How to add a strong name to an existing DLL or assembly

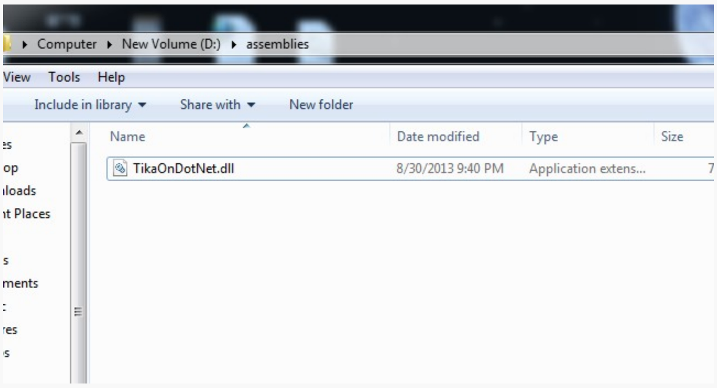
Published 8/30/2013 by Raghav Khunger

In this article I will show you how to add a strong name to an existing dll. The reason to sign an assembly is so ensure that it is your assembly. Since the private key is yours, no one else can sign that assembly with that same key. When the public key of an assembly is the one you know (the one which you original dll has), the assembly is yours and it has not been tampered. Let's add a strong name to an existing dll:

Basically we need to do the following 4 steps in order to add strong name to an existing dlls:

1. Generate a KeyFile.
2. Generate the MSIL for the assembly.
3. Rename the original assembly.
4. Generate the dll from the keyfile and MSIL for the assembly.

Before starting the above mentioned steps I would suggest you to put your assembly at following path "D:\\assemblies". The path can be different but for this article I will keep that path as a reference. For this example I will use TikaOnDotNet.dll for which I have to add strong name.



Computer ▸ New Volume (D:) ▸ assemblies
View Tools Help
Include in library ▾ Share with ▾ New folder

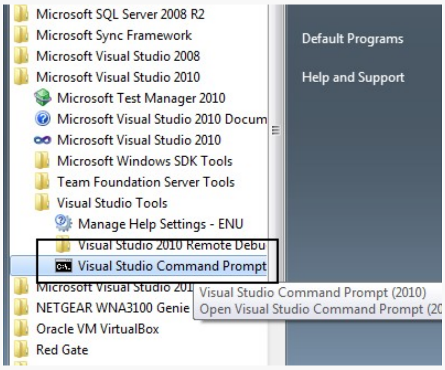
Name	Date modified	Type	Size
TikaOnDotNet.dll	8/30/2013 9:40 PM	Application extens...	7

Let's cover the above 4 points one by one.

Generate a keyfile

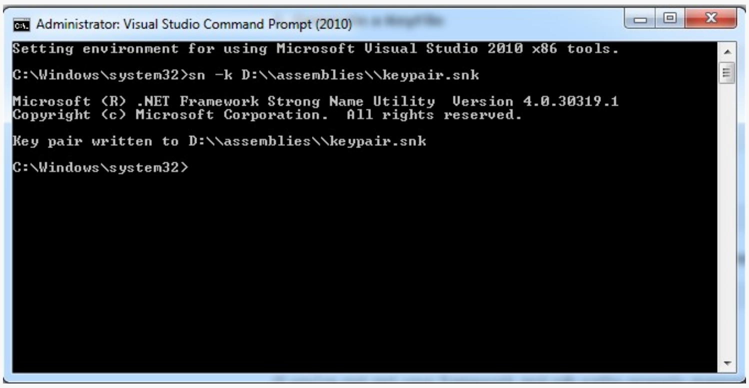
Open the Start >> All Programs >> Microsoft Visual Studio 2010 >> Visual Studio Tools >>

Visual Studio Command prompt. Open it with "Run as admin" privileges.


