**Al.exe (Assembly Linker)**

**.NET Framework 4.5**

The Assembly Linker generates a file that has an assembly manifest from one or more files that are either modules or resource files. A module is a Microsoft intermediate language (MSIL) file that does not have an assembly manifest.

|  |
| --- |
| **Description: NoteNote** |
| To avoid being subject to virtualization on a Windows Vista computer, your assembly must include a win32 manifest that specifies a requested execution level. When using al.exe directly from the command line, you can either embed the manifest in a win32 resource file or else use mt.exe to append the manifest at a later stage in the build process. In Visual Studio 2008 both the C# and Visual Basic compilers automatically embed a win32 manifest into the assembly. For more information, see [/win32manifest (C# Compiler Options)](http://msdn.microsoft.com/en-us/library/bb545961.aspx). |

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_700c40e3-525e-4aef-a70b-8c16a8b78525');" \o "Copy to clipboard.)

al sources options

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

You can specify one or more of the following *sources*.

|  |  |
| --- | --- |
| Source | Description |
| *file*[,*target*] | Copies the contents of *file* (a module) to the file name specified by *target*. After copying, Al.exe compiles *target* into an assembly. |
| **/embed**[**resource**]**:***file*[,*name*[,**private**]] | Embeds the resource specified by *file* in the image that contains the assembly manifest; Al.exe copies the contents of *file* into the portable executable (PE) image.  The *name* parameter is an internal identifier for the resource. By default, resources are public in the assembly (visible to other assemblies). Specifying **private** makes the resource not visible to other assemblies.  If *file* is a .NET Framework resource file created, for example, by the [Resource File Generator (Resgen.exe)](http://msdn.microsoft.com/en-us/library/ccec7sz1.aspx) or in the development environment, it can be accessed with members in the [System.Resources](http://msdn.microsoft.com/en-us/library/system.resources.aspx). For more information, see [ResourceManager](http://msdn.microsoft.com/en-us/library/system.resources.resourcemanager.aspx). For all other resources, use the **GetManifestResource**\* methods in the [Assembly](http://msdn.microsoft.com/en-us/library/system.reflection.assembly.aspx) to access the resource at run time.  If only resource files are passed to Al.exe, the output file is a satellite resource assembly. |
| **/link**[**resource**]:*file*[,*name*[,*target*[,**private**]]] | Links a resource file to an assembly. The resource specified by *file* becomes part of the assembly; the file is not copied. The *file* parameter can be in any file format. For example, you can specify a native DLL as the *file* parameter. This will make the native DLL part of the assembly so that it can be installed into the global assembly cache and accessed from managed code in the assembly. You can also do this by using the **/linkresource** compiler option. For more information, see [/linkresource (C# Compiler Options)](http://msdn.microsoft.com/en-us/library/xawyf94k.aspx).  The*name* parameter is an internal identifier for the resource. The *target*parameter specifies a path and file name into which Al.exe copies the *file.* After copying, Al.exe compiles *target* into an assembly. By default, resources are public in the assembly (visible to other assemblies). Specifying **private** makes the resource not visible to other assemblies.  If *file* is a .NET Framework resource file created, for example, by the Resource File Generator (Resgen.exe) or in the development environment, it can be accessed with members in the [System.Resources](http://msdn.microsoft.com/en-us/library/system.resources.aspx) namespace. For more information, see [ResourceManager](http://msdn.microsoft.com/en-us/library/system.resources.resourcemanager.aspx). For all other resources, use the **GetManifestResource** \* methods in the [Assembly](http://msdn.microsoft.com/en-us/library/system.reflection.assembly.aspx) class to access the resource at run time.  If only resource files are passed to Al.exe, the output file is a satellite resource assembly. |

You can specify the following *options*; you must specify **/out**.

