MSIL 简介

MSIL（Microsoft Intermediate Language）中文名为微软中间语言，它是一种类似于JAVA字节码的语言。在微软语言平台中，不管程序员使用C#、VB.NET或者J#等语言编写 的程序进行编译的时候，编译器将这几种语言编写的源代码编译为MSIL（微软中间语言）语言，此时再通过JIL（Just In Time实时编译器）编译为针对各种不同CPU的指令（注意因为是实时的编译器，所以它运行的时候是只运行需要编译的MSIL语言段，而不是全部一下编译 完，这是为了提高效率。）

MSIL好处：1.本系列文章用于研究和探讨MSIL中间语言，在实际应用中能够让你选择更好的语法，例如：是选择For还是选择Foreach做循环遍历？

2.深入的了解.NET内存的分配以及底层的语言运行原理。例如：.Net值类型和引用类型的内存分配？

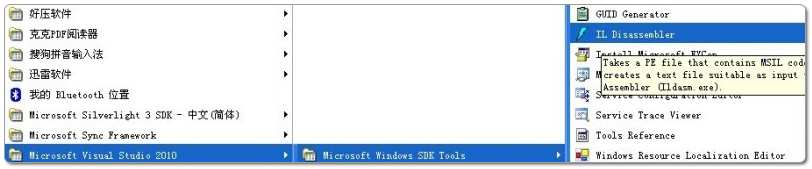
3.使用System.Reflection.Emit下的Emit加载MSIL命令，动态创建类。

初步认识MSIL：

一、我们新建一个C#语言编写的控制台程序命名为ConsoleApplication6，然后键入以下代码：

namespace ConsoleApplication6  
{  
class Program  
{  
staticvoid Main(string[] args)  
{  
string a ="Hello World!";  
Console.WriteLine(a);  
}  
}  
}

二、很明显上面的语句将会在控制台输出一个字符串“Hello World!”，那么这个控制台程序的MSIL语言是怎样的呢？我们下面可以通过VS2010自带的SDK工具IL Disassembler（ILDASM）打开上面应用程序创建的ConsoleApplication6.exe文件，以查看其MSIL语言。如下图方 式以打开ILDASM工具：



打开ILDASM工具界面如下：



本图代表一个ConsoleApplication6的命名空间下有一个类叫ConsoleApplication6.Program，这个类下面有一个 类清单(.class private auto ansi beforefieldinit)+默认构造函数(.ctor:void())+静态函数(Main:void(string[]))。

注意：上图的界面中有一些红色的三角形，或者蓝色的盾形等分别代表更多信息或者是命名空间等信息，其具体含义如下图所示：



三、点击默认构造函数.ctor:void()我们可以看到这个构造函数的MSIL语言如下：

.method public hidebysig specialname rtspecialname   
//.method表示对方法  
instance void .ctor() cil managed  
{  
// 代码大小 7 (0x7)  
.maxstack 8  
IL\_0000: ldarg.0  
IL\_0001: call instance void [mscorlib]System.Object::.ctor()  
IL\_0006: ret  
} // end of method Program::.ctor

ldarg.0 将索引为 0 的参数加载到计算堆栈上

call 调用由传递的方法说明符指示的方法

ret 从当前方法返回，并将返回值（如果存在）从调用方的计算堆栈推送到被调用方的计算堆栈上

意义：将索引为0的参数加载到堆栈上，然后调用System.Object对象的构造函数实例化对象，然后返回。

四、点击静态函数Main:void(string[])，可以打开一个界面，界面内看到MSIL语言如下：

.method private hidebysig staticvoid Main(string[] args) cil managed  
{  
.entrypoint //程序进入点  
// 代码大小 15 (0xf)  
.maxstack 1//堆栈分配  
.locals init ([0] string a)   
IL\_0000: nop   
IL\_0001: ldstr "Hello World!"//压入字符串,堆栈压操作  
IL\_0006: stloc.0//从计算堆栈的顶部弹出当前值并将其存储到索引 0 处的局部变量列表中。   
IL\_0007: ldloc.0//将索引 0 处的局部变量加载到计算堆栈上。  
IL\_0008: call void [mscorlib]System.Console::WriteLine(string)   
//呼叫WriteLine函数打印Hello World  
IL\_000d: nop  
IL\_000e: ret //返回  
} // end of method Program::Main

通过本文章，我们初步的认识了MSIL中间语言。也认识了一些MSIL的指令代码