Microsoft SQL Server 2008 R2
Microsoft Sync Framework
Microsoft Visual Studio 2008
Microsoft Visual Studio 2010
Microsoft Test Manager 2010
Microsoft Visual Studio 2010 Docum
Microsoft Visual Studio 2010
Microsoft Windows SDK Tools
Team Foundation Server Tools
Visual Studio Tools
Manage Help Settings - ENU
Visual Studio 2010 Remote Debu
Visual Studio Command Prompt
Microsoft Visual Studio 201
NETGEAR WNA3100 Genie
Oracle VM VirtualBox
Red Gate
Visual Studio Command Prompt (2010)
Open Visual Studio Command Prompt (20
Default Programs
Help and Support

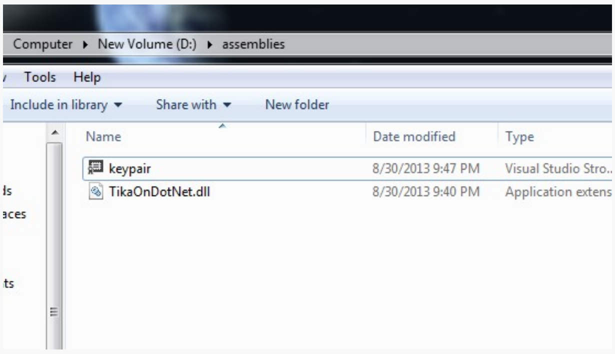
Run the following command to generate the keypair file:

sn -k D:\\assemblies\\keypair.snk



```
Setting environment for using Microsoft Visual Studio 2010 x86 tools.

C:\Windows\system32>sn -k D:\\assemblies\\keypair.snk

Microsoft (R) .NET Framework Strong Name Utility  Version 4.0.30319.1
Copyright (c) Microsoft Corporation.  All rights reserved.

Key pair written to D:\\assemblies\\keypair.snk

C:\Windows\system32>
```

Running the above command will create a new file named keypair.snk in D:\\assemblies folder.

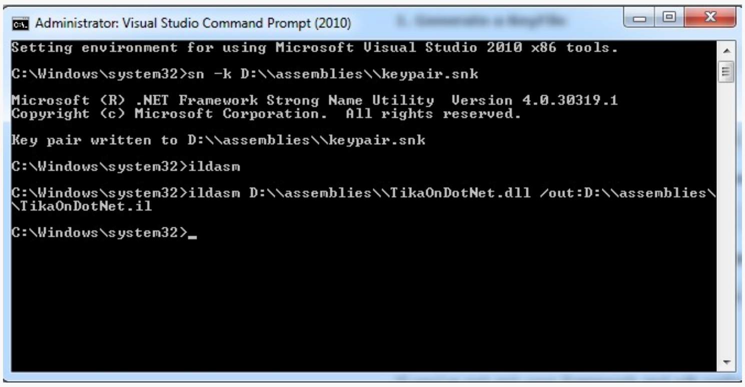


Name	Date modified	Type
keypair	8/30/2013 9:47 PM	Visual Studio Stro..
TikaOnDotNet.dll	8/30/2013 9:40 PM	Application extens

Generate the MSIL for the assembly

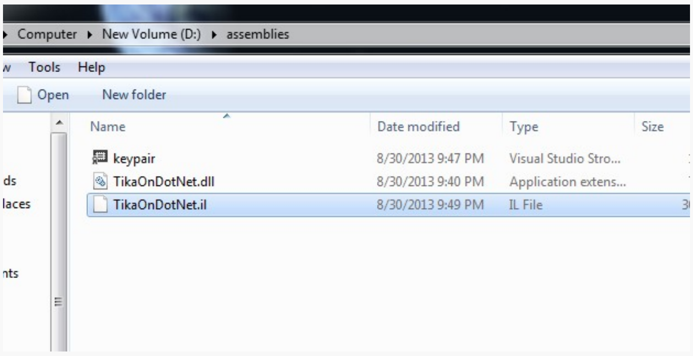
After creating the keypair.snk you need to generate the MSIL for your dll. For that run the following command:

ildasm D:\\assemblies\\TikaOnDotNet.dll /out:D:\\assemblies\\TikaOnDotNet.il



```
Setting environment for using Microsoft Visual Studio 2010 x86 tools.

C:\Windows\system32>sn -k D:\\assemblies\\keypair.snk

Microsoft (R) .NET Framework Strong Name Utility  Version 4.0.30319.1
Copyright (c) Microsoft Corporation.  All rights reserved.

Key pair written to D:\\assemblies\\keypair.snk

C:\Windows\system32>ildasm

C:\Windows\system32>ildasm D:\\assemblies\\TikaOnDotNet.dll /out:D:\\assemblies\
\TikaOnDotNet.il

C:\Windows\system32>_
```