|  |  |
| --- | --- |
| Option | Description |
| **/algid**:*id* | Specifies an algorithm to hash all files in a multifile assembly except the file that contains the assembly manifest. The default algorithm is CALG\_SHA1. See ALG\_ID in the Platform SDK documentation for other algorithms. For the first release of the .NET Framework, only CALG\_SHA1 and CALG\_MD5 are valid.  The hash values are stored in the file table of the assembly manifest. At installation and load time, the assembly's files are checked against their hashes.  You can also specify this option as a custom attribute ([AssemblyAlgorithmIdAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblyalgorithmidattribute.aspx)) in the source code for any module. |
| **/base**[**address**]:*addr* | Specifies the address at which a DLL will be loaded on the user's computer at run time. Applications load faster if you specify the base address of the DLLs, instead of letting the operating system relocate the DLLs in the process space. |
| **/bugreport**:*filename* | Creates a file (*filename*) that contains information for reporting bugs. |
| **/comp**[**any**]:*text* | Specifies a string for the Company field in the assembly. Place the string in double quotation marks (" ") if *text* contains a space. This string is a custom attribute on the assembly and is available for viewing with reflection.  If you do not specify **/win32res**, *text* appears in File Explorer as the **Company** property for the file. If you specify **/win32res**, the company information in the specified resource file appears as the **Company** property in File Explorer.  If text is an empty string (""), the Win32 **Company** resource appears as a single space.  If you specify **/win32res**, **/company** will not affect the Win32 resource information.  You can also specify this option as a custom attribute ([AssemblyCompanyAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblycompanyattribute.aspx)) in the source code for any MSIL module. |
| **/config**[**uration**]:*text* | Specifies a string for the Configuration field in the assembly. Place the string in double quotation marks (" ") if *text* contains a space. This string is a custom attribute on the assembly and is available for viewing with reflection.  If text is an empty string, the Win32 Configuration resource appears as a single space.  You can also specify this option as a custom attribute ([AssemblyConfigurationAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblyconfigurationattribute.aspx)) in the source code for any MSIL module. |
| **/copy**[**right**]:*text* | Specifies a string for the Copyright field in the assembly. Place the string in double quotation marks (" ") if *text* contains a space. This string is a custom attribute on the assembly and is available for viewing with reflection.  If you do not specify **/win32res**, **/copyright** appears in File Explorer as the Win32 Copyright resource.  If text is an empty string, the Win32 Copyright resource appears as a single space.  If you specify **/win32res**, **/copyright** will not affect the Win32 resource information.  You can also specify this option as a custom attribute ([AssemblyCopyrightAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblycopyrightattribute.aspx)) in the source code for any MSIL module. |
| **/c**[**ulture**]:*text* | Specifies the culture string to associate with the assembly. Valid values for cultures are those defined by the Internet Requests for Comments (RFC) document 1766 titled "Tags for the Identification of Languages."  Place the string in double quotation marks (" ") if *text* contains a space. There is no default culture string. This string is available for viewing with reflection.  For information about valid *text* strings, see the [CultureInfo](http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo.aspx).  You can also specify this option as a custom attribute ([AssemblyCultureAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblycultureattribute.aspx)) in the source code for any MSIL module. |
| **/delay**[**sign**][**+|-**] | Specifies whether the assembly will be fully or partially signed. Use **/delaysign-** if you want a fully signed assembly. Use **/delaysign+** if you only want to include the public key in the assembly.  When you request a fully signed assembly, Al.exe hashes the file that contains the manifest (assembly metadata) and signs that hash with the private key. The resulting digital signature is stored in the file that contains the manifest. When an assembly is delay signed, Al.exe does not compute and store the signature, but just reserves space in the file so the signature can be added later.  The default is **/delaysign-**.  The **/delaysign** option has no effect unless used with **/keyfile** or **/keyname**.  For example, using **/delaysign+** enables a tester to put the assembly in the global cache. After testing, you can fully sign the assembly by including the private key in the assembly.   |  | | --- | | **Description: NoteNote** | | Before using the [Gacutil.exe (Global Assembly Cache Tool)](http://msdn.microsoft.com/en-us/library/ex0ss12c.aspx) to put a delay-signed assembly into the global cache, use the [Sn.exe (Strong Name Tool)](http://msdn.microsoft.com/en-us/library/k5b5tt23.aspx) to register the assembly for verification skipping. For example, Sn.exe –Vr delaySignedAssembly. Use this only for development. |   You can also specify this option as a custom attribute ([AssemblyDelaySignAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblydelaysignattribute.aspx)) in the source code for any MSIL module. |
