**SignTool.exe (Sign Tool)**

**.NET Framework 4.5**

Sign Tool is a command-line tool that digitally signs files, verifies signatures in files, and time-stamps files.

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:

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_f062b449-eeae-4376-b7b9-f19a076e953f');" \o "Copy to clipboard.)

signtool [command] [options] [file\_name | ...]

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

|  |  |
| --- | --- |
| Argument | Description |
| *command* | One of four commands (**catdb**, **sign**, **Timestamp**, or **Verify**) that specifies an operation to perform on a file. For a description of each command, see the next table. |
| *options* | An option that modifies a command. In addition to the global **/q** and **/v** options, each command supports a unique set of options. |
| *file\_name* | The path to a file to sign. |

The following commands are supported by Sign Tool. Each command is used with distinct set of options, which are listed in their respective sections.

|  |  |
| --- | --- |
| Command | Description |
| **catdb** | Adds a catalog file to, or removes it from, a catalog database. Catalog databases are used for automatic lookup of catalog files and are identified by GUID. For a list of the options supported by the **catdb** command, see [catdb Command Options](http://msdn.microsoft.com/en-us/library/8s9b9yaz.aspx#catdb). |
| **sign** | Digitally signs files. Digital signatures protect files from tampering, and enable users to verify the signer based on a signing certificate. For a list of the options supported by the **sign** command, see [sign Command Options](http://msdn.microsoft.com/en-us/library/8s9b9yaz.aspx#sign). |
| **Timestamp** | Time-stamps files. For a list of the options supported by the **TimeStamp** command, see [TimeStamp Command Options](http://msdn.microsoft.com/en-us/library/8s9b9yaz.aspx#timestamp). |
| **Verify** | Verifies the digital signature of files by determining whether the signing certificate was issued by a trusted authority, whether the signing certificate has been revoked, and, optionally, whether the signing certificate is valid for a specific policy. For a list of the options supported by the **Verify** command, see [Verify Command Options](http://msdn.microsoft.com/en-us/library/8s9b9yaz.aspx#verify). |

The following options apply to all Sign Tool commands.

|  |  |
| --- | --- |
| Global option | Description |
| **/q** | No output on successful execution and minimal output for failed execution. |
| **/v** | Verbose output for successful execution, failed execution, and warning messages. |

[**catdb Command Options**](javascript:void(0))

The following table lists the options that can be used with the **catdb** command.

|  |  |
| --- | --- |
| Catdb option | Description |
| **/d** | Specifies that the default catalog database is updated. If neither the **/d** nor the **/g** option is used, Sign Tool updates the system component and driver database. |
| **/g** *GUID* | Specifies that the catalog database identified by the globally unique identifier *GUID* is updated. |
| **/r** | Removes the specified catalogs from the catalog database. If this option is not specified, Sign Tool adds the specified catalogs to the catalog database. |
| **/u** | Specifies that a unique name is automatically generated for the added catalog files. If necessary, the catalog files are renamed to prevent name conflicts with existing catalog files. If this option is not specified, Sign Tool overwrites any existing catalog that has the same name as the catalog being added. |

[**sign Command Options**](javascript:void(0))

The following table lists the options that can be used with the **sign** command.

|  |  |
| --- | --- |
| Sign command option | Description |
| **/a** | Automatically selects the best signing certificate. Sign Tool will find all valid certificates that satisfy all specified conditions and select the one that is valid for the longest time. If this option is not present, Sign Tool expects to find only one valid signing certificate. |
| **/ac** *file* | Adds an additional certificate from *file* to the signature block. |
| **/c** *CertTemplateName* | Specifies the Certificate Template Name (a Microsoft extension) for the signing certificate. |
| **/csp** *CSPName* | Specifies the cryptographic service provider (CSP) that contains the private key container. |
| **/d** *Desc* | Specifies a description of the signed content. |
| **/du** *URL* | Specifies a Uniform Resource Locator (URL) for the expanded description of the signed content. |
| **/f** *SignCertFile* | Specifies the signing certificate in a file. If the file is in Personal Information Exchange (PFX) format and protected by a password, use the **/p** option to specify the password. If the file does not contain private keys, use the **/csp** and **/k** options to specify the CSP and private key container name. |
| **/fd** | Specifies the file digest algorithm to use for creating file signatures. The default is SHA1. |
| **/i** *IssuerName* | Specifies the name of the issuer of the signing certificate. This value can be a substring of the entire issuer name. |
| **/kc** *PrivKeyContainerName* | Specifies the private key container name. |
| **/n** *SubjectName* | Specifies the name of the subject of the signing certificate. This value can be a substring of the entire subject name. |
| **/nph** | If supported, suppresses page hashes for executable files. The default is determined by the SIGNTOOL\_PAGE\_HASHES environment variable and by the wintrust.dll version. |
| **/p** *Password* | Specifies the password to use when opening a PFX file. (Use the **/f** option to specify a PFX file.) |
| **/ph** | If supported, generates page hashes for executable files. |
| **/r** *RootSubjectName* | Specifies the name of the subject of the root certificate that the signing certificate must chain to. This value may be a substring of the entire subject name of the root certificate. |
| **/s** *StoreName* | Specifies the store to open when searching for the certificate. If this option is not specified, the My store is opened. |
| **/sha1** *Hash* | Specifies the SHA1 hash of the signing certificate. |
| **/sm** | Specifies that a machine store, instead of a user store, is used. |
| **/t** *URL* | Specifies the URL of the time stamp server. If this option (or **/tr**) is not present, the signed file will not be time stamped. A warning is generated if time stamping fails. This option cannot be used with the **/tr** option. |
| **/td** *alg* | Used with the **/tr** option to request a digest algorithm used by the RFC 3161 time stamp server. |
| **/tr** *URL* | Specifies the URL of the RFC 3161 time stamp server. If this option (or **/t**) is not present, the signed file will not be time stamped. A warning is generated if time stamping fails. This option cannot be used with the **/t** option. |
| **/u** *Usage* | Specifies the enhanced key usage (EKU) that must be present in the signing certificate. The usage value can be specified by OID or string. The default usage is "Code Signing" (1.3.6.1.5.5.7.3.3). |
| **/uw** | Specifies usage of "Windows System Component Verification" (1.3.6.1.4.1.311.10.3.6). |

