We use optional cookies to improve your experience on our websites, such as through social media connections, and to display personalized advertising based on your online activity. If you reject optional cookies, only cookies necessary to provide you the services will be used. You may change your selection by clicking “Manage Cookies” at the bottom of the page. Privacy Statement Third-Party Cookies

Accept Reject Manage cookies

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

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

Parameters

Expand table

Argument	Description
assemblyFile.dll	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
/appdir: *path*	Specifies the root directory of the application.
/appname: *applicationName*	Specifies the name of the COM+ application to either find or create.

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

```
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.

```
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

English (United States) Your Privacy Choices Theme

Manage cookies Previous Versions Blog Contribute Privacy Terms of Use Trademarks

© Microsoft 2024