| **/descr**[**iption**]**:***text* | Specifies a string for the [Description](http://msdn.microsoft.com/en-us/library/system.reflection.assemblydescriptionattribute.description.aspx) field in the assembly. Place the string in double quotation marks (" ") if *text* contains a space. This string is a custom attribute on the assembly and is available for viewing with reflection.  If you do not specify **/win32res**, **/description** appears in File Explorer as the Win32 **Comments** resource.  If text is an empty string, the Win32 **Comments** resource appears as a single space.  If you specify **/win32res**, **/description** will not affect the Win32 resource information.  You can also specify this option as a custom attribute ([Description](http://msdn.microsoft.com/en-us/library/system.reflection.assemblydescriptionattribute.description.aspx)) in the source code for any MSIL module. |
| **/e[vidence]:***file* | Embeds *file* in the assembly with the resource name of Security.Evidence.  You cannot use Security.Evidence for regular resources. |
| **/fileversion:***version* | Specifies a string for the **File Version** field in the assembly. This string is a custom attribute on the assembly and is available for viewing with reflection.  If you do not specify **/win32res**, **/fileversion** will be used as the Win32 **File Version** resource. If you do not specify **/fileversion**, the Win32 **File Version** resource will be populated by the Win32 **Assembly Version** resource.  If **/win32res** is specified, **/fileversion** does not affect the Win32 resource.  You can also specify this option as a custom attribute (AssemblyFileVersionAttribute) in the source code for any MSIL module. |
| **/flags:***flags* | Specifies a value for the **Flags** field in the assembly. Possible values for *flags*:  0x0000  The assembly is side-by-side compatible.  0x0010  The assembly cannot execute with other versions if they are executing in the same application domain.  0x0020  The assembly cannot execute with other versions if they are executing in the same process.  0x0030  The assembly cannot execute with other versions if they are executing on the same computer.  You can also specify this option as a custom attribute ([AssemblyFlagsAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblyflagsattribute.aspx)) in the source code for any MSIL module. |
| **/fullpaths** | Causes Al.exe to use the absolute path for any files that are reported in an error message. |
| **/help** | Displays command syntax and options for the tool. |
| **/keyf[ile]:***filename* | Specifies a file (*filename*) that contains a key pair or just a public key to sign an assembly. The compiler inserts the public key in the assembly manifest and then signs the final assembly with the private key. See the [Strong Name Tool (Sn.exe)](http://msdn.microsoft.com/en-us/library/k5b5tt23.aspx) for information about generating key files and installing key pairs into key containers.  If you are using delayed signing, this file will usually have the public key but not the private key.  The public key (of the key pair) information appears in the .publickey field of the assembly.  You can also specify this option as a custom attribute ([AssemblyKeyFileAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblykeyfileattribute.aspx)) in the source code for any MSIL module.  If both **/keyfile** and **/keyname** are specified (either by command-line option or by custom attribute) in the same compilation, Al.exe will first try the container specified with **/keyname**. If that succeeds, the assembly is signed with the information in the key container. If Al.exe does not find the key container, it will try the file specified with **/keyfile**. If that succeeds, the assembly is signed with the information in the key file and the key information will be installed in the key container (similar to the -i option in [Sn.exe](http://msdn.microsoft.com/en-us/library/k5b5tt23.aspx)) so that on the next compilation, the **/keyname** option will be valid. |
| **/keyn[ame]:***text* | Specifies a container that holds a key pair. This will sign the assembly (give it a strong name) by inserting a public key into the assembly manifest. Al.exe will then sign the final assembly with the private key.  Use Sn.exe to generate a key pair.  The key information appears in the .publickey field of the assembly.  Place *text* in double quotation marks (" ") if there is an embedded space.  You can also specify this option as a custom attribute ([AssemblyKeyNameAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblykeynameattribute.aspx)) in the source code for any MSIL module. |
| **/main:***method* | Specifies the fully qualified name (*class*.*method*) of the method to use as an entry point when converting a module to an executable file. |
| **/nologo** | Suppresses the banner, or logo, displayed at the command line when you invoke Al.exe. |
| **/out:***filename* | Specifies the name of the file produced by Al.exe. This is a required option. |
| **/platform**:*text* | Limits which platform this code can run on; must be one of x86, Itanium, x64, anycpu (the default), or anycpu32bitpreferred. |
| **/prod[uct]:***text* | Specifies a string for the **Product** field in the assembly. Place the string in double quotation marks (" ") if *text* contains a space. This string is a custom attribute on the assembly and is available for viewing with reflection.  If you do not specify **/win32res**, **/product** appears in File Explorer as the Win32 **Product Name** resource.  If text is an empty string, the Win32 **Product Name** resource appears as a single space.  If you specify **/win32res**, **/product** will not affect the Win32 resource information.  You can also specify this option as a custom attribute ([AssemblyProductAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblyproductattribute.aspx)) in the source code for any MSIL module. |
