在.NET平台用C#这么久，自然会发现其版本很多，相应的概念也会很多，常常都是萌萌哒。而在实际工作中经常会遇到需要配置dll版本号，公钥token等场景，因而对C#、NET、CLR、框架类型等基础概念有个大略的了解有很大必要性。当涉及到程序集版本时，注意查看全局程序集缓存GAC（Global Assembly Cache），路径为c:\windows\assembly，还有几个相似的.net framework路径，同时可以使用IL Disassembler (x64)查看DLL中所引用的外部库信息。

* C#特性一览

C#1.0，foreach循环支持自动Dispose等小特性。

C#2.0，主要特性包括泛型、可空类型、匿名方法与委托的增强、迭代块。小特性有分部类型、静态类、包含不同访问修饰符的取值和赋值方法、命名空间别名、pragma指令以及固定大小的缓冲器。

C#3.0，主要特性就是LINQ了，包括：Lambda表达式、表达式树、扩展方法。其他特性包括：自动属性、数组和局部变量的隐式类型、对象和集合的初始化程序、匿名类型。

C#4.0，主要集中于提高互操作性的动态类型。小特性包括命名实参、可选参数、更好的COM交互性、泛型可变性。

C#5.0，主要特性是新的异步操作模型async, await。小特性包括forreach变量捕获的变化和调用者信息特性。

* 框架库特性一览

NET 2.0，ASP.NET添加母版页、预编译功能；WinForm添加TableLayoutPanel等布局类；通过双缓存、新的数据绑定模型、ClickOne部署增强性能；BackgroudWorker用于在多线程应用程序中更新UI。

NET 3.0，提供了四个大库：WPF、WCF、WF、Windows CardSpace安全识别系统

NET 3.5，支持LINQ，包括LINQ to Objects、LINQ to SQL、LINQ to XML，以及表达式树。其他特性包括System.AddIn框架、新增各种加密算法、Client Profile用于客户端精简部署、Entity Framework。

NET 4.0，微调核心API，如String.Join()、增加泛型接口和委托的协变性和逆变性、增加System.Numeric用于数值运算、用于延迟的初始值的Lazy<T>、Pair和Tuple泛型类家族等、MEF（Managed Extensibility Framework）可扩展性框架等。

NET 4.5，异步操作和TPL并行库。

* 运行时(CLR)特性

CLR 2.0：支持64位处理器，以及SQL SERVER中承载CLR的功能，这点其实就是CLR也支持类似于JVM的配置功能，包括内存和线程的细粒度管理。

CLR 4.0：改进了互操作封装性能、一致性IL Stubs Everywhere即IL的一致性、提供更加完善的安全模型、支持同一进程同时执行不同CLR，但最主要确实对程序员透明的垃圾回收。

* 框架模型(经常在安装删除文件是发现很多的Compact Framework，还挺困扰的)

桌面框架：客户端配置(Client profile)、32位&64位JIT、服务器&工作站版本。

精简框架(Compact Framework)：运行与Xbox、WinPhone，垃圾回收机制相对简单（不是分代式）。

微框架(Micro Framework)：是解释型而非JIT编译，是开源的。

Windows Runtime(WinRT)：是Win8引入的新平台，以跨平台的目的创建。

TIP：此部分内容主要作为作为知识点复习浏览所用，只要有个映象就足够了，其中的MEF很像Spring的解耦，动态的查找接口的实现类。