**[GAC](http://www.cnblogs.com/chenlulouis/archive/2010/06/30/1768176.html)**

一、GAC的作用

      全称是Global Assembly Cache作用是可以存放一些有很多程序都要用到的公共Assembly，例如System.Data、System.Windows.Forms等等。这样，很多程序就可以从GAC里面取得Assembly，而不需要再把所有要用到的Assembly都拷贝到应用程序的执行目录下面。举例而言，如果没有GAC，那么势必每个WinForm程序的目录下就都要从C:\WINDOWS\Microsoft.NET\Framework\vX下面拷贝一份System.Windows.Forms.dll，这样显然不如都从GAC里面取用方便，也有利于Assembly的升级和版本控制。

二、强命名程序集

     因为不同的公司可能会开发出有相同名字的程序集来，如果这些程序集都被复制到同一 个相同的目录下，最后一个安装的程序集将会代替前面的程序集。这就是著名的Windows “DLL Hell”出现的原因。

　　很明显，简单的用文件名来区分程序集是不够的，CLR需要支持某种机制来唯一的标识一个程序集。这就是所谓的强命名程序集。

　　一个强命名程序集包含四个唯一标志程序集的特性：文件名（没有扩展名），版本号，语言文化信息（如果有的话），公有秘钥。

　　这些信息存储在程序集的清单（manifest）中。清单包含了程序集的元数据，并嵌入在程序集的某个文件中。

　　下面的字符串标识了四个不同的程序集文件：

　　“MyType, Version=1.0.1.0,

　　Culture=neutral, PublicKeyToken=bf5779af662fc055”

　　“MyType, Version=1.0.1.0,

　　Culture=en-us, PublicKeyToken=bf5779af662fc055”

　　“MyType, Version=1.0.2.0,

　　Culture=neturl, PublicKeyToken=bf5779af662fc055”

　　“MyType, Version=1.0.2.0,

　　Culture=neutral, PublicKeyToken=dbe4120289f9fd8a”

　　如果一个公司想唯一的标识它的程序集，那么它必须首先获取一个公钥/私钥对，然后将共有秘钥和程序集相关联。不存在两个两个公司有同样的公钥/私钥对的情况，正是这种区分使得我们可以创建有着相同名称，版本和语言文化信息的程序集，而不引起任何冲突。

　　与强命名程序集对应的就是所谓的弱命名程序集。（其实就是普通的没有被强命名的程序集）。两种程序集在结构上是相同的。都使用相同的PE文件格式，PE表头，CLR表头，元数据，以及清单（manifest）。二者之间真正的区别在于：强命名程序集有一个发布者的公钥/私钥对签名，其中的公钥/私钥对唯一的标识了程序集的发布者。利用公钥/私钥对，我们可以对程序集进行唯一性识别、实施安全策略和版本控制策略，这种唯一标识程序集的能力使得应用程序在试图绑定一个强命名程序集时，CLR能够实施某些“已确知安全”的策略（比如只信任某个公司的程序集）。

三、如何创建强命名程序集, 如何查看强命名程序集的PublicKeyToken

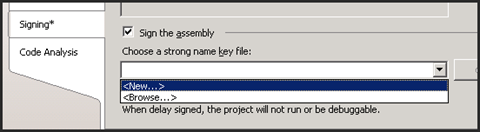
如何创建强命名程序集

===================

1. 在Visual Studio中的class library工程上点右键, 选择properties.

2.  选择左边的Signing选项卡.

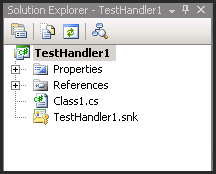
3. 勾选Sign the assembly复选框. 在下拉列表中选择<New...>.



4. 在弹出的对话框中给snk文件起一个名字. 按OK.



5. 程序集强命名完成.



如何查看强命名程序集的public key token

=========================

有时候你需要在web.config文件中或者其他地方引用自己写的强命名程序集, 你需要写入像下面这样的fully qualified name:

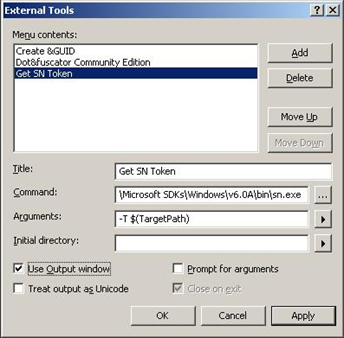
*MyNamespace.MyAssembly, version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089*

前面三个部分比较容易获得, 因为是你自己写的, 你当然知道assembly的名字, 版本, 还有culture信息. 比较麻烦的部分是如何获得自己签名的程序集的public key token. 一种平常的方法是使用[Reflector](http://www.aisto.com/roeder/dotnet/)来打开自己的程序集, 然后获得token(实际上, [Reflector](http://www.aisto.com/roeder/dotnet/)会给你如同上面例子那样的完整信息). 但是这有的时候还是显得有点未免杀鸡用牛刀了. 如果你已经打开了Visual Studio, 那么仅仅是在VS的菜单里点一个菜单项就能获得答案不是更好么? 下面就是步骤.

