**Sn.exe (Strong Name Tool)**

**.NET Framework 4.5**

The Strong Name tool (Sn.exe) helps sign assemblies with [strong names](http://msdn.microsoft.com/en-us/library/wd40t7ad.aspx). Sn.exe provides options for key management, signature generation, and signature verification.

The Strong Name tool is automatically installed with Visual Studio. To start the tool, use the [Visual Studio and Windows SDK Command Prompts](http://msdn.microsoft.com/en-us/library/ms229859.aspx).

|  |
| --- |
| **Description: NoteNote** |
| On 64-bit computers, run both the 32-bit version of Sn.exe by using the Visual Studio Command Prompt and the 64-bit version by using the Visual Studio x64 Win64 Command Prompt. |

This tool is automatically installed with Visual Studio and with the Windows SDK. To run the tool, we recommend that you use the Visual Studio Command Prompt or the Windows SDK Command Prompt (CMD Shell). These utilities enable you to run the tool easily, without navigating to the installation folder. For more information, see [Visual Studio and Windows SDK Command Prompts](http://msdn.microsoft.com/en-us/library/ms229859.aspx).

* If you have Visual Studio installed on your computer: On the taskbar, click **Start**, click **All Programs**, click **Visual Studio**, click **Visual Studio Tools**, and then click **Visual Studio Command Prompt**.

-or-

If you have the Windows SDK installed on your computer: On the taskbar, click **Start**, click **All Programs**, click the folder for the Windows SDK, and then click **Command Prompt** (or **CMD Shell**).

* At the command prompt, type the following:

sn [-quiet][option [parameter(s)]]

[**Parameters**](javascript:void(0))