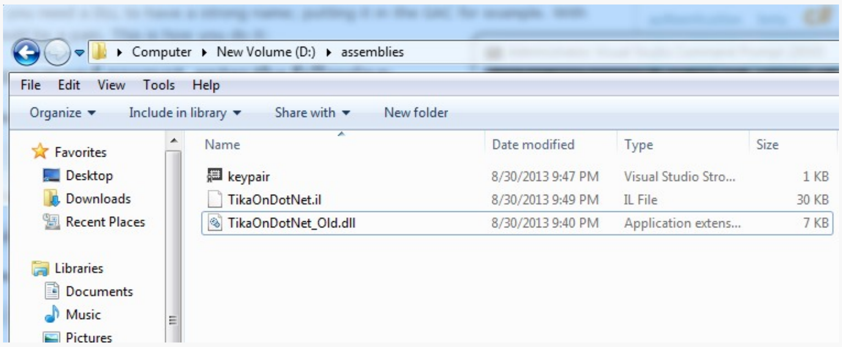
Running the above command will create a new file named TikaOnDotNet.il in D:\\assemblies folder.



Name	Date modified	Type	Size
keypair	8/30/2013 9:47 PM	Visual Studio Stro...	
TikaOnDotNet.dll	8/30/2013 9:40 PM	Application extens...	
TikaOnDotNet.il	8/30/2013 9:49 PM	IL File	3

Rename the original assembly

Since we need to create a new dll with same name at same location we should rename the old dll.


Computer ▸ New Volume (D:) ▸ assemblies
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures

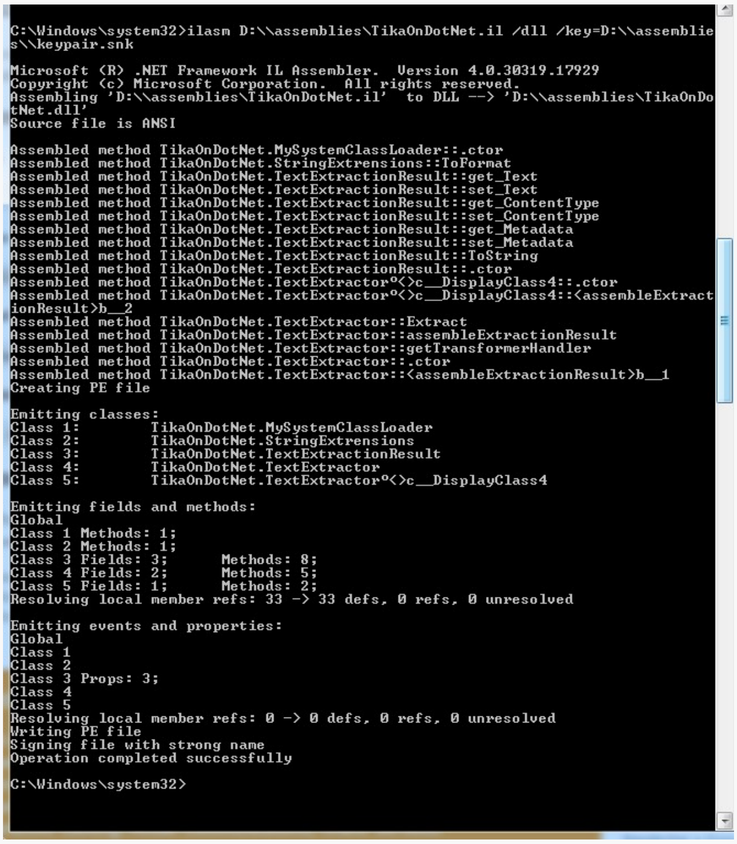
Name	Date modified	Type	Size
keypair	8/30/2013 9:47 PM	Visual Studio Stro...	1 KB
TikaOnDotNet.il	8/30/2013 9:49 PM	IL File	30 KB
TikaOnDotNet_Old.dll	8/30/2013 9:40 PM	Application extens...	7 KB

Above, I have renamed the TikaOnDotNet.dll to TikaOnDotNet_old.dll. This step is optional if you are generating the signed dll with same name at different location.

<u>Generate the dll from the keyfile and MSIL for the assmbly</u>

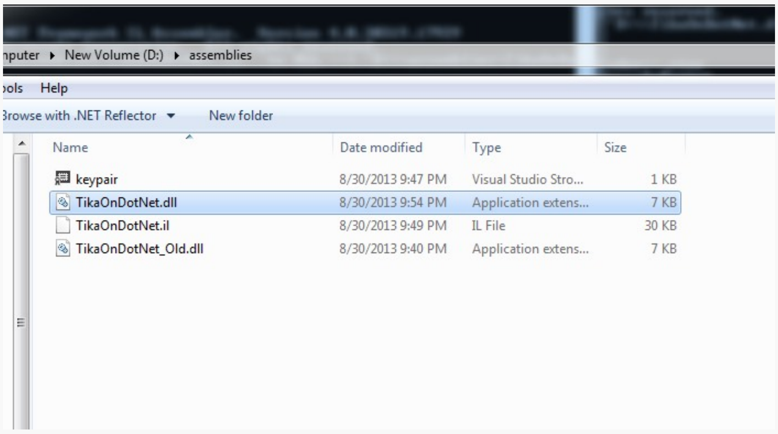
This is the final step where we will generate a new dll (strong name one) with the help of keypair and msil file we generated above. Run the following command:

