示例程序常见问题解答

2015年6月12日

版本：V1.0.2

# 目 录

目录

[目 录 1](#_Toc444779749)

[1. 解决.net framework 4.0环境下遇到版本不同编译不通过的方法详解 1](#_Toc444779750)

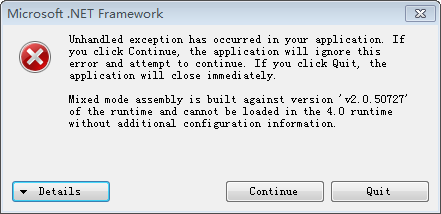
[2. 版本维护 1](#_Toc444779751)

# 解决.net framework 4.0环境下遇到版本不同编译不通过的方法详解

**本文内容：**

a.问题引出

最近部门的开发环境都更新到了WIN7+.NET framework4+VS2010上，在体验新技术和新环境带给我们提高效率的方式方法的同时也带来了一些兼容性的问题；这几天项目闲暇时在研究SQLite，在做实验的时候碰到个问题，代码编译通过执行时反复异常中断，查到后面原来是SQLite.dll是在framework2.0环境下编译的而现在的运行环境是framework4.0，所以就出现了运行异常，如图:



b.问题解决

由于以后的开发过程中，可能会引用一些第三方的组件，有可能是老的framework版本下编译的，同样会出现此类的问题，所以在本文中提供一个解决办法和说明造成此问题的原因。

**查阅了相关资料，stackoverflow.com上建议在config文件中增加配置：**

.代码如下:

<startup useLegacyV2RuntimeActivationPolicy="true">

<supportedRuntime version="v4.0"/>

</startup>

MSDN上也对此方法进行了阐述，参见<http://msdn.microsoft.com/en-us/library/bbx34a2h(VS.100).aspx>

但是使用此方法编译后问题仍然没有解决，但是问题可以明确锁定为.net runtime环境的问题了。

.代码如下:

<startup useLegacyV2RuntimeActivationPolicy="true">

<supportedRuntime version="v4.0"/>

<requiredRuntime version="v4.0.30319" />

</startup>

再次运行，问题解决。这下需要了解是什么原因造成的？

c.原因分析

supportedRuntime标签用来 具体说明应用程序支持的是哪个.framework运行时的版本；

requiredRuntime标签用来 具体说明应用程序只支持1.0版本的公用语言运行时间。如果使用1.1版本或者后面的版本来编译，应用程序必须使用<supportedRuntime>元素；

**注意：**

<supportedRuntime>必须通过1.1版本或后面的版本而编译的应用程序来使用。只支持1.0版本的运行时间的应用程序必须使用<requiredRuntime>。

再次查阅CLR Runtime版本的相关资料，引用以下CLR运行规则和各版本间关系汇总表格，如下：

**规则：**

1. CLR4.0及以上版本编译的应用程序总是运行在应用程序所被编译的CLR版本上；

2. CLR4.0以下版本编译的应用程序优先运行在被编译的CLR版本上，如果此版本不存在，则运行最新的小于CLR4.0的版本；

**汇总如下：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EXE被编译的CLR版本号** | **机器上安装有CLR 1.1？** | **机器上安装有CLR 2.0?** | **机器上安装有CLR 4.0?** | **结果** |
| **1.1** | 是 | 无所谓 | 无所谓 | 加载CLR 1.1 |
| **2.0** | 无所谓 | 是 | 无所谓 | 加载CLR 2.0 |
| **1.1** | 否 | 是 | 无所谓 | 加载CLR 2.0 |
| **1.1** | 否 | 否 | 是 | 失败 |
| **2.0** | 无所谓 | 否 | 是 | 失败 |

至此我们就能很清楚的搞清各个CLR版本之间的联系了。

总结：微软每次版本升级都会造成一些新的问题的出现，早年从CLRv1.0到CLRv2.0曾经也造成很多的困惑，可能也是由于过大的组织造成不能面面俱到吧，希望在今后的版本升级中能真正做到无缝的版本联接。

# 版本维护

|  |  |  |
| --- | --- | --- |
| 版本 | 日期 | 备注 |
| V1.0.0 | 2016-3-3 | 初版 |