|  |  |
| --- | --- |
| Option | Description |
| **-a** *identityKeyPairFile* *signaturePublicKeyFile* | Generates [AssemblySignatureKeyAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblysignaturekeyattribute.aspx) data to migrate the identity key to the signature key from a file. |
| **-ac** *identityPublicKeyFile* *identityKeyPairContainer* *signaturePublicKeyFile* | Generates [AssemblySignatureKeyAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblysignaturekeyattribute.aspx) data to migrate the identity key to the signature key from a key container. |
| **-c** [*csp*] | Sets the default cryptographic service provider (CSP) to use for strong name signing. This setting applies to the entire computer. If you do not specify a CSP name, Sn.exe clears the current setting. |
| **-d** *container* | Deletes the specified key container from the strong name CSP. |
| **-D** *assembly1 assembly2* | Verifies that two assemblies differ only by signature. This is often used as a check after an assembly has been re-signed with a different key pair. |
| **-e** *assembly outfile* | Extracts the public key from *assembly* and stores it in *outfile.* |
| **-h** | Displays command syntax and options for the tool. |
| **-i** *infile container* | Installs the key pair from *infile* in the specified key container. The key container resides in the strong name CSP. |
| **-k** [*keysize*] *outfile* | Generates a new [RSACryptoServiceProvider](http://msdn.microsoft.com/en-us/library/system.security.cryptography.rsacryptoserviceprovider.aspx) key of the specified size and writes it to the specified file. Both a public and private key are written to the file.  If you do not specify a key size, a 1,024-bit key is generated by default if you have the Microsoft enhanced cryptographic provider installed; otherwise, a 512-bit key is generated.  The *keysize* parameter supports key lengths from 384 bits to 16,384 bits in increments of 8 bits if you have the Microsoft enhanced cryptographic provider installed. It supports key lengths from 384 bits to 512 bits in increments of 8 bits if you have the Microsoft base cryptographic provider installed. |
| **-m** [**y***|***n**] | Specifies whether key containers are computer-specific, or user-specific. If you specify *y*, key containers are computer-specific. If you specify *n*, key containers are user-specific.  If neither y nor n is specified, this option displays the current setting. |
| **-o** *infile* [*outfile*] | Extracts the public key from the *infile* and stores it in a .csv file. A comma separates each byte of the public key. This format is useful for hard-coding references to keys as initialized arrays in source code. If you do not specify an *outfile*, this option places the output on the Clipboard.   |  | | --- | | **Description: NoteNote** | | This option does not verify that the input is only a public key. If the *infile* contains a key pair with a private key, the private key is also extracted. | |
| **-p** *infile outfile* [*hashalg*] | Extracts the public key from the key pair in *infile* and stores it in *outfile*, optionally using the RSA algorithm specified by *hashalg*. This public key can be used to delay-sign an assembly using the **/delaysign+** and **/keyfile** options of the [Assembly Linker (Al.exe)](http://msdn.microsoft.com/en-us/library/c405shex.aspx). When an assembly is delay-signed, only the public key is set at compile time and space is reserved in the file for the signature to be added later, when the private key is known. |
| **-pc** *container* *outfile* [*hashalg*] | Extracts the public key from the key pair in *container* and stores it in *outfile*. If you use the *hashalg* option, the RSA algorithm is used to extract the public key. |
| **-Pb** [**y***|***n**] | Specifies whether the strong-name bypass policy is enforced. If you specify *y*, strong names for full-trust assemblies are not validated when loaded into a full-trust [AppDomain](http://msdn.microsoft.com/en-us/library/system.appdomain.aspx). If you specify *n*, strong names are validated for correctness, but not for a specific strong name. The [StrongNameIdentityPermission](http://msdn.microsoft.com/en-us/library/system.security.permissions.strongnameidentitypermission.aspx) has no effect on full-trust assemblies. You must perform your own check for a strong name match.  If neither *y* nor *n* is specified, this option displays the current setting. The default is *y*.   |  | | --- | | **Description: NoteNote** | | On 64-bit computers, you must set this parameter in both the 32-bit and the 64-bit instances of Sn.exe. | |
| **-q**[**uiet**] | Specifies quiet mode; suppresses the display of success messages. |
| **-R**[**a**] *assembly* *infile* | Re-signs a previously signed or delay-signed assembly with the key pair in *infile*.  If **-Ra** is used, hashes are recomputed for all files in the assembly. |
| **-Rc**[**a**] *assembly container* | Re-signs a previously signed or delay-signed assembly with the key pair in *container*.  If **-Rca** is used, hashes are recomputed for all files in the assembly. |
| **-Rh** *assembly* | Recomputes hashes for all files in the assembly. |
| **-t**[**p**] *infile* | Displays the token for the public key stored in *infile*. The contents of *infile* must be a public key previously generated from a key pair file using **-p**. Do not use the **-t[p]** option to extract the token directly from a key pair file.  Sn.exe computes the token by using a hash function from the public key. To save space, the common language runtime stores public key tokens in the manifest as part of a reference to another assembly when it records a dependency to an assembly that has a strong name. The **-tp** option displays the public key in addition to the token. If the [AssemblySignatureKeyAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblysignaturekeyattribute.aspx) attribute has been applied to the assembly, the token is for the identity key, and the name of the hash algorithm and the identity key is displayed.  Note that this option does not verify the assembly signature and should not be used to make trust decisions. This option only displays the raw public key token data. |
| **-T**[**p**] *assembly* | Displays the public key token for *assembly.* The *assembly* must be the name of a file that contains an assembly manifest.  Sn.exe computes the token by using a hash function from the public key. To save space, the runtime stores public key tokens in the manifest as part of a reference to another assembly when it records a dependency to an assembly that has a strong name. The **-Tp** option displays the public key in addition to the token. If the [AssemblySignatureKeyAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblysignaturekeyattribute.aspx) attribute has been applied to the assembly, the token is for the identity key, and the name of the hash algorithm and the identity key is displayed.  Note that this option does not verify the assembly signature and should not be used to make trust decisions. This option only displays the raw public key token data. |
| **-TS***assembly* *infile* | Test-signs the signed or partially signed *assembly* with the key pair in *infile*. |
| -**TSc** *assembly* *container* | Test-signs the signed or partially signed *assembly* with the key pair in the key container *container*. |
| **-v** *assembly* | Verifies the strong name in *assembly*, where *assembly* is the name of a file that contains an assembly manifest. |
| **-vf** *assembly* | Verifies the strong name in *assembly.* Unlike the **-v** option, **-vf** forces verification even if it is disabled using the **-Vr** option. |
| **-Vk** *regfile.regassembly* [*userlist*] [*infile*] | Creates a registration entries (.reg) file you can use to register the specified assembly for verification skipping. The rules for assembly naming that apply to the **-Vr** option apply to **–Vk** as well. For information about the *userlist* and *infile* options, see the **–Vr** option. |
| **-Vl** | Lists current settings for strong-name verification on this computer. |
| **-Vr** *assembly* [*userlist*] [*infile*] | Registers *assembly* for verification skipping. Optionally, you can specify a comma-separated list of user names the skip verification should apply to. If you specify *infile*, verification remains enabled, but the public key in *infile* is used in verification operations. You can specify *assembly* in the form *\*, strongname* to register all assemblies with the specified strong name. For *strongname*, specify the string of hexadecimal digits representing the tokenized form of the public key. See the **-t** and **-T** options to display the public key token.   |  | | --- | | **Description: Caution noteCaution** | | Use this option only during development. Adding an assembly to the skip verification list creates a security vulnerability. A malicious assembly could use the fully specified assembly name (assembly name, version, culture, and public key token) of the assembly added to the skip verification list to fake its identity. This would allow the malicious assembly to also skip verification. | |
|  |  |
| **-Vu** *assembly* | Unregisters *assembly* for verification skipping. The same rules for assembly naming that apply to **-Vr** apply to **-Vu**. |
| **-Vx** | Removes all verification-skipping entries. |
| **-?** | Displays command syntax and options for the tool. |
| **Description: NoteNote** | |
| All Sn.exe options are case-sensitive and must be typed exactly as shown to be recognized by the tool. | |

[**Remarks**](javascript:void(0))

The **-R** and **–Rc** options are useful with assemblies that have been delay-signed. In this scenario, only the public key has been set at compile time and signing is performed later, when the private key is known.

|  |
| --- |
| **Description: NoteNote** |
| For parameters (for example, –**Vr)** that write to protected resources such as the registry, run SN.exe as an administrator. |

[**Examples**](javascript:void(0))

The following command creates a new, random key pair and stores it in keyPair.snk.

sn -k keyPair.snk

The following command stores the key in keyPair.snk in the container MyContainer in the strong name CSP.

sn -i keyPair.snk MyContainer

The following command extracts the public key from keyPair.snk and stores it in publicKey.snk.

sn -p keyPair.snk publicKey.snk

The following command displays the public key and the token for the public key contained in publicKey.snk.

sn -tp publicKey.snk

The following command verifies the assembly MyAsm.dll.

sn -v MyAsm.dll

The following command deletes MyContainer from the default CSP.

sn -d MyContainer