This page is specific to

Microsoft Visual Studio 2010/.NET Framework 4

.NET Framework 4

**How to: Create a Public/Private Key Pair**

To sign an assembly with a strong name, you must have a public/private key pair. This public and private cryptographic key pair is used during compilation to create a strong-named assembly. You can create a key pair using the [Strong Name tool (Sn.exe)](http://msdn.microsoft.com/en-us/library/k5b5tt23.aspx). Key pair files usually have an .snk extension.

|  |
| --- |
| **Description: NoteNote** |
| In Visual Studio, the C# and Visual Basic project property pages include a **Signing** tab that enables you to select existing key files or to generate new key files without using Sn.exe. In Visual C++, you can specify the location of an existing key file in the **Advanced** property page in the **Linker** section of the **Configuration Properties** section of the **Property Pages** window. The use of the [AssemblyKeyFileAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblykeyfileattribute.aspx) attribute to identify key file pairs has been made obsolete beginning with Visual Studio 2005. |

**To create a key pair**

* At the command prompt, type the following command:

**sn –k** <*file name*>

In this command, *file name* is the name of the output file containing the key pair.

The following example creates a key pair called sgKey.snk.

sn -k sgKey.snk

If you intend to delay sign an assembly and you control the whole key pair (which is unlikely outside test scenarios), you can use the following commands to generate a key pair and then extract the public key from it into a separate file. First, create the key pair:

sn -k keypair.snk

* Next, extract the public key from the key pair and copy it to a separate file:

sn -p keypair.snk public.snk

* Once you create the key pair, you must put the file where the strong name signing tools can find it.

When signing an assembly with a strong name, the [Assembly Linker (Al.exe)](http://msdn.microsoft.com/en-us/library/c405shex.aspx) looks for the key file relative to the current directory and to the output directory. When using command-line compilers, you can simply copy the key to the current directory containing your code modules.

If you are using an earlier version of Visual Studio that does not have a **Signing** tab in the project properties, the recommended key file location is the project directory with the file attribute specified as follows:

Visual Basic

<Assembly:AssemblyKeyFileAttribute("keyfile.snk")>

C#

[assembly:AssemblyKeyFileAttribute("keyfile.snk")]

Visual C++

[assembly:AssemblyKeyFileAttribute("keyfile.snk")];