CLR支持的程序集类型。弱命名程序集（weakly named assembly）和强命名程序集（strongly named assembly）。两者结构相同，都使用PE文件格式、PE32(+)头、CLR头、元数据、清单表、IL，生成工具也相同。二者的区别：强命名程序集使用发布者的公钥/私钥进行了签名。这一堆密钥允许对程序集进行唯一性的标识、保护和版本控制。

程序集的部署方式。私有或全局。私有部署的程序集是指部署到应用程序基目录或子目录的程序集，弱命名程序集只能以私有方式部署。全局部署的程序集是指部署到一些公认位置的程序集（GAC），CLR在查找程序集时，会检查这些位置，强命名程序集既可以私有部署，也可以全局部署，弱命名程序集只能进行私有部署。

强命名程序集有4个重要特性：文件名、版本号、语言文化和公钥（由于公钥数字很大，所以通常使用从公钥派生的小哈希值，称为公钥标记）。以下是一个强命名程序集标识字符串及说明：

表格

低可信度描述已自动生成

对文件进行签名的含义：生成强命名程序集时，程序集的FileDef清单元数据表列出构成程序集的所有文件。每将一个文件名添加到清单，都对文件内容进行哈希处理。哈希值和文件名一道存储到FileDef表中。

强命名程序集的引用：

任何程序集都包含对其他强命名程序集的引用，因为System.Object定义在MSCorLib.dll中，MSCorLib.dll就是强命名程序集。使用csc.exe的/reference编译器开关指定想引用的程序集文件时，若提供完整路径，csc.exe会加载指定文件，并根据它的元数据生成程序集；若指定不包含路径的文件名，csc.exe会在如下目录查找程序集：

工作目录

csc.exe所在目录，目录中还包含CLR的各种DLL文件

使用/lib编译器开关指定的任何目录

使用LIB环境变量指定的任何目录

虽然编译时会在这里查找程序集，但运行时不会在这里加载程序集。安装.NET Framework时，实际会安装两套程序集文件。一套在CLR目录，另一套在GAC的子目录。CLR目录下的文件便于生成程序集，而GAC中的文件便于运行时加载。

强命名程序集防能篡改

用私钥对程序集进行签名，并将公钥嵌入程序集中，CLR就可以验证程序集是否被修改或破坏。程序集安装到GAC时，系统对包含清单的那个文件的内容进行哈希处理，将哈希值与PE文件中嵌入的RSA数字签名进行比较，值完全相同则表明文件内容未被篡改。此外，系统还会对程序集的其他内容进行哈希处理，并与清单文件的FileDef表进行比较。

图示

描述已自动生成

CLR如何加载程序集：

1定位程序集（搜索程序集）：CLR需要知道程序集的位置才能加载它。程序集可以位于本地文件系统、网络共享或者在内存中。CLR使用程序集的名称、版本号和公钥标记来定位程序集。

2加载程序集：一旦程序集被定位，CLR会将其加载到内存中。CLR会读取程序集的元数据，包括程序集的结构、类型定义、方法、属性等信息。

3验证程序集：CLR会对程序集进行验证，以确保其符合.NET的安全和类型安全规范。验证过程包括检查代码的正确性、访问权限以及类型的一致性等。

4分配内存：一旦程序集通过验证，CLR会为程序集分配内存空间。这包括分配内存来存储程序集的类型、方法、字段和其他数据。

5解析依赖项：程序集可能引用其他程序集或库，CLR会解析这些依赖项并加载它们。CLR会检查程序集的元数据，找到所需的依赖项，并加载它们。

6执行静态初始化：CLR会执行程序集中的静态初始化代码。静态初始化代码用于初始化静态字段和执行其他静态操作。这些代码在程序集加载时执行，以确保在使用程序集之前，静态资源已经准备就绪。

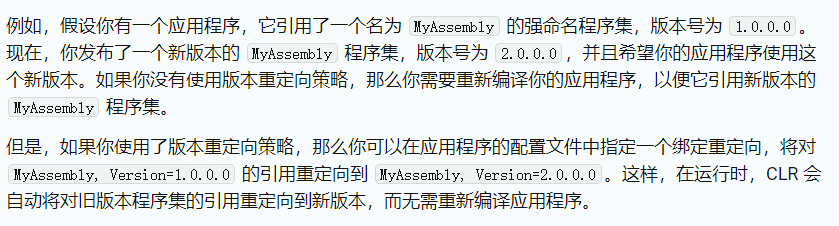
7创建类型：CLR会根据程序集中的元数据创建类型。这包括创建类、结构体、接口等。CLR会为每个类型分配内存，并初始化其字段和方法。

8执行代码：一旦类型和相关资源准备就绪，CLR就可以执行程序集中的代码。CLR使用即时编译（JIT）技术将中间语言（IL）代码转换为本机机器码，并执行这些机器码。

9处理异常：在执行代码期间，如果发生异常，CLR会捕获并处理它们。CLR提供了异常处理机制，可以捕获和处理各种类型的异常，以确保程序的稳定性和可靠性。

第23章查漏补缺

版本重定向：



如果没有使用版本重定向就需要重新编译程序来改变程序集的版本。