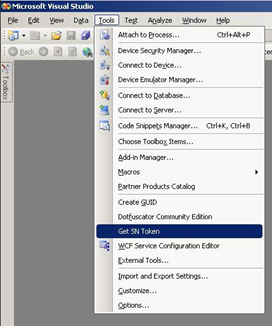
1. 在Visual Studio中, 打开Tools菜单, 然后点击External Tools这个菜单项.

2. 在弹出的External Tools对话框中, 点击Add按钮.

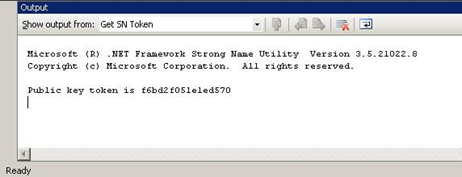
3. 按照下图进行配置. sn.exe这个工具在不同版本的VS下处于不同的文件夹中. 最简单的找到它的方式是在VS Command Prompt中输入"where sn.exe". 在参数框里写入"-T $(TargetPath)". 然后勾选"Use Output Window". 这样的话, 结果就会在VS的output window. 然后点击OK,



4. 结果如图.



5. 在输出窗口可以看到结果. 这在你的solution里有多个project的时候也是可以正常工作的. 只需要点击一下Solution Explorer中的Project, 然后点击我们的菜单项就可以了.

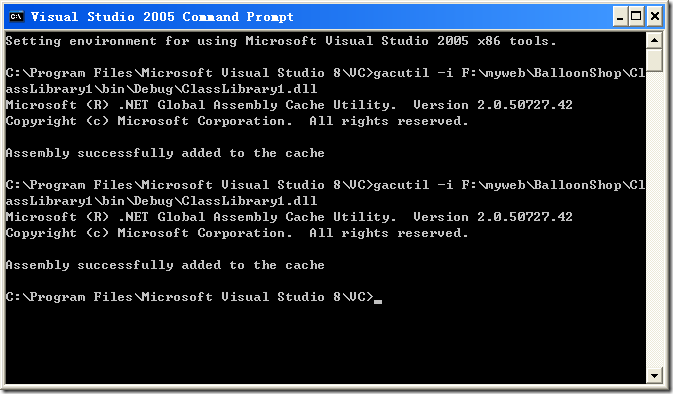


四、如何将自己的dll注册到GAC中

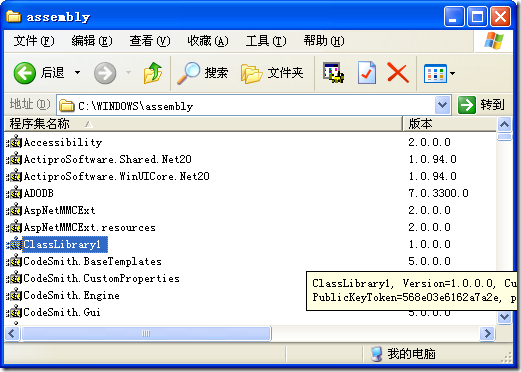
在开发和测试中，最常用的工具就是GACUtil.exe。 在**GAC**中**注册**程序集跟COM**注册**差不多，但相对更容易：  
    1．把程序集添加**到GAC**中： GACUtil /i sample.dll （参数/i是安装的意思）  
    2．把程序集移出**GAC** GACUtil /u sample.dll （参数/u就移除的意思）  
注意：不能将一个弱命名程序集安装**到GAC**中。  
如果你试图把弱命名程序集加入**到GAC**中，会收**到**错误信息：”  
    Failure adding assembly to the cache: Attempt to install an assembly without a strong name”  
    d)强命名程序集的私有部署

例子：

C:\Program Files\Microsoft Visual Studio 8\VC>gacutil -i F:\myweb\BalloonShop\Cl  
assLibrary1\bin\Debug\ClassLibrary1.dll

[](http://images.cnblogs.com/cnblogs_com/smallstone/WindowsLiveWriter/GAC_BDEE/image_2.png)

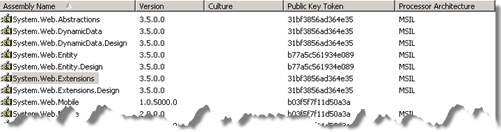
在C:\WINDOWS\assembly将会看到ClassLibrary1，注册成功

[](http://images.cnblogs.com/cnblogs_com/smallstone/WindowsLiveWriter/GAC_BDEE/image_4.png)

五、查看GAC文件内容以及将DLL复制出来

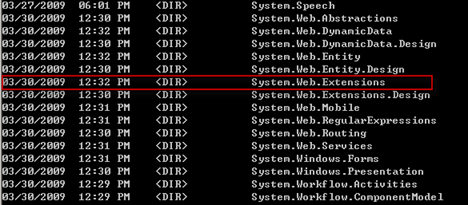
在项目中我们常常会引入第三方的dll，一般情况下我们都可以将所需的dll文件复制到硬盘上的一个地方，然后在项目中添加引用，这个操作很简单！但有时候我们会遇到这样的情况，就是所要引用的dll在目标机器的GAC里，这时我们就不能手动将它拷贝出来了。

