

 Filter by title

- ildasm.exe (IL disassembler)
- Installutil.exe (Installer tool)
- Lc.exe (License compiler)
- Mage.exe (Manifest generation and editing tool)
- MageUI.exe (Manifest generation and editing tool, graphical client)
- MDbg.exe (.NET Framework command-line debugger)
- Mgmtclassgen.exe (Management strongly typed class generator)
- Mpgo.exe (Managed profile guided optimization tool)
- Ngen.exe (Native image generator)
- Peverify.exe (PEVerify tool)
- Regasm.exe (Assembly registration tool)
- Regsvcs.exe (.NET services installation tool)**
- Resgen.exe (Resource file generator)
- SecAnnotate.exe (.NET security annotator tool)
- SignTool.exe (Sign tool)
- Sn.exe (Strong name tool)
- SOS.dll (SOS debugging extension)
- SqlMetal.exe (Code generation tool)
- Storeadm.exe (Isolated storage tool)
- Tlbexp.exe (Type library exporter)
- Tlbimp.exe (Type library importer)
- Winmdexp.exe (Windows runtime metadata export tool)
- Winmdexp.exe error messages
- Winres.exe (Windows Forms resource editor)

> Additional APIs

> What's new and obsolete

Code analysis

 **Download PDF**

[Learn](#) / [.NET](#) / [.NET Framework](#) /

  

Regsvcs.exe (.NET Services Installation Tool)

Article • 09/15/2021 • [14 contributors](#)

 [Feedback](#)

In this article

- [Syntax](#)
- [Parameters](#)
- [Remarks](#)
- [Examples](#)
- [See also](#)

The .NET Services Installation tool performs the following actions:

- Loads and registers an assembly.
- Generates, registers, and installs a type library into a specified COM+ application.
- Configures services that you have added programmatically to your class.

To run the tool, use [Visual Studio Developer Command Prompt](#) or [Visual Studio Developer PowerShell](#).

At the command prompt, type the following:

Syntax

Console  Copy

```
regsvcs [/c | /fc | /u] [/tlb:typeLibraryFile] [/extlb]
[/reconfig] [/comonly] [/appname:applicationName]
[/nologo] [/quiet]assemblyFile.dll
```

Parameters

 **Expand table**

Argument	Description
<i>assemblyFile.dll</i>	The source assembly file. The assembly must be signed with a strong name. For more information, see Signing an Assembly with a Strong Name .

 **Expand table**

Option	Description
<i>/appdir: path</i>	Specifies the root directory of the application.
<i>/appname: applicationName</i>	Specifies the name of the COM+ application to either find or create.
<i>/c</i>	Creates the target application.

<code>/componly</code>	Configures components only; ignores methods and interfaces.
<code>/exapp</code>	Specifies to the tool to expect an existing application.
<code>/extlb</code>	Uses an existing type library.
<code>/fc</code>	Finds or creates the target application.
<code>/help</code>	Displays command syntax and options for the tool.
<code>/noreconfig</code>	Does not reconfigure an existing target application.
<code>/nologo</code>	Suppresses the Microsoft startup banner display.
<code>/parname: <i>name</i></code>	Specifies the name or id of the COM+ application to either find or create.
<code>/reconfig</code>	Reconfigures an existing target application. This is the default.
<code>/tlb: <i>typelibraryfile</i></code>	Specifies the type library file to install.
<code>/u</code>	Uninstalls the target application.
<code>/quiet</code>	Specifies quiet mode; suppresses the logo and success message display.
<code>/?</code>	Displays command syntax and options for the tool.

Remarks

Regsvcs.exe requires a source assembly file specified by *assemblyFile.dll*. This assembly must be signed with a strong name. For more information on strong name signing, see [Signing an Assembly with a Strong Name](#). The names of the target application and the type library file are optional. The *applicationName* argument can be generated from the source assembly file and will be created by Regsvcs.exe, if it does not already exist. The *typelibraryfile* argument can specify a type library name. If you do not specify a type library name, Regsvcs.exe uses the assembly name as the default.

When Regsvcs.exe registers a component's methods, it is subject to the [demands](#) and [link demands](#) on those methods. Because the tool executes in a fully-trusted environment, most demands for a permission succeed. However, Regsvcs.exe cannot register components with methods protected by a demand or link demand for the [StrongNameIdentityPermission](#) or the [PublisherIdentityPermission](#).

You must have administrative privileges on the local computer to use Regsvcs.exe.

If Regsvcs.exe fails while performing any of these actions, it displays corresponding error messages.

Examples

The following command adds all public classes contained in `myTest.dll` to `myTargetApp` (an existing COM+ application) and produces the `myTest.tlb` type library.

ConsoleCopy

```
regsvcs /appname:myTargetApp myTest.dll
```

The following command adds all public classes contained in `myTest.dll` to `myTargetApp` (an existing COM+ application) and produces the `newTest.tlb` type library.

ConsoleCopy

```
regsvcs /appname:myTargetApp /tlb:newTest.tlb myTest.dll
```

See also

- [Tools](#)
- [How to: Sign an Assembly with a Strong Name](#)
- [Developer command-line shells](#)

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see [our contributor guide](#).



.NET feedback

.NET is an open source project. Select a link to provide feedback:

 [Open a documentation issue](#)

 [Provide product feedback](#)