ilasm D:\\assemblies\TikaOnDotNet.il /dll /key=D:\\assemblies\\keypair.snk



```
C:\Windows\system32>ilasm D:\\assemblies\TikaOnDotNet.il /dll /key=D:\\assemblie
s\\keypair.snk

Microsoft (R) .NET Framework IL Assembler.  Version 4.0.30319.17929
Copyright (c) Microsoft Corporation.  All rights reserved.
Assembling 'D:\\assemblies\TikaOnDotNet.il'  to DLL --> 'D:\\assemblies\TikaOnDo
tNet.dll'
Source file is ANSI

Assembled method TikaOnDotNet.MySystemClassLoader::.ctor
Assembled method TikaOnDotNet.StringExtrensions::ToFormat
Assembled method TikaOnDotNet.TextExtractionResult::get_Text
Assembled method TikaOnDotNet.TextExtractionResult::set_Text
Assembled method TikaOnDotNet.TextExtractionResult::get_ContentType
Assembled method TikaOnDotNet.TextExtractionResult::set_ContentType
Assembled method TikaOnDotNet.TextExtractionResult::get_Metadata
Assembled method TikaOnDotNet.TextExtractionResult::set_Metadata
Assembled method TikaOnDotNet.TextExtractionResult::ToString
Assembled method TikaOnDotNet.TextExtractionResult::.ctor
Assembled method TikaOnDotNet.TextExtractor°<>c__DisplayClass4::.ctor
Assembled method TikaOnDotNet.TextExtractor°<>c__DisplayClass4::<assembleExtract
ionResult>b__2
Assembled method TikaOnDotNet.TextExtractor::Extract
Assembled method TikaOnDotNet.TextExtractor::assembleExtractionResult
Assembled method TikaOnDotNet.TextExtractor::getTransformerHandler
Assembled method TikaOnDotNet.TextExtractor::.ctor
Assembled method TikaOnDotNet.TextExtractor::<assembleExtractionResult>b__1
Creating PE file

Emitting classes:
Class 1:        TikaOnDotNet.MySystemClassLoader
Class 2:        TikaOnDotNet.StringExtrensions
Class 3:        TikaOnDotNet.TextExtractionResult
Class 4:        TikaOnDotNet.TextExtractor
Class 5:        TikaOnDotNet.TextExtractor°<>c__DisplayClass4

Emitting fields and methods:
Global
Class 1 Methods: 1;
Class 2 Methods: 1;
Class 3 Fields: 3;      Methods: 8;
Class 4 Fields: 2;      Methods: 5;
Class 5 Fields: 1;      Methods: 2;
Resolving local member refs: 33 -> 33 defs, 0 refs, 0 unresolved

Emitting events and properties:
Global
Class 1
Class 2
Class 3 Props: 3;
Class 4
Class 5
Resolving local member refs: 0 -> 0 defs, 0 refs, 0 unresolved
Writing PE file
Signing file with strong name
Operation completed successfully

C:\Windows\system32>
```

After running the above command a new file named TikaOnDotNet.dll is generated. This dll will have strong name.


nputer ▸ New Volume (D:) ▸ assemblies
ools Help
Browse with .NET Reflector ▾
New folder

Name	Date modified	Type	Size
keypair	8/30/2013 9:47 PM	Visual Studio Stro...	1 KB
TikaOnDotNet.dll	8/30/2013 9:54 PM	Application extens...	7 KB
TikaOnDotNet.il	8/30/2013 9:49 PM	IL File	30 KB
TikaOnDotNet_Old.dll	8/30/2013 9:40 PM	Application extens...	7 KB

Summary:

Here is the summary of command which you need to run to add strong name to an existing dll:

1. sn -k keypair.snk
2. ildasm assembly.dll /out:assembly.il
3. ilasm assembly.il /dll /key= keypair.snk

tags : assembly-signing, dll