[**TimeStamp Command Options**](javascript:void(0))

The following table lists the options that can be used with the **TimeStamp** command.

|  |  |
| --- | --- |
| TimeStamp option | Description |
| **/t** *URL* | Specifies the URL of the time stamp server. The file being time stamped must have previously been signed. Either the **/t** or the **/tr** option is required. |
| **/td** *alg* | Requests a digest algorithm used by the RFC 3161 time stamp server. **/td** is used with the **/tr** option. |
| **/tr** *URL* | Specifies the URL of the RFC 3161 time stamp server. The file being time stamped must have previously been signed. Either the **/tr** or the **/t** option is required. |

[**Verify Command Options**](javascript:void(0))

|  |  |
| --- | --- |
| Verify option | Description |
| **/a** | Specifies that all methods can be used to verify the file. First, the catalog databases are searched to determine whether the file is signed in a catalog. If the file is not signed in any catalog, Sign Tool attempts to verify the file's embedded signature. This option is recommended when verifying files that may or may not be signed in a catalog. Examples of these files include Windows files or drivers. |
| **/ad** | Finds the catalog by using the default catalog database. |
| **/as** | Finds the catalog by using the system component (driver) catalog database. |
| **/ag** *CatDBGUID* | Finds the catalog in the catalog database that is identified by the *CatDBGUID*. |
| **/c** *CatFile* | Specifies the catalog file by name. |
| **/d** | Specifies that Sign Tool should print the description and the description URL. |
| /**kp** | Specifies that verification should be performed with the kernel-mode driver signing policy. |
| **/o** *Version* | Verifies the file by operating system version. *Version* has the following form: *PlatformID*:*VerMajor*.*VerMinor*.*BuildNumber* |
| **/pa** | Specifies that the Default Authenticode Verification Policy should be used. If the **/pa** option is not specified, Sign Tool uses the Windows Driver Verification Policy. This option cannot be used with the **catdb** options. |
| **/pg** *PolicyGUID* | Specifies a verification policy by GUID. The *PolicyGUID* corresponds to the ActionID of the verification policy. This option cannot be used with the **catdb** options. |
| **/ph** | Specifies that Sign Tool should print and verify page hash values. |
| **/r** *RootSubjectName* | Specifies the name of the subject of the root certificate that the signing certificate must chain to. This value can be a substring of the entire subject name of the root certificate. |
| **/tw** | Specifies that a warning should be generated if the signature is not time stamped. |

[**Return Value**](javascript:void(0))

Sign Tool returns one of the following exit codes when it terminates.

|  |  |
| --- | --- |
| Exit code | Description |
| 0 | Execution was successful. |
| 1 | Execution has failed. |
| 2 | Execution has completed with warnings. |

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

The following command adds the catalog file MyCatalogFileName.cat to the system component and driver database. The **/v** option generates a unique name if necessary to prevent replacing an existing catalog file named MyCatalogFileName.cat.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_836470b5-d7ad-46e6-aeee-7ee70e11f133');)

signtool catdb /v /u MyCatalogFileName.cat

The following command signs a file automatically by using the best certificate.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_19082c66-3dc0-4d21-8fb9-eb6dea80ba96');)

signtool sign /a MyFile.exe

The following command digitally signs a file by using a certificate stored in a password-protected PFX file.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_291999b9-0b79-4093-8779-b50ffb49b622');)

signtool sign /f MyCert.pfx /p MyPassword MyFile.exe

The following command digitally signs and time-stamps a file. The certificate used to sign the file is stored in a PFX file.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_e74a8e59-0ab9-45d0-9186-973a138cad21');)

signtool sign /f MyCert.pfx /t http://timestamp.verisign.com/scripts/timstamp.dll MyFile.exe

The following command signs a file by using a certificate located in the My store that has a subject name of My Company Certificate.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_37eaed29-9ec4-42ee-9fb9-db36b9ff641a');)

signtool sign /n "My Company Certificate" MyFile.exe

The following command signs an ActiveX control and provides information that is displayed by Internet Explorer when the user is prompted to install the control.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_0d353bba-3991-4c0b-989e-bbfecb6a1e79');)

Signtool sign /f MyCert.pfx /d: "MyControl" /du http://www.example.com/MyControl/info.html MyControl.exe

The following command time-stamps a file that has already been digitally signed.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_d2254c75-ac98-42e3-9949-bad41a3e45a7');)

signtool timestamp /t http://timestamp.verisign.com/scripts/timstamp.dll MyFile.exe

The following command verifies that a file has been signed.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_932f433d-5cc8-4128-a334-263cd5fe1c95');)

signtool verify MyFile.exe

The following command verifies a system file that may be signed in a catalog.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_a4f8c62e-e896-49cb-9706-52244f6d2493');)

signtool verify /a SystemFile.dll

The following command verifies a system file that is signed in a catalog named MyCatalog.cat.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_9577a66f-61ca-4097-b88d-dd74d78ec3a5');)

signtool verify /c MyCatalog.cat SystemFile.dll