| **/productv[ersion]:***text* | Specifies a string for the **Product Version** field in the assembly. Place the string in double quotation marks (" ") if *text* contains a space. This string is a custom attribute on the assembly and is available for viewing with reflection.  If you do not specify **/win32res**, **/productversion** will be used as the Win32 **Product Version** resource. If you do not specify **/productversion**, the Win32 **Product Version** resource will be populated by the Win32 **File Version** resource.  If you specify **/win32res**, **/productversion** will not affect the Win32 resource information.  You can also specify this option as a custom attribute ([AssemblyInformationalVersionAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblyinformationalversionattribute.aspx)) in the source code for any MSIL module. |
| **/t[arget]:lib[rary] | exe | win[exe]** | Specifies the file format of the output file: lib[rary] (code library), exe (console application), or win[exe] (Windows-based application). The default is lib[rary]. |
| **/template:***filename* | Specifies the assembly, *filename*, from which to inherit all assembly metadata, except the culture field.  An assembly that you create with **/template** will be a satellite assembly. |
| **/title:***text* | Specifies a string for the **Title** field in the assembly. Place the string in double quotation marks (" ") if *text* contains a space. This string is a custom attribute on the assembly and is available for viewing with reflection.  If you do not specify **/win32res**, **/title** appears in File Explorer as the Win32 **Description** resource, which is used by the shell as the friendly name of an application. It is also displayed on the **Open With** submenu of the shortcut menu for a file type for which there are multiple supporting applications.  If text is an empty string, the Win32 **Description** resource appears as a single space.  If you specify **/win32res**, **/title** will not affect the Win32 resource information.  You can also specify this option as a custom attribute ([AssemblyTitleAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblytitleattribute.aspx)) in the source code for any MSIL module. |
| **/trade[mark]:***text* | Specifies a string for the **Trademark** field in the assembly. Place the string in double quotation marks (" ") if *text* contains a space. This string is a custom attribute on the assembly and is available for viewing with reflection.  If you do not specify **/win32res**, **/trademark** appears in File Explorer as the Win32 **Trademark** resource.  If text is an empty string, the Win32 **Trademark** resource appears as a single space.  If you specify **/win32res**, **/trademark** will not affect the Win32 resource information.  You can also specify this option as a custom attribute ([AssemblyTrademarkAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblytrademarkattribute.aspx)) in the source code for any MSIL module. |
| **/v[ersion]:***version* | Specifies version information for this assembly. The format of the version string is *major*.*minor*.*build*.*revision*. The default value is 0.  If you do specify **/version**, you must specify *major*. If you specify *major* and *minor*, you can specify an asterisk (\*)for *build*. This causes *build* to be equal to the number of days since January 1, 2000, local time, and *revision* to be equal to the number of seconds since midnight of the current day, local time, divided by 2.  If you specify *major*, *minor*, and *build*, you can specify an asterisk for *revision*. This causes *revision* to be equal to the number of seconds since midnight of the current day, local time, divided by 2.  To summarize, the valid version strings are as follows:  X  X.X  X.X.\*  X.X.X  X.X.X.\*  X.X.X.X  where X is any unsigned short constant except 65535 (0-65534).  If you do not specify **/win32res**, **/version** will be used as the Win32 **Assembly Version** resource.  If you do not specify **/win32res**, **/productversion**, and **/fileversion**, **/version** will be used for the **Assembly Version**, File Version, and **Product Version** Win32 resources.  If you specify **/win32res**, **/version** will not affect the Win32 resource information.  You can also specify this option as a custom attribute ([AssemblyVersionAttribute](http://msdn.microsoft.com/en-us/library/system.reflection.assemblyversionattribute.aspx)) in the source code for any MSIL module. |
| **/win32icon:***filename* | Inserts an .ico file in the assembly. The .ico file gives the output file the desired appearance in File Explorer. |
| **/win32res:***filename* | Inserts a Win32 resource (.res file) in the output file. A Win32 resource file can be created by using the Resource Compiler. The Resource Compiler is invoked when you compile a Visual C++ program; a .res file is created from the .rc file. |
| *@filename* | Specifies a response file that contains Al.exe commands.  Commands in the response file can appear one per line or on the same line, separated by one or more spaces. |
| **/?** | Displays command syntax and options for the tool. |

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

All Visual Studio compilers produce assemblies. However, if you have one or more modules (metadata without a manifest), you can use Al.exe to create an assembly with the manifest in a separate file.

To install assemblies in the cache, remove assemblies from the cache, or list the contents of the cache, use the [Global Assembly Cache Tool (Gacutil.exe)](http://msdn.microsoft.com/en-us/library/ex0ss12c.aspx).

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

The following command creates an executable file t2a.exe with an assembly from the t2.netmodule module. The entry point is the Main method in MyClass.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode_b87b8165-5544-4285-ae28-031c3ed9828b');)

al t2.netmodule /target:exe /out:t2a.exe /main:MyClass.Main