

M1. Introduction to the C# Language and the .NET Framework

Objective:

- *create a multi-file assembly;*
- *build a strong name assembly;*
- *add and remove assemblies from GAC;*
- *use ILDASM, JetBrains dotPeek tools to analyze the contents of assemblies.*

1. Сборка будет состоять из двух модулей:

<pre>*FirstModule.cs – Блокнот Файл Правка Формат Вид Справка using SecondModule; public class Program { public static void Main() { SecondModuleType secondModule = new SecondModuleType(); string text = secondModule.Print(); System.Console.WriteLine("Hello from the FirstModule!"); System.Console.WriteLine(text); System.Console.ReadLine(); } }</pre>	<pre>*SecondModule.cs – Блокнот Файл Правка Формат Вид Справка namespace SecondModule { public class SecondModuleType { public string Print() { return "Hello from the SecondModule!"; } } }</pre>
---	--

2. Делаем модуль из SecondModule.cs:

```
C:\Users\Denis\Desktop>csc /t:module SecondModule.cs
```

3. Делаем общую сборку Result.exe:

```
C:\Users\Denis\Desktop>csc /out:Result.exe /t:exe /addmodule:SecondModule.netmodule FirstModule.cs
```

4. Смотрим содержимое сборки Result через ildasm.exe:

```
C:\Users\Denis\Desktop>ildasm Result.exe
```

5. Метаданные сборки Result:

```
=====
ScopeName : Result.exe
MVID      : {6FF8926F-6216-4962-853B-34C86A401C0F}
=====
Global functions
-----

Global fields
-----

Global MemberRefs
-----

TypeDef #1 (02000002)
-----
  TypDefName: Program (02000002)
  Flags      : [Public] [AutoLayout] [Class] [AnsiClass] [BeforeFieldInit] (00100001)
  Extends    : 01000002 [TypeRef] System.Object
  Method #1 (06000001) [ENTRYPOINT]
  -----
    MethodName: Main (06000001)
    Flags      : [Public] [Static] [HideBySig] [ReuseSlot] (00000096)
    RVA        : 0x00002050
    ImplFlags  : [IL] [Managed] (00000000)
    CallConvntn: [DEFAULT]
```

ReturnType: Void
No arguments.

Method #2 (06000002)

MethodName: .ctor (06000002)
Flags : [Public] [HideBySig] [ReuseSlot] [SpecialName] [RTSpecialName] [.ctor] (00001886)
RVA : 0x00002083
ImplFlags : [IL] [Managed] (00000000)
CallCnvtn: [DEFAULT]
hasThis
ReturnType: Void
No arguments.

TypeRef #1 (01000001)

Token: 0x01000001
ResolutionScope: 0x23000001
TypeRefName: System.Runtime.CompilerServices.RuntimeCompatibilityAttribute
MemberRef #1 (0a000001)

Member: (0a000001) .ctor:
CallCnvtn: [DEFAULT]
hasThis
ReturnType: Void
No arguments.

TypeRef #2 (01000002)

Token: 0x01000002
ResolutionScope: 0x23000001
TypeRefName: System.Object
MemberRef #1 (0a000007)

Member: (0a000007) .ctor:
CallCnvtn: [DEFAULT]
hasThis
ReturnType: Void
No arguments.

TypeRef #3 (01000003)

Token: 0x01000003
ResolutionScope: 0x23000001
TypeRefName: System.Runtime.CompilerServices.CompilationRelaxationsAttribute
MemberRef #1 (0a000002)

Member: (0a000002) .ctor:
CallCnvtn: [DEFAULT]
hasThis
ReturnType: Void
1 Arguments
Argument #1: I4

TypeRef #4 (01000004)

Token: 0x01000004
ResolutionScope: 0x1a000001
TypeRefName: SecondModule.SecondModuleType
MemberRef #1 (0a000003)

Member: (0a000003) .ctor:
CallCnvtn: [DEFAULT]
hasThis
ReturnType: Void
No arguments.

MemberRef #2 (0a000004)

Member: (0a000004) Print:

CallCnvtn: [DEFAULT]
hasThis
ReturnType: String
No arguments.