      其实Windows的GAC是有对应的目录的，一般来说为c:\Windows\assembly\，这个目录有一些特殊，它里面存放的是本机已安装和注册的类库dll，并且不允许用户直接对其中的元素进行相关操作（如复制、剪切、粘贴、修改名称等），不过你可以直接将另一位置的dll文件直接拖放到这个目录下进行dll的安装，但是我们不能直接将已经安装进去的dll再拷贝出来。这里我将介绍一种方法来完成这个操作。

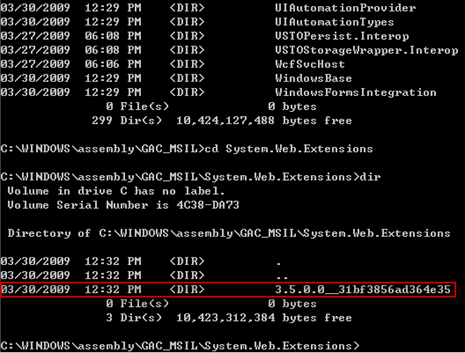
[](http://images.cnblogs.com/cnblogs_com/jaxu/WindowsLiveWriter/GACDLL_9A3C/6-19-2009%2010-12-55%20AM_2.png)

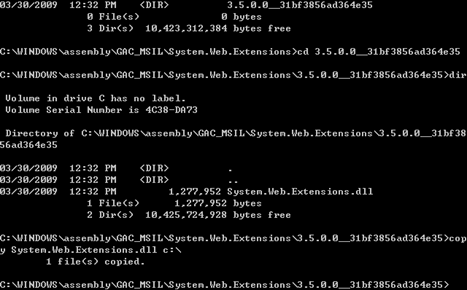
首先我们切换到Windows的命令行方式，即开始-运行-cmd-回车，然后转到GAC所在的目录，利用dir命令查看一下其中的内容，如下图。

[](http://images.cnblogs.com/cnblogs_com/jaxu/WindowsLiveWriter/GACDLL_9A3C/6-19-2009%2010-24-50%20AM_2.png) 似乎可以明白GAC中的目录结构了，基本上我们可以根据GAC目录中的Processor Architecture列来区分dir的类型，例如我们要找的System.Web.Extensions属于MSIL，在CMD方式下它应该就对应GAC\_MSIL，然后切换到这个目录下并dir。

[](http://images.cnblogs.com/cnblogs_com/jaxu/WindowsLiveWriter/GACDLL_9A3C/6-19-2009%2010-56-42%20AM_2.png)

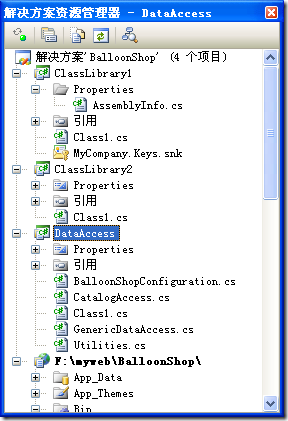
看到我们要找的System.Web.Extensions程序集了，它也是一个dir，继续切进去并dir。

[](http://images.cnblogs.com/cnblogs_com/jaxu/WindowsLiveWriter/GACDLL_9A3C/6-19-2009%2010-59-47%20AM_2.png)这时只有一个目录了，继续切进去，然后dir就可以看到我们最终想要的dll文件了，然后通过copy命令将它复制出来就OK了！

[](http://images.cnblogs.com/cnblogs_com/jaxu/WindowsLiveWriter/GACDLL_9A3C/6-19-2009%2011-02-37%20AM_2.png)

小技巧：在CMD方式下使用命令时，如果要输入的文件名或目录名太长，可以先敲部分字符，然后通过Tab键自动补全，Windows的command工具会自动为你找到相匹配的内容！

六、例子

[](http://images.cnblogs.com/cnblogs_com/smallstone/WindowsLiveWriter/GAC_BDEE/image_6.png)

如上图所示，新建了2个类库文件：ClassLibrary1、ClassLibrary2

1、使用上面的第三点创建了强类型程序集ClassLibrary1，并且注册到GAC中(可以使用上面第三点的方法，也可以使用反编译器进行反编译，可以查看到PublicKeyToken值。ClassLibrary2为null,ClassLibrary1为568e03e6162a7a2e)。

2、在DataAccess中引用ClassLibrary1、ClassLibrary2，编译DataAccess。

3、进入DataAccess工程的bin\debug文件夹下，只有ClassLibrary2.dll与DataAccess.dll（没有ClassLibrary1.dll）

由此可知：程序是从GAC中直接取得ClassLibrary1.dll的而不是从ClassLibrary1工程中将ClassLibrary1.dll拷贝到自身的debug中进行引用。