/zh-cn/library/system.exception.aspx) 派生的异常。这类异常称为“非 CLS 异常”或“非异常”。在 Visual C# 中，您不能引发非 CLS 异常，但是可以采用两种方法捕捉这类异常：

* 在 catch (Exception e) 块中作为 [RuntimeWrappedException](https://msdn.microsoft.com/zh-cn/library/system.runtime.compilerservices.runtimewrappedexception.aspx) 捕捉。

默认情况下，Visual C# 程序集将非 CLS 异常作为包装异常来捕捉。如果需要访问可通过 [WrappedException](https://msdn.microsoft.com/zh-cn/library/system.runtime.compilerservices.runtimewrappedexception.wrappedexception.aspx) 属性访问的原始异常，请使用此方法。本主题后面的过程将解释如何使用此方式捕捉异常。

* 在位于 catch (Exception) 或 catch (Exception e) 块之后的常规 catch 块（没有指定异常类型的 catch 块）中。

如果您要执行某个操作（如写入日志文件）以响应非 CLS 异常，并且不需要访问异常信息，请使用此方法。默认情况下，公共语言运行时会包装所有异常。要禁用此行为，请将程序集级别特性添加到您的代码中，该特性通常在 AssemblyInfo.cs 文件中。[assembly: RuntimeCompatibilityAttribute(WrapNonExceptionThrows = false)]

**捕捉非 CLS 异常**

1. 在 catch(Exception e) block 中，使用 **as** 关键字来测试 e 能否强制转换为 [RuntimeWrappedException](https://msdn.microsoft.com/zh-cn/library/system.runtime.compilerservices.runtimewrappedexception.aspx)。
2. 通过 [WrappedException](https://msdn.microsoft.com/zh-cn/library/system.runtime.compilerservices.runtimewrappedexception.wrappedexception.aspx) 属性访问原始异常。

下面的示例演示如何捕捉由使用 C++/CLR 编写的类库引发的非 CLS 异常。请注意，在此示例中，Visual C# 客户端代码事先已经知道要引发的异常类型为 [System.String](https://msdn.microsoft.com/zh-cn/library/system.string.aspx)。只要 [WrappedException](https://msdn.microsoft.com/zh-cn/library/system.runtime.compilerservices.runtimewrappedexception.wrappedexception.aspx) 属性的原始类型可通过您的代码来访问，就可以将该属性强制转换回其原始类型。

// Class library written in C++/CLR.

ThrowNonCLS.Class1 myClass = new ThrowNonCLS.Class1();

try

{

// throws gcnew System::String(

// "I do not derive from System.Exception!");

myClass.TestThrow();

}

catch (Exception e)

{

RuntimeWrappedException rwe = e as RuntimeWrappedException;

if (rwe != null)

{

String s = rwe.WrappedException as String;

if (s != null)

{

Console.WriteLine(s);

}

}

else

{

// Handle other System.Exception types.

}

}