TypeRef #5 (01000005)

Token: 0x01000005
ResolutionScope: 0x23000001
TypeRefName: System.Console
MemberRef #1 (0a000005)

Member: (0a000005) WriteLine:
CallCnvtn: [DEFAULT]
ReturnType: Void
1 Arguments
Argument #1: String

MemberRef #2 (0a000006)

Member: (0a000006) ReadLine:
CallCnvtn: [DEFAULT]
ReturnType: String
No arguments.

ModuleRef #1 (1a000001)

ModuleRef: (1a000001) SecondModule.netmodule:

Signature #1 (0x11000001)

CallCnvtn: [LOCALSIG]
2 Arguments
Argument #1: Class SecondModule.SecondModuleType
Argument #2: String

Assembly

Token: 0x20000001
Name : Result
Public Key :
Hash Algorithm : 0x00008004
Version: 0.0.0.0
Major Version: 0x00000000
Minor Version: 0x00000000
Build Number: 0x00000000
Revision Number: 0x00000000
Locale: <null>
Flags : [none] (00000000)
CustomAttribute #1 (0c000001)

CustomAttribute Type: 0a000001
CustomAttributeName: System.Runtime.CompilerServices.RuntimeCompatibilityAttribute :: instance void .ctor()
Length: 30
Value : 01 00 01 00 54 02 16 57 72 61 70 4e 6f 6e 45 78 > T WrapNonEx<
: 63 65 70 74 69 6f 6e 54 68 72 6f 77 73 01 >ceptionThrows <
ctor args: ()

CustomAttribute #2 (0c000002)

CustomAttribute Type: 0a000002
CustomAttributeName: System.Runtime.CompilerServices.CompilationRelaxationsAttribute :: instance void
.ctor(int32)
Length: 8
Value : 01 00 08 00 00 00 00 00 > <
ctor args: (8)

AssemblyRef #1 (23000001)

Token: 0x23000001

Public Key or Token: b7 7a 5c 56 19 34 e0 89
Name: mscorlib
Version: 4.0.0.0
Major Version: 0x00000004
Minor Version: 0x00000000
Build Number: 0x00000000
Revision Number: 0x00000000
Locale: <null>
HashValue Blob:
Flags: [none] (00000000)

File #1 (26000001)

Token: 0x26000001
Name : SecondModule.netmodule
HashValue Blob : 77 48 e0 d6 b6 b3 cb 3f 94 6b ff 7f db a2 32 82 15 07 93 ed
Flags : [ContainsMetaData] (00000000)

ExportedType #1 (27000001)

Token: 0x27000001
Name: SecondModule.SecondModuleType
Implementation token: 0x26000001
TypeDef token: 0x02000002
Flags : [Public] [AutoLayout] [Class] [AnsiClass] [BeforeFieldInit] (00100001)

User Strings

70000001 : (27) L"Hello from the FirstModule!"

Coff symbol name overhead: 0

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

6. Генерируем файл с публичным и приватным ключами:

```
C:\Users\Denis\Desktop>sn -k MyKeys.snk
```

7. Выделяем публичный ключ в отдельный файл:

```
C:\Users\Denis\Desktop>sn -p MyKeys.snk MyKeys.MyPublicKey
```

8. Смотрим публичный ключ:

```
C:\Users\Denis\Desktop>sn -tp MyKeys.MyPublicKey

Программа для работы со строгами именами Microsoft (R) .NET Framework, версия 4.0.30319.0
© Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Открытый ключ (алгоритм хэширования: sha1):
00240000048000000940000000602000000240000525341310004000001000100f17981c3825dc4
bf3aa3216069625ce20e4ed9afb7389e94203684fd81ee132334a6f696b682c5101bffbcb414223
5b5c2cedb02b9d682a02ed772ab24efe5fb1348bc6c6e9f4b1d31b78db8ac09df83acf23d57e8f
34364b21c4785e19e3c89d382791908f853b0955397673da90a461b7ef23a17e02271b59345eab
82f7fbc6

Токен открытого ключа: 36b085f3f5b9719e
```

9. Делаем общую подписанную сборку (сборку со строгим именем) SignedResult.exe:

```
C:\Users\Denis\Desktop>csc /out:SignedResult.exe /t:exe /addmodule:SecondModule.netmodule /keyfile:MyKeys.snk FirstModule.cs
```

10. Смотрим содержимое сборки SignedResult через ildasm.exe:

```
C:\Users\Denis\Desktop>ildasm SignedResult.exe
```

11. Видно, что сборка подписана:

```
// Metadata version: v4.0.30319
.module extern SecondModule.netmodule
.assembly extern nscorlib
{
    .publickeytoken = (B7 7A 5C 56 19 34 E0 89 ) // .z\U.4..
    .ver 4:0:0:0
}
.assembly SignedResult
{
    .custom instance void [mscorlib]System.Runtime.CompilerServices.RuntimeCompatibilityAttribute::.ctor() = ( 01 00 01 00 54 02 16 57 72 61 70 4E 6F 6E 45 78 // ....T..WrapNonEx
        63 65 70 74 69 6F 6E 54 68 72 6F 77 73 01 ) // ceptionThrows.
    .custom instance void [mscorlib]System.Runtime.CompilerServices.CompilationRelaxationsAttribute::.ctor(int32) = ( 01 00 00 00 00 00 00 00 )
    .publickey = (00 24 00 00 04 80 00 00 94 00 00 00 06 02 00 00 // $.RSA1.....
        00 24 00 00 52 53 41 31 00 04 00 00 01 00 01 00 // ..$..RSA1.....
        F1 79 81 C3 82 5D C4 BF 3A A3 21 60 69 62 5C E2 // .y...].:..t`ib\
        0E 4E D9 AF B7 38 9E 94 20 36 84 F0 81 EE 13 23 // .N...8.. 6....#
        34 A6 F6 96 B6 82 C5 10 1B FF BC 41 42 23 5B 5C // 4.....AB#[\
        2C ED B0 2B 9D 68 2A 02 ED 77 2A B2 4E FE 5F B1 // ,..+.h*..u*.N_..
        34 8B C6 C6 E9 F4 B1 D3 1B 78 DB 8A C0 9D F8 3A // 4.....x.....
        CF 23 D5 7E 8F 34 36 4B 21 C4 78 5E 19 E3 C8 9D // .#..~.46kt.x^....
        38 27 91 90 8F 85 38 09 55 39 76 73 DA 90 A4 61 // 8'.....U9vs...a
        B7 EF 23 A1 7E 02 27 1B 59 34 5E AB 82 F7 FB C6 ) // ..#..~.'V4^.....
    .hash algorithm 0x00000004
    .ver 0:0:0:0
}
.file SecondModule.netmodule
    .hash = (77 48 E0 D6 B6 B3 CB 3F 94 6B FF 7F DB A2 32 82 // uH.....?.k....2.
        15 07 93 ED )
.class extern public SecondModule.SecondModuleType
{
    .file SecondModule.netmodule
    .class 0x02000002
}
.module SignedResult.exe
// MVID: (2F8F0F1B-A0B3-4432-8DD1-45F3CF213B28)
.imagebase 0x00400000
.file alignment 0x00000200
.stackreserve 0x00100000
.subsystem 0x0003 // WINDOWS_CUI
.corFlags 0x00000009 // ILONLY
// Image base: 0x0C580000
```

12. Добавляем сборку SignedResult в GAC (глобальный кэш сборки):

```
C:\Users\Denis\Desktop>gacutil /i SignedResult.exe
Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.0
с Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Сборка успешно добавлена в кэш
```

13. Выводим список всех сборок в GAC:

```
C:\Users\Denis\Desktop>gacutil /l
```

14. Видно, что сборка SignedResult находится в GAC:

```
Renci.SshNet, Version=2020.0.0.0, Culture=neutral, PublicKeyToken=1cee9f8bde3db106, processorArchitecture=MSIL
SignedResult, Version=0.0.0.0, Culture=neutral, PublicKeyToken=36b085f3f5b9719e, processorArchitecture=MSIL
SMDiagnostics, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089, processorArchitecture=MSIL
```

15. Удаляем сборку SignedResult из GAC:

```
C:\Users\Denis\Desktop>gacutil /u SignedResult
Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.0
с Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Сборка: SignedResult, Version=0.0.0.0, Culture=neutral, PublicKeyToken=36b085f3f5b9719e, processorArchitecture=MSIL
Выполнено удаление: SignedResult, Version=0.0.0.0, Culture=neutral, PublicKeyToken=36b085f3f5b9719e, processorArchitecture=MSIL
Число удаленных сборок = 1
Число ошибок = 0
```